Digital Terrain Models Generation from Airborne LiDAR Point Clouds Using A Multi-scale Terrain Filtering Method

by

Hufeng Chen

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Science

in

Geography

Waterloo, Ontario, Canada, 2012

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Abstract

A Digital Terrain Model (DTM) is an important topographic product, required in many applications. Data needed to create a DTM was traditionally obtained via land surveying, however this method can be costly and time consuming depending on the size of the geographic area. Over time, the land surveying was partially replaced by photogrammetry. Today, airborne Light Detection and Ranging (LiDAR) has become another powerful alternative that collect 3D point clouds for digital surface models (DSM) acquisition. LiDAR is especially useful when dealing with heavily vegetated areas using a canopy penetration feature of laser pulse. Nowadays, LiDAR plays an important role in DTM generation.

This thesis presents a hierarchical recovery method to generate DTMs from a cloud of 3D points composed of "single returns" and "multiple returns" from laser pulses using the idea of layering. The proposed method will begin by registering the last return points, then layering them. The layering is done by dividing the points into different height layers and assigning layer numbers to each point. The layer numbers are used as a comparison feature in a later identification process. Then a series of rasterized pyramid levels, which consists of the lowest points in each cell, are generated. After layering, outliers are removed; cells

in the top level are assumed as terrain points and used as references for identifying cells in the second level. The identification process will identify the cells of the second level into terrain cells and off-terrain cells, and an interpolation will then occur in the cells which identified as off-terrain. The interpolated level will be used as references for the next level and the same process is then repeated for each level that comes after. Once this process has been completed for the bottom level, the proposed method adjusts the results based on the first return feedback, followed by another interpolation. As a result, the final DTM is produced.

The developed method is data driven, and does not assume a prior knowledge about the scene complexity. The proposed method was tested with three airborne LiDAR datasets, covering different terrain types and filtering difficulties. Results illustrated that the proposed method can perform well for areas of flat terrain or gentle slope A comparative study was conducted over existing filters and showed that results of the proposed method has similar accuracy in above mentioned area and faster speed than two comparing algorithms.

Acknowledgements

First of all, I am extremely grateful to my supervisor, Dr. Jonathan Li, who leaded me into remote sensing society, coached me how to do research step by step from beginning, provided me precious comments and insightful advice not only for my research but also for the courses I took, guided me finishing and refining my thesis. Dr. Jonathan Li also provided me a lot of personal support especially my first two years in Canada. His constant stream of ideas, insights and suggestions and great help are invaluable to me.

Next, I would like to express my sincere gratitude to Dr. Su-Yin Tan, Dr. Alexander Brenning and Dr. James R. Craig, for agreeing to be part of my thesis committee and for reading my thesis as well as giving critical comments and valuable suggestions. I am so grateful to the time they spent for me.

Many thanks also go to of Remote Sensing and Geospatial Technology team, especially thanks to the great help provided by Haiyan Guan and Yuanming Shu. I would not be able to finish this thesis without their precious input from their experiences. Also great thanks to Sabrina Li helps to proof read the entire thesis, her revisions are great.

Finally, I would like to express my indebtedness to my parents. They have been always trying their best to support me in all aspect with no exception from the beginning of my life. I would be nothing without them.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ALTM	Airborne Laser Terrain Mapper
ALS	Airborne LiDAR System
ASM	Activities Shape Model
DSM	Digital Surface Model
DTM	Digital Terrain Models
FEM	Finite Element Method
FPC	Filtered Point Cloud
GIS	Geographical Information Systems
GPS	Global Positioning System
IDE	Integrated Development Environment
IDW	Inverse Distance Weighted
IMU	Inertial Measurement Unit
InSAR	Interferometric Synthetic Aperture Radar
ISPRS	International Society for Photogrammetry and Remote Sensing
LiDAR	Light Detection and Ranging
MHT	Multi-scale Hermite transform
MLS	Mobile LiDAR System
MTF	Multi-scale Terrain Filtering
nDSM	Normalized Digital Surface Model
REIN	Repetitive Interpolation
SMA	Surveying and Mapping Agencies
TIN	Triangular Irregular Network
TLS	Terrestrial LiDAR System
WGS	World Geodetic System

Chapter 1. Introduction

In this chapter, the definition and generation of the Digital Terrain Model (DTM) are introduced in Sections 1.1, as well as the advantages of generating the DTM from airborne light detection and ranging (LiDAR) point clouds. Section 1.2 addresses the current difficulties in developing DTM from airborne LiDAR data. In Section 1.3, the objectives and scope of this thesis are presented, and the structure of this thesis is outlined in Section 1.4.

1.1 Background

In this section, the definition, application and significance of DTMs are discussed. Then DTM generation methods are introduced, including a comparison between the traditional way and remote sensing. Also, a comparison between the photogrammetric method and laser scanning method is further discussed.

1.1.1 DTM Definition

"A digital terrain model is a continuous function that maps from 2D planimetric position to terrain elevation $z=f(x,y)$" (Pfeifer and Mandlburger, 2008, p. 2). Essentially a DTM is a digital 3D representation of the terrain's surface. In some scenarios, this term can be interchangeable with Digital Elevation Model (DEM). The difference is that a DTM may

include some other topographic features such as break lines (Pfeifer, 2008). The digital surface model (DSM, also known as digital canopy model, DCM) is the same as the DTM in open areas. In contrast to DTM, DSM includes off-terrain objects such as vegetation, buildings, etc. (Pfeifer, 2008).

DTM is one of the most critical topography products. It plays an important role in fields such as mapping, civil engineering, hydrology, hydro-geography, natural resource management, and disaster management. Its widespread applications have ranged from traditional usage to newer and more innovative utilizations such as measurement of the forest depth and density, flood mapping, avalanches and landslides, route mapping, national defense and aerial surveying (Pike, 1988; Toutin, 2008; Korupa et al., 2010). As a result, it is clear that DTM has a wide range of usages in geosciences and engineering, its applications are now considered as the norm of geographical information systems (GIS) industry (Li et al., 2005).

1.1.2 DTM Generation

A DTM was traditionally produced by direct land surveying. It was then mainly replaced by analytical photogrammetry, which requires manual surface feature observation and interpolation methods to generate the DTM. In the last decade, high resolution and automatic remote sensing methods such as automated image matching, Interferometric

Synthetic Aperture Radar (InSAR), and Light Detection and Ranging (LiDAR), have become available (Briese, 2010).

Automatic DTM generation algorithms have partially replaced human interpolation and become a post-processing step after the data acquisition from these remote sensing systems. According to Briese (2010), this phase is usually divided into two steps: classification and interpolation. The classification step extracts the bare earth information (such as elevation, intensity, multiple-returns, or some calculated features like normal vector, segments) from the acquired data, which automatically classifies the gathered data into terrain and off-terrain. This process is also known as "filtering" in the airborne laser scanning community. Subsequently, the DTM can be generated by some interpolation of the extracted terrain data (Briese, 2010).

1.1.3 DTM Generation from LiDAR Data

LiDAR (Light Detection and Ranging) is also known as Laser Ranging, Laser Altimetry, Laser Scanning, or Laser Detection and Ranging (Jiang et al., 2005). LiDAR technology was developed in the late 1960s (Mantis, 2010). The first commercial LiDAR mapping system was developed in 1993, and was used in topographic mapping (Liadsky, 2007). In 1994, a discussion about the new method to generate the DTM by using laser scanning was introduced by Surveying and Mapping Agency (SMA) of the Federal States of Germany

(Petzold et al., 1999). After 1996, more companies began to develop commercial LiDAR systems, and offered many kinds of mapping services (Jiang et al., 2005). In the last decade, there are increasing requirements and research for this technology which enabled the rapid development of the LiDAR technology.

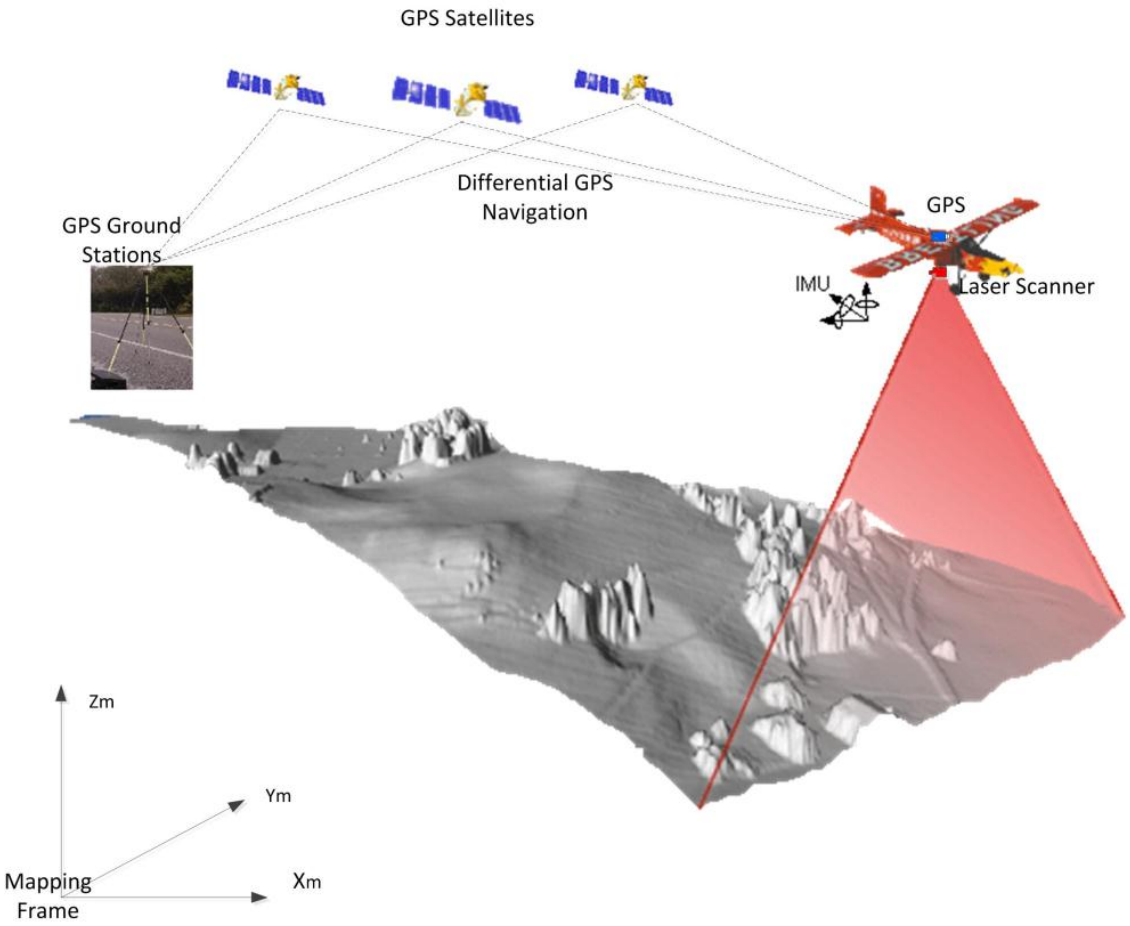


Figure 1-1 Principle demonstration of LiDAR system (Guan et al., 2011)

According to the difference of installation platforms, LiDAR systems can be divided into Airborne LiDAR System (ALS), Terrestrial LiDAR System (TLS) and Mobile LiDAR System (MLS). As shown in **Figure 1-1**, an airborne LiDAR system consists of a laser

scanner, a Global Positioning System (GPS) and an Inertial Measurement Unit (IMU). The laser scanner usually is installed under the bottom of the aircraft. It discharges laser pulses and "determines the distance between ground objects and the sensor by measuring the time a pulse of transmitted energy takes to return to the LiDAR sensor" (Meng et al., 2010, p. 833). Based on the distance from the laser scanner, the platform position recorded by GPS and the aircraft attitude information from IMU, the coordinates of the measured objects can be calculated (Liu, 2008). Though a couple of coordinate transformations, the system can acquire high-accuracy World Geodetic System 1984 (WGS84) coordinates (Guan, 2011).

Airborne LiDAR technology has its applications in topographic mapping, vegetation mapping in forest and wetlands, mapping of roads, power lines, and coast lines, 3D city modeling, disaster assessment, and more (Liu, 2008). For DTM generation, the airborne LiDAR system has many advantages compared to traditional photogrammetric surveys. LiDAR is an active sensor, which is not affected by the sunlight or shadows thus can be used during the day and night. Photogrammetry is limited in gathering and analyzing the target objects, such as dense urban area, forest, coast lines, wetland, desert, and ice surface. On the other hand, airborne LiDAR systems can efficiently handle surveys and mapping (Baltsavias, 1999). Since the LiDAR data is embedded with the elevation value of the objects being measured, it is convenient to generate DSM or even DTM from the LiDAR data. As previously mentioned, the laser has the ability to partially penetrate vegetation and

reach the ground under the canopy, therefore airborne LiDAR systems can perform very well in generating the DTM/DSM in forested areas. Likewise, the high resolution of the LiDAR data makes it a better fit for the significantly increasing demand of high quality DTMs from the GIS community for 3D virtual-reality environments (Liu, 2008).

1.2 Problems Addressed

As a preliminary task of DTM generation using airborne LiDAR point cloud data, filtering terrain and off-terrain points is critical and fundamental to feature extraction and classification (Briese, 2010). The identified terrain points are used as input for interpolation processes in many developed algorithms. The inappropriate identification will cause deviation in the interpolation process, which could potentially lead to further error and less precision in DTM products (Guo et al., 2010). Filtering is often very challenging and time consuming because of the necessary for processing large amounts of data. Therefore, an efficient and effective filter algorithm is important for DTM generation.

Current filtering algorithms are facing difficulties in handling complex circumstances such as outliers (points lie far above or below the most points), complex objects, steep slopes, attached objects, uncertainty of the terrain definition (such as the ramp of a bridge), vegetation (such as shrubs), discontinuities of the terrain, low elevation objects like road curbs and railway tracks, as well as the combined complex scene (Sithole and Vosselman,

2004; Meng et al., 2010). Some of these problems are critical. The outliers, especially low outliers, can affect the selection of reference points in algorithms that adopt the lowest points as reference terrain points. Scenarios with different building sizes will face a dilemma in choosing a filtering window size. Applying a small window size will mislead the algorithm in identifying a point on a large building as a terrain feature whereas applying a large window size will overlook small terrain relief variations. Objects of low elevation are hard to remove because their heights are very close to that of the terrain. A lot of research has been dedicated to DTM generation, especially to filtering, during the last decade (Meng, et al., 2010). However the problems mentioned above have always acted as a barrier in developing a fast, robust, and reliable automatic filter, creating a major obstacle in DTM generation from airborne LiDAR data (Meng, et al., 2010).

1.3 Thesis Objectives and Scope

Developing a fast, accurate and reliable method for terrain point identification and DTM generation using airborne LiDAR data is challenging (Sithole and Vosselman, 2004; Meng et al., 2010). In this thesis, such challenges are reviewed and the main objective is to develop an improved comprehensive automated identification algorithm to generate DTMs from airborne LiDAR point cloud data. This study also attempts to tackle some of the aforementioned problems including those problems related to outliers, urban complexity and vegetation. This thesis investigates the feasibility of the proposed method

on DTM generation in urban areas, residential areas and forest areas. Specific objectives of the thesis are those according to the following:

(1) To study and understand the three problems stated in the terrain point filtering and DTM generation using airborne LiDAR point cloud data.

(2) To develop a comprehensive identification method that can separate terrain points from off-terrain points to solve the aforementioned problems.

(3) To perform a quantitative accuracy assessment and qualitative visual analysis based on the three types of study sites to assess the performance of the developed method and other existing approaches, and then to determine the superiority of the developed method.

1.4 Thesis Structure

The rest of this thesis is organized as follows:

Chapter 2 reviews previous studies on filtering techniques of separating terrain and off-terrain points, as well as the difficulties in developing filters, and some characteristics of the LiDAR data.

Chapter 3 presents the framework and a detailed explanation of the developed method.

Chapter 4 demonstrates experimental results obtained using different datasets from

different types of study areas. This chapter also includes a performance assessment and comparison of the proposed method and other existing method.

Chapter 5 gives the conclusions including the summary of the proposed method and recommendations for future research.

Chapter 2. Review of Ground Filtering Algorithms

Extraction and interpolation are vital steps in the generation of a DTM. Ground filters, which are used to classify the point cloud into terrain points and off-terrain points, have become a challenging issue for researchers throughout the years. Several types of filter algorithms have been developed. However, the majority of these developed algorithms still encounter problems when processing a complicated topography. This chapter first introduces by the characteristics of LiDAR data in Section 2.1, and then provides a review of previous studies on these filters in Section 2.2. The difficulties of developing a filter are discussed in Section 2.3, followed by the conclusion in Section 2.4.

2.1 Characteristics of LiDAR Data

Before discussing algorithms designed for processing LiDAR point clouds, features of LiDAR data, such as density, accuracy, data distribution and noise should be considered.

- Density of data

The density of the LiDAR data depends on the flight altitude, the atmospheric refraction, the transmit frequency of the laser pulse, and the scanning angle (Axelsson, 1999). The density of data varies with the type of application. For example, 3D city modeling and power line detection requires higher density, while density requirement of DTM

generation is relatively lower. Currently, airborne LiDAR systems can obtain very high density point clouds (1 point/m^2 or more), which can provide very detailed information of the ground in addition to processing difficulties.

- Accuracy of data

When the flight altitude is less than 1000 m, altitudinal accuracy can be as high as 15 to 20 cm, and planimetric accuracy can be around 30 to 100 cm (Baltsavias, 1999). The altitude accuracy can be affected by system error, surface slope, surface roughness, flight altitude, and scanning angle. The higher the flight altitude is, the lower the altitude accuracy will be.

- Data distribution

The LiDAR point cloud data is usually unevenly distributed. Data can be combined with several scanning strips. Adjacent strips usually overlap with each other in order to avoid empty spots. Therefore, in the overlap area data density will be higher, as shown in **Figure 2-1**.

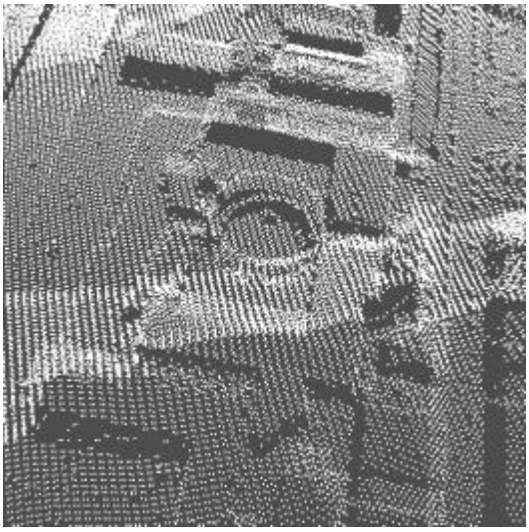


Figure 2-1 Uneven distribution of LiDAR point clouds (Liu, 2008)

In **Figure 2-1**, the scanning router for the LiDAR system is in a "Z" shape (however this can vary), a group of linearly distributed distance measures are scanned with a frequency higher than 5 kHz. Currently some systems can even reach 100 kHz (Pfeifer, 2011). As a result, this feature will lead to a result that the distance between points will be smaller in a strips overlapped area. Moreover, the sample method adopted can make the planimetric distribution uneven. For example, in flat terrain, flight speed can be higher or pulse frequency can be lower to achieve fewer ground points; while in the mountain areas or urban areas which are more complicated require lower flight speed or higher pulse frequency.

- Disturbances

System errors in airborne LiDAR systems mare mainly due to the limitation of the laser scanning devices, and modeling the system error is a way to improve the measurement

accuracy (Baltsavias, 1999). Beside system error, there may be empty spots and outliers. High outliers are generated because the pulse reflected from birds or aircraft, and these points are obviously higher than the points surround them. The pulse may be reflected on the ground multiple times and then received by the sensor, and in this case low outliers can be generated, these points are lower than the points surrounding them. Empty spots of the data may happen in some situations, such as covering, limitation of scanning route, and resolution limitation of the sensors.

Aside from generating the 3D coordinates of the ground surface, many airborne LiDAR systems can collect intensity, multiple return signals, and offer optical image information from digital cameras (Axelsson, 1999). As the reflection rate varies according to the surface and the target material, the intensity information of the points from vegetation, bare-earth, or even water surface are different. Therefore, intensity can help in the classification of point cloud data (Hu, 2003). As the laser impulse can penetrate vegetation (typically the reflection rate is 60% on deciduous trees and 30% on coniferous trees (Pfeifer, 2011), in summer time the reflection rate on deciduous trees falls to 25% (Kilian et al, 1996)), sensors can collect multiple returns in a canopy covered area. This feature can be used in helping to classify vegetation points and ground points, and to estimate the forest volume (Pfeifer, 2011). Furthermore, as the radius of the laser beam footprint on the ground can be larger than 1 meter, multiple returns can be collected on

the edge of buildings as well. Some LiDAR systems can simultaneously obtain optical images which offer extra reference for filtering and classification purpose (Pfeifer, 2011).

2.2 Ground Filtering Algorithms

Data of airborne LiDAR are point clouds distributed in 3D space. In the point clouds some points are terrain points and others are man-made objects, such as buildings, bridges, cars and natural objects, including trees or bushes. To separate or identify these terrain points from off-terrain points (including man-made objects and natural objects) is known as ground filtering. Although there are many different kinds of algorithms that process and interpret LiDAR data based on all kinds of applications, filtering is usually the first step of the process (Guan et al., 2011).

To separate terrain and off-terrain points is always challenging because of the complexities of the problem. This is especially true when filtering in places with complicated geographic and geomorphic conditions. Many algorithms have been developed to solve this problem. According to Briese (2010), these existing ground filtering algorithms may be classified into four categories: morphological filters, progressive densification filters, surface-based filters and segmentation-based filters. In the following section, these four types of filters will be briefly introduced and discussed.

2.2.1 Morphological Filtering

This group of ground filtering method is based on some morphological operators in digital image processing, such as *dilation*, *erosion*, and their combinations *opening* and *closing* as shown in **Figure 2-2**. A *dilation* operator will make the original image longer and thicker, while images shrink after *erosion* operations. An opening operator is a combination of erosion and dilation operations in sequence, while a closing is a combination of dilation and erosion (Gonz á lez et al., 2004). By adopting a certain structure element, also called *window* or *kernel* in different papers, the opening operator can be used for minimum determination of the points, which leads to an approximation of the DTM (Briese, 2010). In 1993, Lindenberger first applied this method based on a robust time series analysis (Briese, 2010; Petzold et al., 1999).

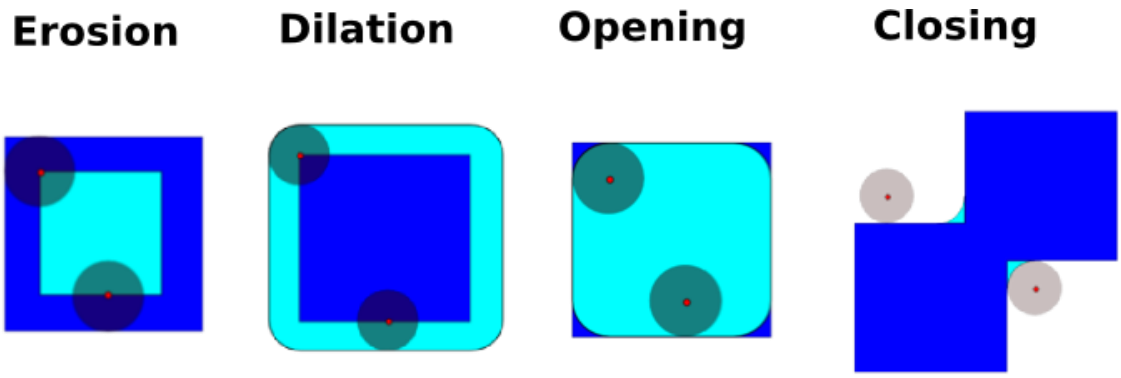


Figure 2-2 The cyan parts are the results of the disk operation on the blue squares

Based on Lindenberger's method, the company TopScan adopted the following technique in 1999. First, a large size moving window is used to select the lowest points in every grid and generate a rough terrain model. Then the generated rough terrain model is used as a

reference to filter all the points. The points with height values over a given threshold are identified as off-terrain points. And a more accurate DEM was built by identified terrain points. The final DEM is consequently generated after repeating this process a few times with a smaller and smaller size of the window (Petzold et al., 1999). The different window size and the given threshold will change the result. If the window size is too large or the threshold is too small, some of the small relief and discontinued area will be smoothed or removed. As well, if the final threshold is too high, many vegetation points will be identified as terrain points. These parameters should clearly be differentiated according to the topographic environment of the study area.

In order to overcome the limitation of window size, Kilian (1996) operated the opening process several times based on different window sizes. Each time different weights are assigned to the laser points within the band width with the weight value depending on the size of the window (Kilian et. al., 1996). A large weight would be assigned when a big window is adopted. After all the opening processes, the points with high weight are likely to be terrain points, and the points with low weight are likely to be off-terrain. Once this has been completed, the DEM can be generated by interpolating the weighted points.

The algorithm developed by Masaharu and Ohtsubo in 2002 consist of two steps. The first step is similar to Kilian's algorithm, which selects the lowest points in a window. Since the window size is relatively small, the whole window might be inside an off-terrain object. In

this case, the lowest point in a window is not a terrain point. Therefore, the second step is designed to eliminate these points. The algorithm creates a buffer (has a size bigger than the window) to each selected point, and calculated the average value of all the selected points inside the buffer. If the elevation difference between the point and the average value is over a user given threshold, the point will be removed. To repeat the removal process three to four times, the algorithms can reach a stable group of terrain points which is used to generate the DTM (Masaharu and Ohtsubo, 2002).

A filter algorithm proposed by Vosselman in 2000 is another type of morphological filter, which is closely related to the erosion operator in mathematical morphology. This algorithm identifies a point by the height value differences between this point and all other points (implemented as comparing the altimetry value with its neighbor points). As shown in **Figure 2-3**, if any of these differences over the threshold, the point will be identified as an off-terrain point (Vosselman, 2000). The threshold of the difference $\Delta h_{max}(d)$ is a function of the distance between two points, which is called the filter kernel function and usually is a non-decreasing function. Three methods for determining the kernel function were introduced by Vosselman, and these methods always try to keep topography features in a DEM. As a result, it may loosen the filtering restriction and lead to some commission errors (misidentifying off-terrain points as terrain points). As named by Vosselman (2000), this algorithm is also classified as a slope-based filter by some researchers (Liu, 2008). It is

based on an assumption that terrain slopes do not rise over a certain threshold (Sithole, 2001). Therefore, it performs well on flat terrain but may misidentify some steep landforms as off-terrain objects.

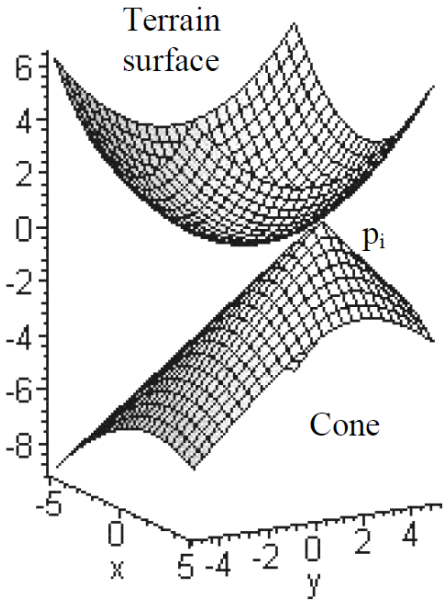


Figure 2-3 The cone surface is determined by the kernel function. If there is no point under the cone surface, the point P_i is identified as terrain point (Sithole, 2001).

In order to improve the applicability to steep landforms, Sithole (2001) modified the kernel function by adding a factor *m* representing the gradient of the terrain slope. This factor varies along with the local terrain, and has different formats for concave surfaces, convex surfaces and flat ground. A *slope map* generated by the local lowest points is also *priori* required to determine the factor *m*. This localized kernel function reduces the type I error (omission error, misidentifying terrain points as off-terrain points). Similarly, Roggero (2001) applied local linear estimation to estimate the terrain slope.

Wack and Wimmer (2002) utilized a hierarchical weighting morphological filter method. They first interpolated a low resolution DEM from the original LiDAR point clouds, and filtered most of the building and thick vegetation by height difference threshold and Laplacian of Gaussian (log) operator, and then calculated the weight function considering the standard deviation of each element. Then the algorithm generates a low resolution DEM based on the weight of the point, and hierarchically generates high resolution DEMs by interpolating low resolution DEMs.

Zhang (2003) gradually increased the window size and applied the height difference limitation in his algorithm in order to eliminate points from cars, vegetation and buildings, while keeping the terrain points. The interpolation from unregulated points to rasterized grids will cause the removal of some terrain points. The low outliers may lead to big errors for this algorithm and the consuming time will increase linearly along with increasing of data. Zhang's method effectively removed most of the off-terrain points, but it is based on an assumption that the slope gradient is constant. The method developed by Chen and Peng (2007) then overcame this slope restriction.

Silván-Cárdenas and Wang (2006) generated DEMs by using a multi-resolution approach based on multi-scale Hermite transform (MHT). This method eliminated the outliers first and then interpolated the LiDAR point clouds into multi-scale rasterized grids. Once this is completed, the algorithm self-adapting eliminated the off-terrain objects by an erosion

operator and multi-scale threshold calculated by MHT. This method can effectively extract the edge and keep a high positioning accuracy. Overall, the size of the structure element has a big influence in filters based on mathematical morphology (Guan et al., 2011).

2.2.2 Progressive Densification

This group of filtering algorithms uses a small number of pre-classified points in the beginning, and then continued by adding qualified points iteratively. The DTMs are typically being reconstructed together with the filtering, and no further interpolation (required in most morphological filters) is needed (Briese, 2010).

Axelsson (2000) introduced a filtering algorithm based on the triangular irregular network (TIN). This algorithm generated a TIN subset of the data with a big cell size at first. The subsets condensed by iteratively adding additional identified terrain points. The TIN is initially under other points, and its curvature is restricted by given parameters (Axelsson, 2000). This algorithm performs well in handling discontinuous surfaces, and can be applied in dense urban areas (Sithole and Vosselman, 2003b).

Krzystek (2003) proposed a method that constructed a rough TIN convex hull at first and continuously adjusted the grid accuracy by applying the finite element method (FEM) to generate a DTM. It was successively applied in a forest area with various kinds of forest structure.

In order to identify more points into the TIN, Sohn and Dowman (2002) developed a further step of Axelsson's method. This algorithm is divided into a "downward" step and an "upward" step. After the initial TIN built by the lowest points in the four corners of the data, the "downward" step keeps on adding lowest points in each triangle to the TIN until no more points occur under it. Afterwards, a buffer is applied in the "upward" densification step to identify higher points.

Another method similar to the TIN progressive densification was introduced by Kobler (2007). A preprocessing of eliminating most off-terrain objects by slope threshold was carried out first. The points within the selecting threshold are used to generate an initial DTM. The TIN is then generated by the seed points choosing from that DTM. The height differences between the TIN and rest points are calculated. If a difference is no larger than the threshold, the point will be identified as a terrain point, and vice versa (Kobler et al., 2007).

Algorithms based on the TIN progressive densification usually have an assumption that the surface is continuous or flat; therefore this type of algorithms has relatively poor performance in keeping the topographic discontinuity and relief (Sithole and Vosselman, 2003b). Furthermore, TIN models store data points and their topology relationship, whose consistency is time consuming and difficult to maintain. Therefore, the adopting of the TIN model also increases the required storage space and processing time of these algorithms

(Jiang et al., 2005).

2.2.3 Surface-based Filtering

Same as progressive densification filters, a surface-based filter uses a surface to represent the DTM and the process generates intermediate DTMs. However, instead of using a region growing from a group of seed points in progressive densification filters, surface-based filtering methods iteratively adjust the weight of points above or under the surface until reaching a stable situation (the result does not change significantly) (Briese, 2010).

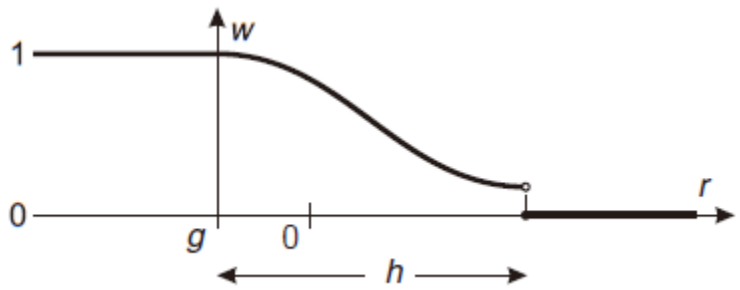


Figure 2-4 The weight function: the horizontal axis r represents the residuals and the vertical axis *w* represents the weight assigned, g represents the ground. The weight assigned is getting lower when the residual getting higher, and when over the given threshold h, the weight will be assigned as zero.

This type of filter was first designed by Kraus and Pfeifer (1998). In their algorithm, the robust interpolation method combines the processes of DEM interpolation and filtering. First, an average surface between the real DTM and the DSM is calculated based on that all the points are assigned to the same weight. Then the residuals (distance between the points

and generated surface) are calculated to determine the new weights. As shown in **Figure 2-4**, the big negative residual points are assigned to a larger weight, because they are closer to the real terrain. The small negative residual points are assigned to a smaller weight for the same reason. For the big positive residual points, they should be eliminated because they probably are not terrain points. Then the algorithm can do the next round of surface interpolation with the new assigned weights. The interpolations and weight assignments are carried out iteratively until the difference between two contiguous surfaces meets the given threshold. This algorithm can perform very well in the situation well mixed with terrain and off-terrain points, e.g. wooded areas, and it has been implemented on the SCOP develop package (a remote sensing tool developed by Institute of Photogrammetry and Remote Sensing (I.P.F.), Vienna, and INPHO GmbH, Stuttgart (SCOP++, 2007)) for DEM and DTM generation in a sparse forest area (Guan et al., 2011). Although it can correctly detect steep topography in above mentioned areas, when it comes with only a cluster of off-terrain points, e.g. large building areas or dense forest areas, the algorithm fails to eliminate the commission errors (misidentifying off-terrain points as terrain points) (Briese, 2010).

Some improvements of this algorithm have been developed afterwards. Kraus and Rieger (1999) added first and last return (laser echoes) information. To overcome the limitation of the mixture of terrain and off-terrain points, Pfeifer (2001) further applied hierarchical

pyramid layers to the method of Kraus and Pfeifer (1998). The modified algorithm first built a rough surface model on the top level, and iteratively used the surface built in higher level to a lower level (with higher resolution), and finally reaches the bottom level which has the best resolution. This algorithm can eliminate large buildings and other off-terrain objects even in dense forest areas. In order to get an even higher quality of the DTM, Kraus and Pfeifer (2001) analyzed vertices of the landform by simulating the rain flow on the ground. By eliminated these vertices, a smoother terrain can be generated.

Elmqvist (2002) adopted the active shape model (ASM) from the digital image processing to estimate the surface model. In his method, an ASM is a surface under the LiDAR dataset. The surface iteratively changes towards the points under the control of the stiffness of the surface (restrict the change of the surface) and the energy function (a connection between the surface and the points) value. The surface connected to the terrain points has the least energy function value. By minimizing the energy function it matches the ASM to the DTM. The shape of the ASM decides the shape of the bare earth. Any point in the buffer of the surface will be identified as terrain point.

2.2.4 Segmentation-based Filtering

Different from the previous three kinds of filters, segmentation-based filtering methods classify a segment (groups of neighboring points with similar properties) instead of a single

point. For example, if a segment is higher than its neighbor segments, all points from this segment will be classified as off-terrain points (Guan et al., 2011). Usually aggregating the points into segments and classifying the segments are two steps of this type of method. Different algorithms can be chosen for both steps. For example, region growing techniques (detect the segments from some "seed points") and detecting clusters (group of points with the similar feature values) are two options for the data segmentation step. Geometrical properties such as height, normal vector, gradient, curvature and other criteria can be adopted in both data segmentation and segments classification steps (Briese, 2010).

One representative sample of the segmentation-based filtering method is presented by Sithole (2005) and Sithole and Vosselman (2005). They partitioned the data into continuous profiles with different orientations. Based on certain criteria, points on a profile are connected as line segments. And by comparing the common points of the line segments from all profiles, the whole segments of the data can be generated.

Roggero (2002) proposed the segmentation method by region growing and major elements analysis based on the laser scanning data. Akel (2003) interpolated the raw LiDAR data into rasterized grids and constituted a TIN model. Then he carried out the region growing based on the threshold of normal vectors of neighboring triangles in the TIN model. Subsequently the algorithm calculates the normal vectors, edges and height differences of segments and extracts roads from the data. At last, the method iteratively constituted DTM

based on the seed points from the extracted roads (Akel and Zilberstein, 2003; Akel and Zilberstein, 2004; Akel and Kremeike, 2005).

Tóvári and Pfeifer (2005) developed another representative approach based on the region growing. The approach selects a random seed point, and chooses several points from its neighborhood. Then it calculates three parameters: the normal vector of the plane constituted by these points, the distance between the points and the plane, and the distance between the seed point and its neighbor points. Based on these parameters, the method continues the region growing process until no more points can be added in. After the segmentation, iteratively weighted interpolation and grouping are carried out to produce a DTM.

Forlani and Nardinocchi (2007) rasterized the raw data and set the lowest height value as the cell value. Then the raster is separated into segments by a self-adapting region growing method. Subsequently the raster is classified into outliers, vegetation, building and terrain based on the geometrical properties and topological relation of segments. Then the approximate terrain surface is calculated by these classified terrain cells, and the algorithm calculates the distance between original point clouds to this surface. If a distance is under the user defined threshold, the according point will be classified as a terrain point.

Many segmentation-based filtering methods are implemented in a raster data format which is easy to borrow image processing algorithms. These methods are usually not based on

geometrical hypothesis (such as “terrain is continuous.”) to describe topographic information. They are using geometry, optics, mathematical statistics and other features to identify point cloud data as larger entities rather than single points. Therefore, they are not influenced significantly by noise and can overcome the problem that wrongly classifies an individual point into a different class than its neighbors from the same segment.

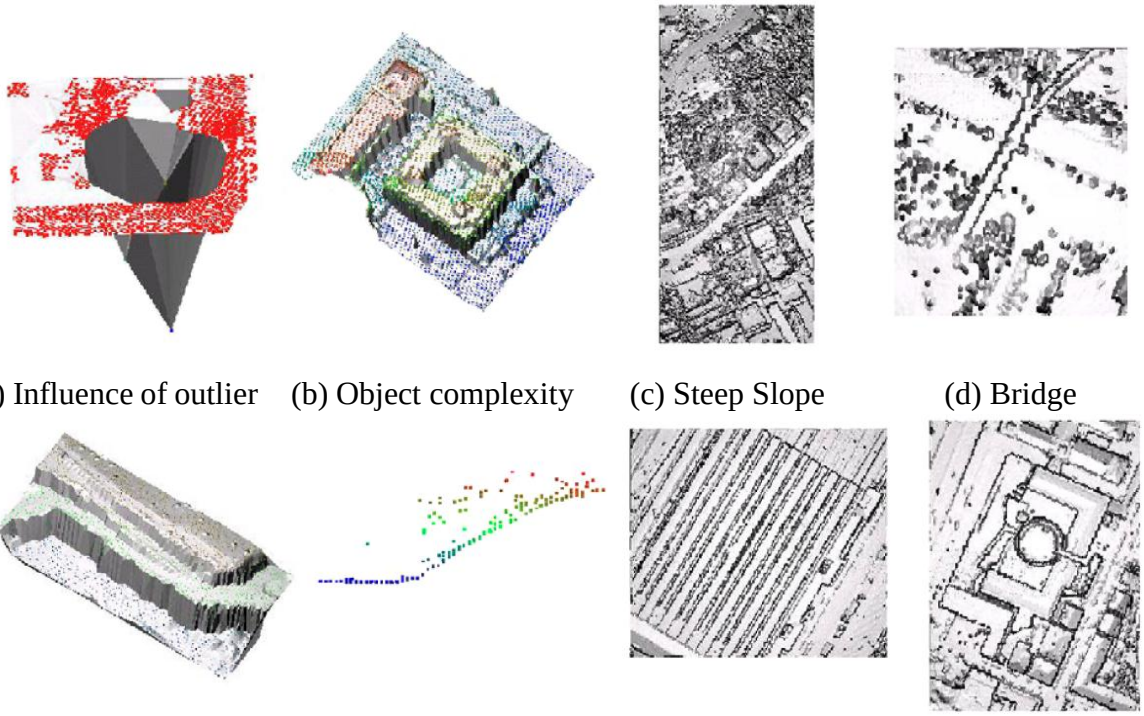
Including the above mentioned methods, there are many methods developed (Briese, 2010; Sithole and Vosselman, 2004; Pfeifer and Mandlburger, 2008; Meng et al., 2010; Liu, 2008). Meng et al. (2009) concluded that these filtering algorithms developed in the last decade could also be classified into more subsets including directional scanning, contour-based filters, TIN-based filters, and interpolation-based filters.

2.3 Difficulties in Ground Filtering Algorithms

The filtering of airborne LiDAR data is usually based on the height, gradient, regional similarity of the point clouds or breaklines of the landform. For example, points with a large height value on the local scale have only a small possibility of being terrain points, and vice versa. The points with big height differences from their neighbors could probably be off-terrain objects. However the diversity and complexity of the terrain and the off-terrain objects will cause difficulty in filtering. These problems include steep slope, sharp ridges, super large or irregular buildings, gross error, vegetation, and characters of

laser scanning data.

According to the report by Sithole and Vosselman (2004), the filtering difficulty can be cataloged as outlier, object complexity, attached objects, vegetation, and discontinuity as shown in **Figure 2-5**.



(a) Influence of outlier (b) Object complexity (c) Steep Slope (d) Bridge

(e) Discontinuities (f) Vegetation on slopes (g) Railway: low bare-earth count (h) Object complexity

Figure 2-5 Data sample of the filtering difficulty (Sithole and Vosselman, 2004)

(1) Outlier

Many algorithms initially adapted the lowest points in an area as the terrain point (Sithole and Vosselman, 2004). This makes the algorithms easily affected by low outlier and multi-storage buildings, and makes unreliable identification of terrain points, such as

shown in **Figure 2-5** (a).

(2) Object complexity

In some complex scenarios, there are very large objects, very small objects, very low objects, etc. If very large objects, such as conjoined buildings as show in **Figure 2-5** (b) and (h), have a larger size than the given window, it cannot be removed automatically, and the low objects, such as cars, low vegetation, is difficult to be separated from the bare earth as shown in **Figure 2-5** (g). Platforms with stairs connected to the terrain are also easily classified as terrain. Objects with complex shape, such as multistoried building, buildings with court yard, are also hard to remove as shown in **Figure 2-5** (b).

(3) Attached objects

Attached objects such as buildings on the slope, bridges, and ramps at least have one end of the structure seamlessly connected to the bare earth. Therefore it is hard to find the border between the objects and the terrain consequently leads to the problem of separating them as shown in **Figure 2-5** (d).

(4) Vegetation

Vegetation usually can be eliminated by the height difference when comparing to the terrain points. However, if the vegetation is on a slope, or very close to the bare earth or buildings, they are hard to eliminate as shown in **Figure 2-5** (f).

(5) Discontinuity

The off-terrain objects in LiDAR point clouds are usually discontinuous to the terrain

which is an important feature to filtering them. However, this rule will also apply to some natural piecewise continuous bare-earth (e.g. steep slopes and sharp ridges as shown in **Figure 2-5** (c) and (e)); consequently this process will cause the deficiency of discontinuity feature in DTM.

According to the review by Ment et al. (2010), specific objects or features are summarized as a barrier of ground filtering algorithms: shrubs, short walls along walkways, bridges, complex buildings, hill cliff, mixed land cover, low and high relief terrain, and not reliable assessment. The list has some common objects with category of Sithole and Vosselman (2004), but is more specific.

2.4 Chapter Summary

Although more criteria are utilized to separate off-terrain points from terrain points, such as breaklines (Dragos, 2004), intensities and full-waveform ALS data (Doneus and Briese, 2006), most methods are based on the position information of the LiDAR point clouds data. These point clouds filtering algorithms are developed by different concepts and understanding of terrain points and off-terrain points. Some of the algorithms can be directly applied in an irregular distributed point cloud dataset, while some others need to rasterize the data first and use well developed digital image processing technologies. Some of the algorithms compare one point to another point, while others compare one

point to a group of points, or even datasets to datasets. These filtering algorithms measure various discontinuous parameters such as height difference, gradient, or the minimum distance from a point to a surface etc. They are under different hypotheses such as slope, surface, morphology, segments etc. Furthermore, some algorithms finish their processing in one step; others may iteratively carry out the process. (Sithole and Vosselman, 2004)

Sithole and Vosselman (2004) compared and analyzed the results, mechanism and features of eight different filtering algorithms in various scenarios based on the urban and forest datasets from International Society for Photogrammetry and Remote Sensing (ISPRS). They listed major problems to these algorithms: the reliable filtering in complicated environment, processing buildings on a hillside, handling unconnected landforms, and keeping the discontinuities of the ground surface etc. The research result concluded that the tested algorithms generally have a good result in processing not too complicated scenarios, which have gentle gradient, include only small buildings, have sparse vegetation, and include large amounts of terrain points. However in other cases such as large building, steep gradient, and discontinuous landform, the reliability becomes lower.

Therefore, manual work is still required in pre-mentioned areas even though the algorithms are very well automated now. Work flows of combining different algorithms have been researched as well, but so far, no method can perfectly handle all scenarios automatically.

One reason could be that previous research has focused only on the point position information and the relationship with its local neighborhoods (Briese, 2010). Therefore, a globalized view of the data could be a direction to explore and is the motivation of this thesis.

Chapter 3. DTM Generation by Iterative Recovery

In this chapter, an overview of the proposed method will be introduced in Section 3.1. The framework and development of the proposed method are detailed in three sections. The pre-processing is discussed in Section 3.2, followed by detailed information about the iteration terrain recovery method in Section 3.3. The refinement of DTM is addressed in Section 3.4. A qualitative and quantitative evaluation method is presented in Section 4.2. At last, the conclusion is in Section 3.5.

3.1 Overview of Iterative Recovery Method

This chapter presents a Multi-scale Terrain Filtering method for automated generation of DTMs from single- and multi-return LiDAR point clouds. The proposed multi-scale terrain filtering (MTF) method identifies terrain points by iteratively recovering terrain models from rasterized pyramid levels (coarse-to-fine multi-scale pseudo-grid images). As shown in **Figure 3-1**, the method consists of three steps: point cloud pre-processing, multi-scale terrain filtering, and DTM refinement.

In the point cloud pre-processing step, all laser scanning points must be pre-processed to retain last-return points of multiple returns (laser echoes), and then are layered with regard to the statistical height histogram of the whole dataset. Two objectives of height

histogram-based layering are to assign the layer numbers to each point for the following MTF implementation, and to remove lower outliers and noises.

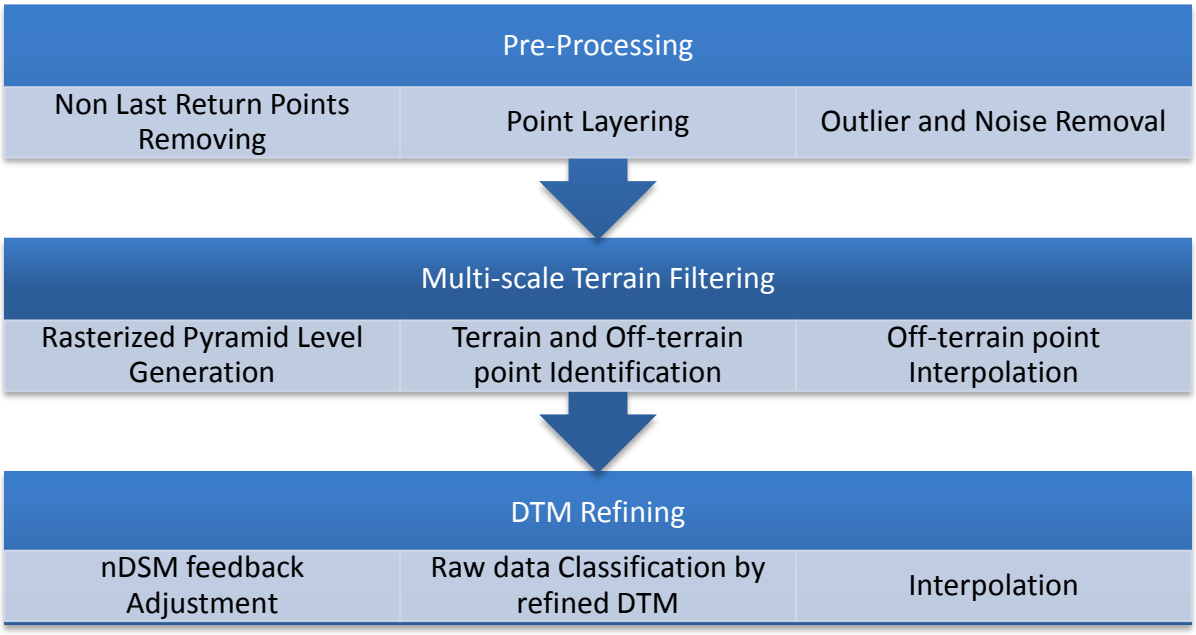


Figure 3-1 Flowchart of Multi-scale Terrain Filtering Method

The second step is the Multi-scale Terrain Filtering, which includes the rasterized pyramid level generation, iterative point identification and interpolation. Several rasterized pyramid levels are generated at first, and lowest points in every grid at every level are marked as representative points. The highest level is referred to an initial digital terrain model, from which the proposed MTF is employed as a reference. Then the identification and interpolation is iteratively processed in every level from the second highest level to the lowest level in the pyramid. The identification is based on comparing two features: one is the layer numbers generated in layering; another one is the slope gradient between the reference point and identifying point. This is followed by the interpolation of identified

off-terrain points. The points from a processed level then become reference points in the identification of next level. Iteratively, digital terrain models are recovered and densified from coarse scales to fine scales.

```
Input Lidar_Image;
//Pre-processing
points.SelectLastReturns();
points.HistogramGeneration();
points.Layering();
points.NoiseElimination();
//Point identification and interpolation
level[n] = points.PyramidLevelsGeneration();
for(level j = n-1 to 0){
   if(j==n){points in level j = reference}
   for(points[i] in level j){
      if(points[i].layernum==reference){
         points[i]=terrain point;}
      else{
         points[i]=off_terrain point;}}
   for(points[i] in level j){
      if(points[i]==off_terrain point){
         points[i].z=points[i].interpolation();
         points[i].layernumRenew();}}}
//DTM Refinement
Rough_DTM = RasterGeneration(points[i]);
Refined_DTM = nDSM_Adjustment(LidarImage.First_Returns, Rough_DTM);
Terrain_Points = Fitering(Refined_DTM, Lidar_Image.Last_Return);
FinalDTM = TerrainPoints.IDW_Interpolation();
Output DTM;
```

Figure 3-2 Pseudo-code of the proposed method in a C# language format

Then the method adjusts the terrain results based on the normalized Digital Surface Model (nDSM). After that, all original laser scanning points are selected based on the refined

digital terrain model. As a result, this produces the final and complete DTM. The pseudo-code of the proposed method is shown in **Figure 3-2**.

3.2 Pre-Processing

The first step in handling the LiDAR data is pre-processing. In this proposed method, three parts are involved: the removal of non-last returns, the layering of data, and the elimination of outliers and noises. In this section, the motivation and the detail of these processes will be discussed.

3.2.1 Non-last Return Points Removal

Many airborne LiDAR systems can record multiple returns while scanning the ground surface. Some features of multiple returns are applied in the proposed method. As aforementioned, a LiDAR device "determines the distance between ground objects and sensor by measuring the time a pulse of transmitted energy takes to return to the LiDAR sensor" (Meng et al., 2010, p. 833). Since the airplane is flying in a high elevation, the laser beam becomes a larger spot when shooting on the ground surface. Therefore, the laser beam may be partially reflected during the traveling. For this reason, LiDAR systems can record a discrete number of echoes in the return signal as shown in **Figure 3-3**. When two or more echoes are detected by LiDAR systems; the first and last echo are referred to the first- and last- return in literature (Pfeifer, 2011). Some LiDAR systems

like the Leica ALS50-II are able to record up to five returns.

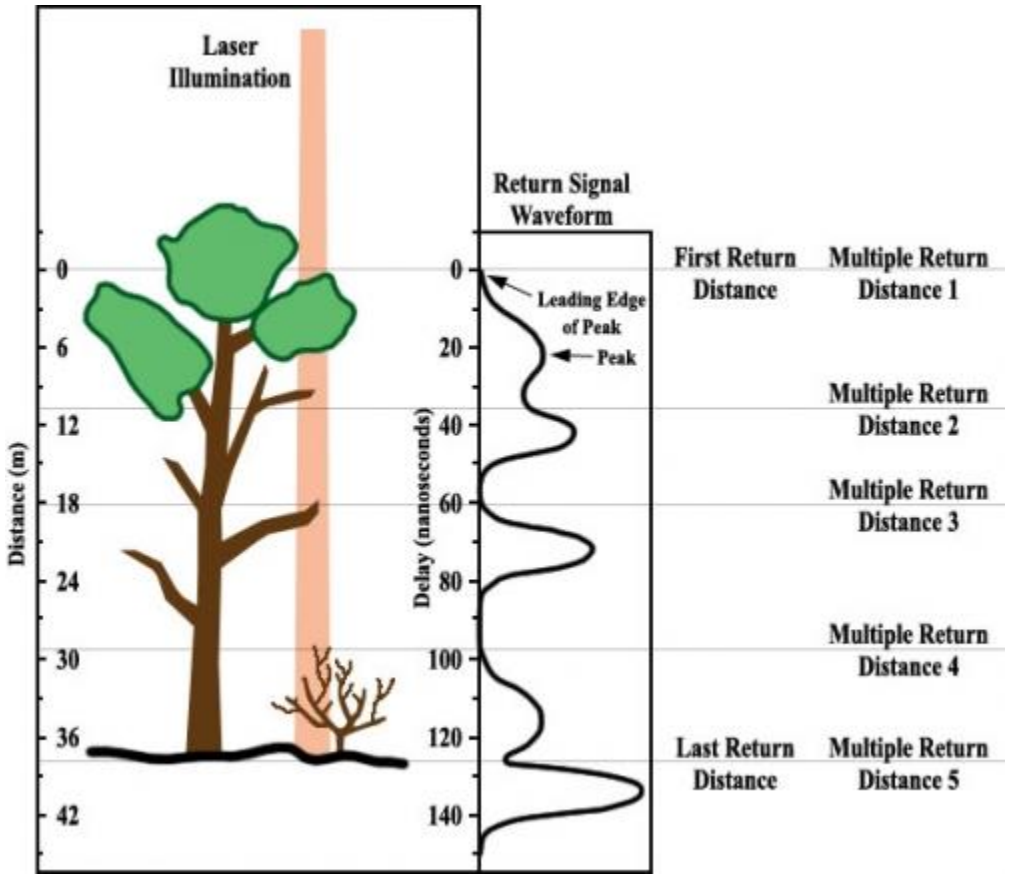


Figure 3-3 Multiple Returns Principle (Schuckman and King, 2011)

Due to the fact that the laser pulse can penetrate the canopy of trees, the laser pulse shooting on the trees can be sporadically reflected by leaves or branches in different elevation. And if the laser pulse shooting on the edge of the buildings, the laser pulse can be partially reflected by the building roof and partially reflected by the connected ground. Furthermore, in most cases the availability of multiple returns can lead to a conclusion that there is vegetation or pinpoint the edge of building (Beraldin et al., 2010).

Since the last returns are always the final one received by the sensor, it can represent the lowest elevation a laser pulse reached among the multiple returns of the pulse. Thus, the non-last return points are the points reflected by vegetation and the edge of buildings. The removal of these non-last return LiDAR points help to eliminate these vegetation or building edge points which are not used in DTM generation, it also can help to partially decrease the processing time. Therefore, the proposed algorithm will only use last returns to extract DTMs. It should be noted that a laser pulse with only one return will be treated as a last return in the proposed method.

3.2.2 Height Histogram Based Layering

The second part of the pre-processing is height histogram based layering. The purpose of layering is to assign a layer number to each point for MTF implementation. In the context of digital image processing, a histogram (a graph of pixel intensity values) is a significant tool for image enhancement, segmentation, matching, etc. The histogram can be viewed as a discrete probability distribution since the relative height of a particularly non-intersecting bar is normalized by the total number of pixels. In the most cases of threshold-based segmentation, constructing histogram is an easy way to individually select each specific histogram mode, and then select the corresponding area by proper thresholds. Since histogram-related methods were first introduced in the field of laser scanning data processing in 1994, they have been widely used in navigation, localization, recognition and

mapping. Histograms give a fast and applicable solution for position estimation. Other than extracting geometric primitives from the laser scanning data, the histogram combines the geometric position information into the discrete probability distribution based graph (Qiu and Han, 2008). Similarly, by finding proper thresholds based on height histogram, range images of laser scanning points can be segmented into several areas of interest.

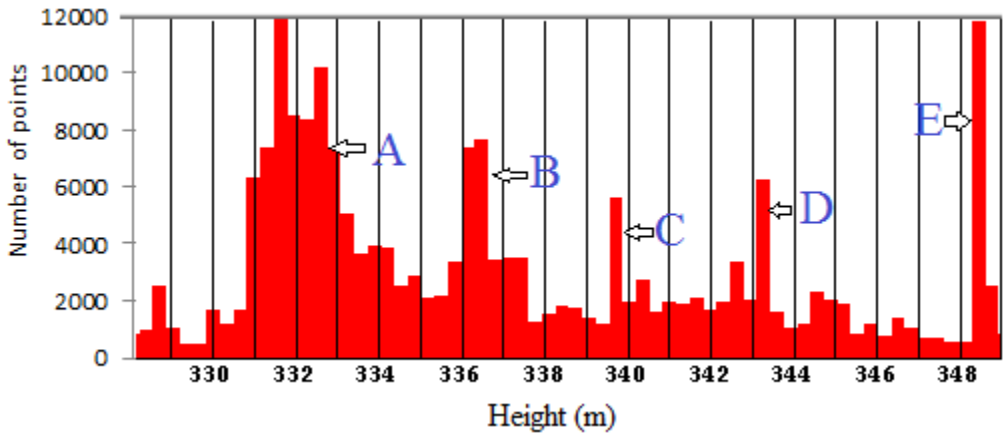
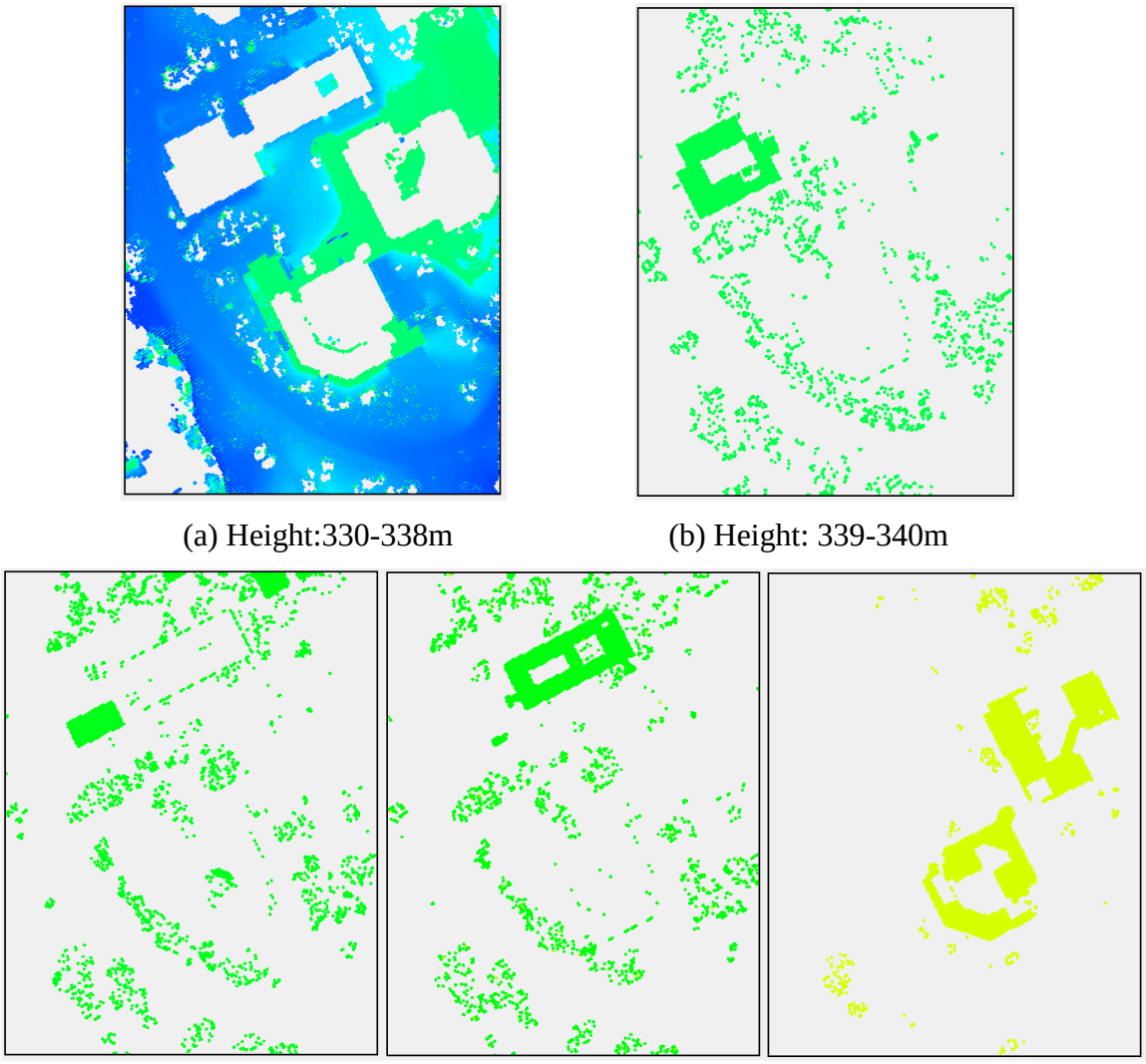


Figure 3-4 Height histogram of Site A
(More information of this study site can be found in Section 4.1.2)

As shown in **Figure 3-4** and accordingly on **Figure 3-5**, it can be observed that buildings and terrain within a small area can be well separated, which can be reflected from a height histogram, where terrain points are aggregated in the lower height sections, and buildings and other high-rise object points are placed in the higher height sections. For example, the two waves A and B on **Figure 3-4**, whose values are from 330 to 335 and from 335 to 338, represent the terrain with a slope. The three bars C, D and E which look like pulses are the roofs of the buildings, which can be seen on **Figure 3-5**.



(a) Height:330-338m (b) Height: 339-340m

(c) Height:342-343m (d) Height: 343-344m (e) Height: 348-349m

Figure 3-5 Points in the different height ranges in Site A

According to the height histogram, the height value range of the terrain points is relatively larger than of building roof points. This is because the buildings are built in a regular shape, and the terrain slope is random. Compared to the vegetation, the point density of terrain is higher. This can be interpreted as a higher number of points in the height histogram. In other words, the different regions in the height histogram correspond to distinct terrain and

object features. As a result, if proper separation values can be found in the height histogram, terrain points can be separated from other object points. Therefore, an automatic way of calculating a separation value for layering the histogram will be discussed.

A peak on the height histogram may correspond to a group of terrain points or building roof points. The separation values should be the values of wave troughs. Based on a given width of a height unit, the height axis is divided into several units. By counting the number of points in each unit, the height histogram can be obtained.

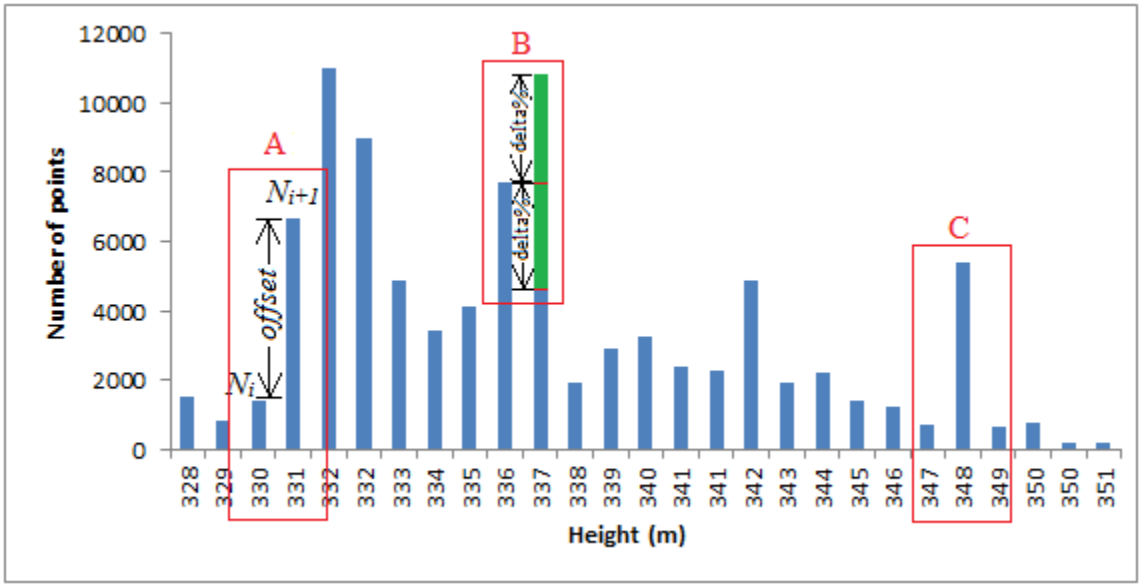


Figure 3-6 Height histogram (modified from histogram of site A) to demonstrate the proposed layering algorithm

To detect peaks from a histogram, there are some algorithms available. Gonzalez and Woods (2002) described an iterative algorithm to automatically detect the separation values. It assumes that there are two peaks in the histogram. A single-threshold selection method is applied to find the best separation value. The algorithm is described as follows:

1) Select an initial estimate threshold (T_0) regarding the height histogram;
2) Calculate the mean grey values μ_1 and μ_2 of the two separations of ground and object points (G and O) and partition them accordingly;
3) Calculate a new threshold ($T_i = (\mu_1 + \mu_2)/2$) to partition the range image;
4) Repeat the step 1-3 until $T_i = T_{i-1}$

In reality, there are many objects over the ground. In other words, more than two peaks exist in the histogram generated from laser scanning points. Therefore, this algorithm which can only separate the histogram into two peaks is not very helpful in this study.

Other algorithms like Kernal desnity estimation which using Gaussian kernel smoothing the histogram to detect peaks (Wand and Jones, 1995). However it will smooth the pulse bar like rectangle **C** in **Figure 3-6**, and those bar usually represents a big flat surface such as the roof of a building which should be assigned into separate layer. Therefore, the proposed method applied the following algorithm.

$$offset = |N_{i+1} - N_i| < delta\% * N_i \quad \textbf{(3-1)}$$

Where N_i represents the number of points in a histogram bin *i*. As shown in **Figure 3-6**, *Offset* is the difference between two adjacent bins. *Delta%* is a user assigned parameter, which defines the acceptable ratio between *Offset* and N_i. If *offset* is smaller than *delta%* of N_i (e.g. inside the green bar in **Figure 3-6**), the two bins will be assigned to the same layer. If the relation between N_i and N_{i+1} does not meet the requirement as shown in

Equation (3-1), histogram bin *i+1* will be assigned to a different layer than histogram bin *i*. The border between histogram bin *i+1* and histogram bin *i* will then become a separation value of the layers. Layers are then generated by dividing the data using these separation height values. The number of layers is determined by the width of height histogram bin and parameter *delta%*, which will be discussed in Section 4.4.1.

3.2.3 Outlier and Noise Removal

In Sithole (2004), outliers are categorized into two types: low and high outliers. High outliers originate from the hits off objects like birds, low flying aircrafts or errors in the laser scanner, which generally have not much influence to the algorithm. Low outliers originate from multi-path errors or the hits in the wall. They have great impact on the generation of DTM because the proposed algorithm assumes that the lowest point in a local neighborhood is a ground point. If low outliers cannot be removed from point clouds before applying MTF, they will cause the erosion of the terrain.

To remove those outliers, which is also known as de-spiking, can be finished in different ways. Examination of the frequency distribution of the elevation values is a common method to detect those outliers, which is applied in the proposed method. Using Delaunay Triangulation to comparing each point to a local elevation reference and then identified the outlier is another way (Meng et al., 2010). Since the proposed method is applying raster to

maintain the data, this method is not applicable. Other than these two, manual examination is another approach to detect and remove the outliers.

In the proposed method, the height histogram, an estimation of the probability density distribution of LiDAR points' heights, is used. As a fact that low outliers are generally a very small amount of points or single points in a local area, they can be discarded if a bin of the height histogram is lower than a specified threshold (parameter: *Minimum Layer*). Meanwhile, some "noises" also can be removed based on the specified threshold. Theoretically, a terrain surface is a continuous and intensive distributed area. Sparsely scattered points in layers have a high possibility of representing other objects that can be eliminated as noises. Those noises might not be real noises, but they are little relevant to in the case of separating terrain from object points.

After removing lower outliers and noises, layer numbers are assigned to each point which belongs to. Those assigned points will used as a rule of MTF for DTM generation.

3.3 Multi-scale Terrain Filtering

Based on the pre-processed points with the assigned layer numbers, the multi-scale terrain filtering method is going to produce a rough DTM (an intermediate result). As shown in **Figure 3-7**, this part includes three steps: generation of rasterized pyramid levels, identification of topographic cells and interpolation of the off-terrain cells.

The rasterized pyramid levels are the framework for the following process, and the levels are divided into regular cells (grids) with a representative height value. The method identifies the cells in a level as either terrain or off-terrain using the reference points from the previous level. Then the cells which are identified as off-terrain will be interpolated. The identification and interpolation will be processed from the second highest level (Level N-1) to the lowest level (Level 0). This method will generate a rough terrain model for the further processing in the end.

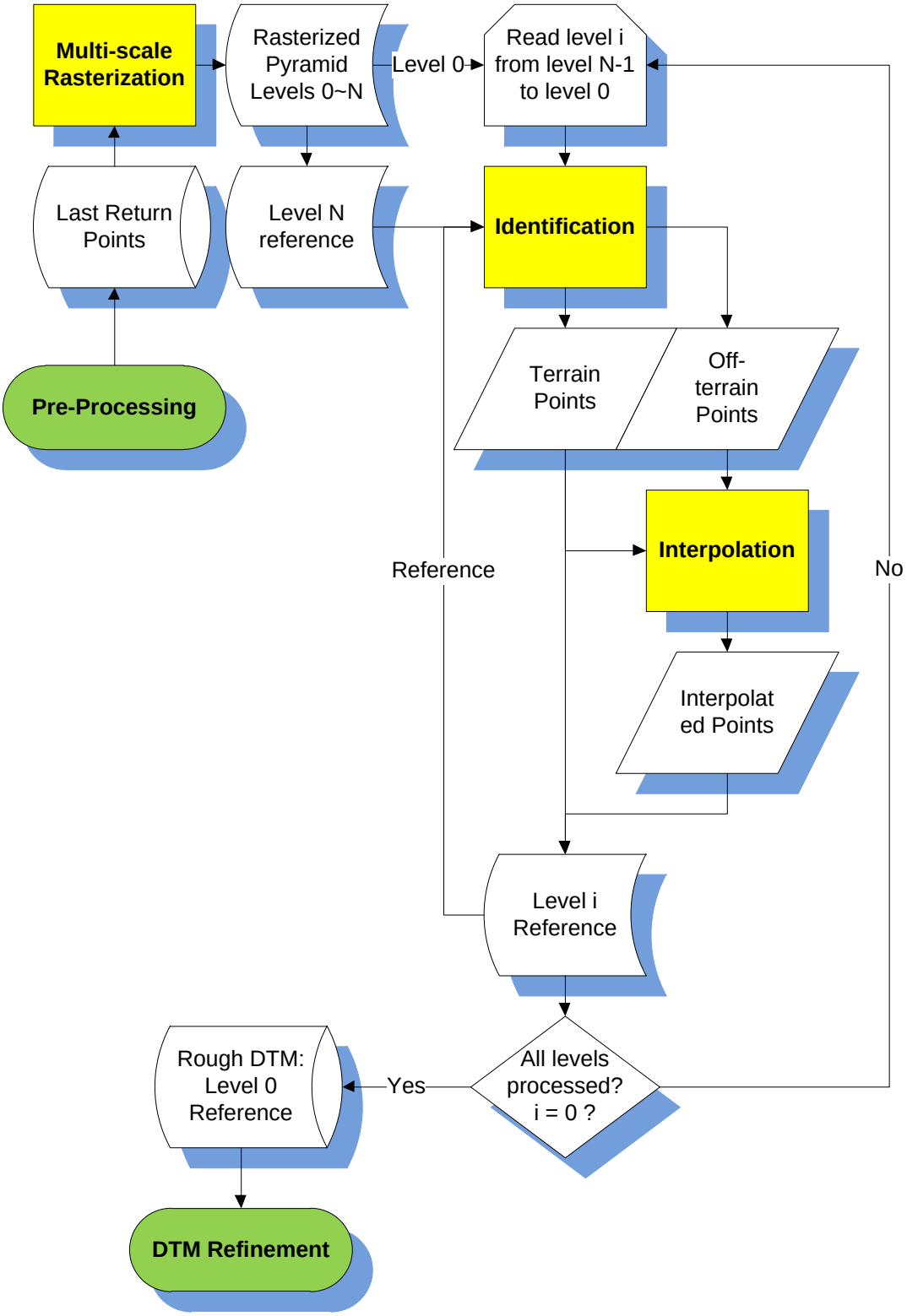


Figure 3-7 Flowchart of Multi-scale Terrain Filtering (MTF)

3.3.1 Multi-scale Rasterized Pyramid Level Generation

Considering that objects are presented in various ways depending on the scale of observation, a theory for multi-scale representation called scale-space theory had been originally developed by computer vision community for automatically analyzing and deriving information from signals that are the results of real-world measurements. Specifically, the scale-space theory is a framework for representing signal and imagery data at different scales through the re-sampling of the original data model (Ali, 2010). Besides machine vision (object recognition and manipulation, visual guided navigation), it has been widely used in typically visual-related tasks, including image processing (enhancement, visualization), signal processing, industrial inspection, remote sensing, automated cartography, data compression. An original model at coarse scale should have details fewer and simpler than that at fine scale in the multi-scale representation. In the fields of image processing and remote sensing, the most common used multi-scale representation is pyramid, which describes grey-level data in combination with the sub-sampling operation with a smoothing step (Keller and Averbuch, 2006; Bunting et al., 2010). Since the structure of raster digital surface models (DSMs) or digital terrain models (DTMs) directly generated from airborne LiDAR data is similar to that of imagery, it is applicable to LiDAR range images. A range image normally is a single-band image composed by a grid of cells, whose values represent those cells' elevations. Converting

three-dimensional (3D) point data to 2D grid format is a major topic in the laser scanning data processing world, and there are a plethora of approaches are based on the format of range image for final LiDAR-driven products, in terms of processing speed, memory, accuracy for particular purposes. As a continuous model, DSM or DTM can be represented by a continuous function $z = f(x, y)$, where z is the elevation in terms of the location (x, y). The Gaussian (linear) scale-space representation of $f(x, y) = f^{(k)}$ is a family of derived continuous representations. In the proposed method, a modified version of this model is applied, and a serious pyramid levels (resampled images) for the presentation of digital surface terrain is employed to avoid a loss in data accuracy.

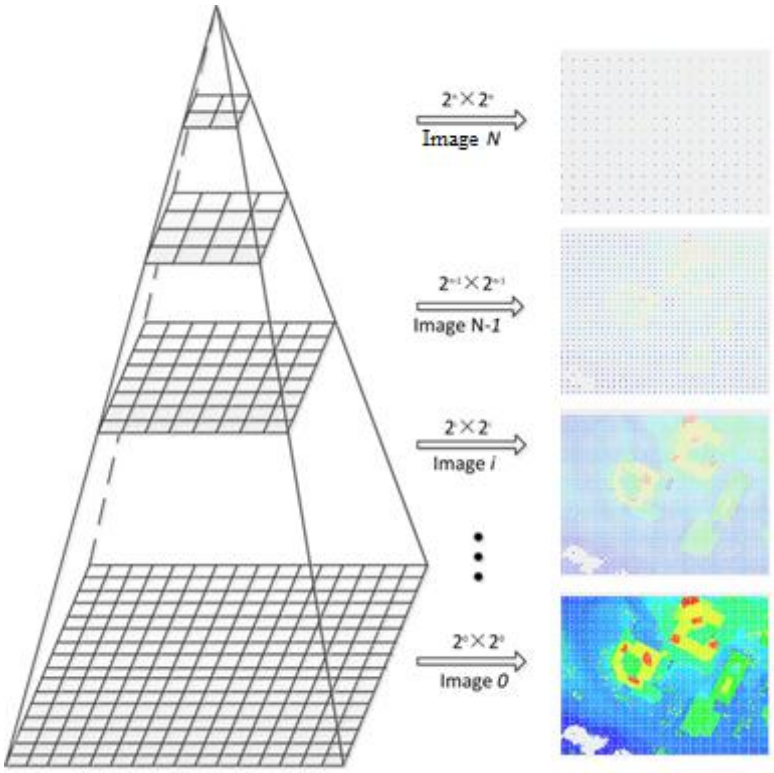


Figure 3-8 Illustration of the data pyramid (images are resampled from Site A)

First, the bottom level $f^{(0)}$ is a raster resampled from the original LiDAR point clouds. The size of the cell (grid) is a user assigned parameter *K* (unit cell size), which should larger than the point density of the LiDAR data. Since there may be more than one point in a cell, the lowest point in the cell represents the cell, which means that the value of each cell is the elevation value (z) of the lowest point in the cell.

Then, any level $f^{(k)}$ from the second level to the top level are resampled from the bottom level $f^{(0)}$, the cell size $g^{(k)}$ and the value of the cell $f^{(k+1)}(x, y)$ are defined in Equations **(3.2**) and **(3.3**).

$$g^{(k)} = Kt^k \quad \textbf{(3.2)}$$

Where, *K* is the aforementioned unit cell size; t is the variance of the Gaussian function, which indicates the scale level being defined. The more the *t* value increases, the further the resolution of the original continuous terrain model is decreased. In other words, a coarser-resolution model $f^{(k+1)}$ is basically a re-sampled representation created at a lower resolution than the finer-resolution model $f^{(k)}$. In this study, it assumes that the scale t is 2; which means that the cell size of $f^{(k+1)}$ is half of that of $f^{(k)}$, as can be seen in **Figure 3-8**.

$$f^{(k+1)}(x, y) = D(x, y; t) = G_t(x, y) f^{(0)} \quad \textbf{(3.3)}$$

Where, *D* represents one cell in $f^{(k+1)}$ and is defined by the convolution of $f^{(0)}(x_0, y_0)$ and the bivariate Gaussian probability density function $G_t(x, y)$ (Ali, 2010). Although the

definition of D works for scales $t \geq 0$, only a finite number of levels in the scale-space representation would be considered. There are two considerations in the determination of the number of images in the data pyramid: the point density (d) and the maximum estimation of building sizes (B_s). To guarantee the lowest point in a grid cell as terrain point, the top level of pyramid should have the cell size ($g_{\max}$) equal to or larger than B_s. Meanwhile, to minimize the loss of accuracy, the bottom level should keep the cell size ($g_{\min}$) close to the point density d. Like it, a series of multi-scale images (level) are generated. In this way, most of small unwanted objects are gradually excluded from bottom to top levels in the pyramid.

3.3.2 Identification of Terrain and Off-terrain Points

According to the given maximum building size (B_s) and the point density (d), *N* levels of a data pyramid are generated from bottom to top (from finer to coarser in scale, 0, 1, 2… N). The top level (f^N), also called as the coarsest scale level, is considered an initial DTM reference by reason that the condition of $g_{\max} > B_s$ theoretically guarantees that its cells have higher possibility of being a terrain point than cells in finer scale levels. Starting from the initial DTM reference, it iteratively search terrain points and recover DTM from coarse-to-fine scalar levels.

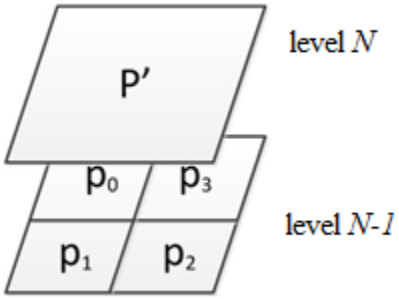


Figure 3-9 Cells in two levels

Assuming that Level N is a terrain reference, the identification of terrain points starts from Level N-1, as shown in **Figure 3-9**. Due to the scale of 2, one cell in Level *N* corresponds to four cells of Level *N-1*. According to the identified terrain point *P'* in Level N, un-identified points p_0-p_3 in Level *N-1* will be labeled using two criterion: layering information and slope calculation between the known terrain point *P'* and unknown points p_0-p_3. The identification of terrain points is stated in the following two steps:

1) Compare layer number between *P'* and p_i (i=0-3):

Label p_i as terrain point if they have the same layer number;

Otherwise, go to the next step;

2) Calculate the slope between P' and p_i: by **(3.4)**

Label p_i as terrain point If the slope is smaller than a given slope threshold $Tan\theta$;

Else, label p_i as off-terrain point, and remove it;

$$Tan\theta = \frac{|Z' - Z_i|}{\sqrt{(X' - X_i)^2 + (Y' - Y_i)^2}} \quad \textbf{(3.4)}$$

Where, (*X', Y', Z'*) and (X_i, Y_i, Z_i) are the coordinates of P' and p_i respectively.

The algorithm repeats Step 1- 2 for each point in the terrain reference to find terrain points in the processed image until no point is left. After that, an interpolation is used to fill holes where off-terrain points exist.

The previous identification works well in a flat area, but this is not the case with topographic areas. Since the value of each cell is always from the lowest point in that cell, the four layer numbers in the lower level are the same as the reference level, or higher than reference level. If there is a relief in the area, the higher layer number may still represent the terrain points, but the cell will be wrongly identified into off-terrain points. Therefore, a tolerance threshold is needed in identification, especially at high levels.

$$T = \text{INT}\ (\ \textit{layer Number / Identification Tolerance}\ ) \quad \textbf{(3-5)}$$

Where, *T* represents the biggest tolerable difference threshold between the layer numbers of the identifying cell and the reference cell. And *Identification Tolerance* is a user assigned parameter (see Section 4.4.5).

3.3.3 Interpolation at Off-terrain Points

As discussed above, the terrain identification in Level *N-1* is based on the terrain reference of Level *N*. For Level *N-1*, there are holes due to the removal of off-terrain points, and there are some cells with no representative point as well. To be a terrain reference for the next scalar level, those holes must be interpolated. Commonly used interpolations include Inverse Distance Weighted (IDW), spline, Kriging, etc. (Liu, 2008). Among them, the IDW interpolation is intuitive and efficient (Anderson, 2010); it works best with evenly distributed points which can be supplied by the multi-scale levels. Thus it was applied in this thesis study. IDW assumes that each point has an influence to the prediction point which diminishes with distance. In other words, the closer the point to the estimated location, the more weight it will hold, as shown in **Equations (3-6)** and **(3-7)**.

$$F(x,y) = \sum_{i=1}^{n} w_i f_i \qquad \textbf{(3-6)}$$

$$w_i = \frac{h_i^{-2}}{\sum_{j=1}^{n} h_j^{-2}} \qquad \textbf{(3-7)}$$

where, *F(x,y)* is the interpolated value of the target point, f_i is the height value of a neighbor near to the target, w_i is the neighbor's weight, and *n* is the number of neighbors. **Equations (3-6)** and **(3-7)** gives the way of calculating weight w_i, and *h* is the distance from each neighbor to the target point. This process is effective for dense and evenly

distributed sample points. However, the uneven distribution or low density of the points will lead to a not continuous interpolated surface. Therefore it is difficult to predict these areas. It is not possible for the weighted average method to estimate outside the data range (Liu et al., 2007). However, if there are a lot of data points and complex terrain this is a very effective method. Considering LiDAR points being denser and relatively even distributed pattern this IDW method is effective.

Figure 3-10 shows the terrain filtering result is from coarse level to fine level. After the interpolation in Level 0, the generated image is a rough DTM.

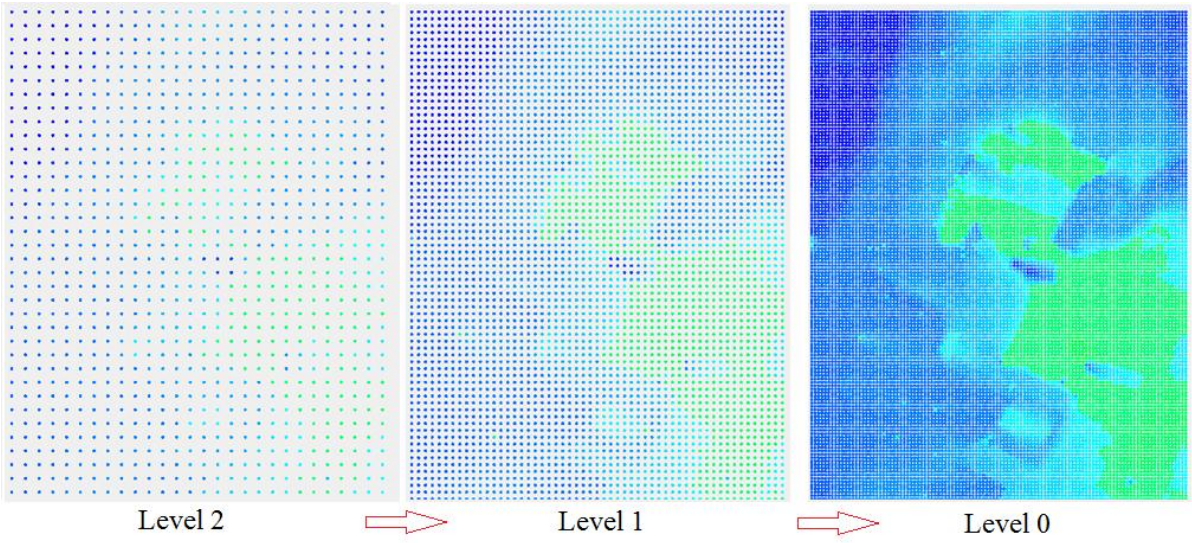


Figure 3-10 A level with interpolation can be used as a reference in the next level.

3.4 DTM Refining

All terrain filtering from Image *N-1* to Image *0* is carried out to finally obtain the terrain reference. As a result of that those pseudo-gridded levels are generated from the lowest

points, there are non-lowest points that belong to terrain points are missed. Meanwhile, many terrain points in the final terrain reference are interpolated, which generate differences or errors between real laser points. Therefore, in order to improve the accuracy and generate a high resolution result, a refinement of the rough DTM is required.

As shown in **Figure 3-11**, in the developed method the refinement of the rough DTM is done by three steps: adjusting the result from nDSM, filtering the original LiDAR point cloud data based on the refined DTM, and applying the separated terrain points to generate the final DTM through another IDW interpolation.

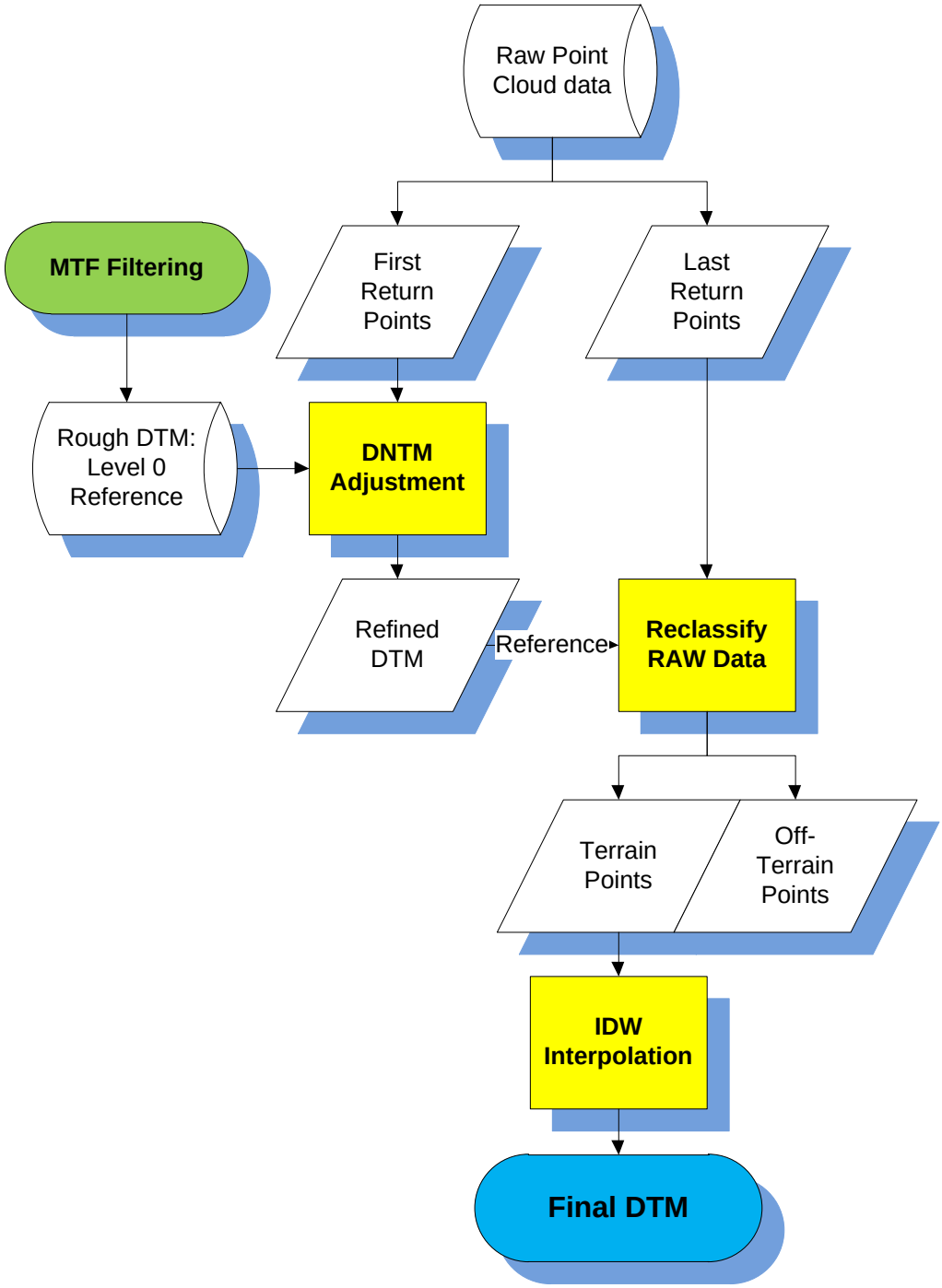


Figure 3-11 Flowchart of DTM Refining

During the interpolation, features like pits, curb and hill peaks may be wrongly removed and interpolated. This will make some cells in the generated rough DTM have higher

elevation values than the real terrain, and the number can be even higher than DSM in the according location. A normalized Digital Surface Model (nDSM) is introduced here for this problem (Hu and Tao, 2005). The nDSM is generated by subtracting the DSM from the DTM, and represents the net height of the off-terrain objects, such as buildings, vegetation, as shown in **Figure 3-12**.

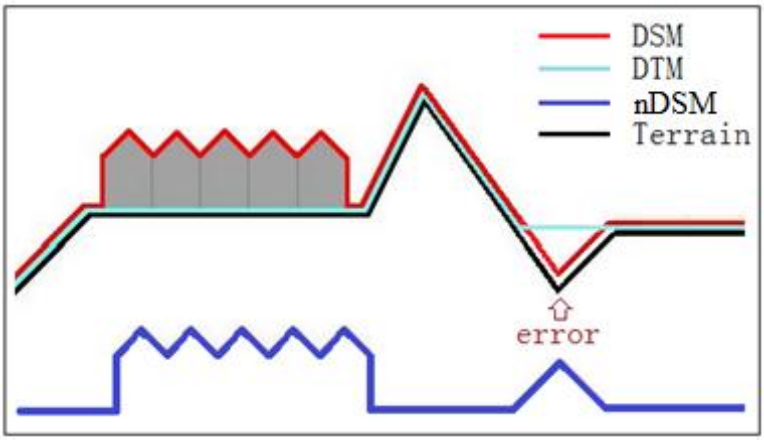


Figure 3-12 nDSM = DSM - DTM

In this method, DSM is represented by the original LiDAR first return points data, and the DTM is the product generated by the Multi-scale Terrain Filtering. As the representation of off-terrain objects, the nDSM usually has a positive value. Therefore the negative values of nDSM show where the errors are. And the method will replace these errors by the according value of the original data to generate the refined DTM.

Finally, the refined DTM generated from nDSM feedback adjustment is used as reference to separate the original last return points into ground points and non-ground points based on their layer numbers. Then a final IDW interpolation of these separated ground points

will generate the final DTM.

3.5 Chapter Summary

In this chapter, a new DTM generation method from airborne LiDAR point clouds has been proposed. This method has three steps: data pre-processing, Multi-scale Terrain Filtering (MTF), and refinement of the rough DTM. In pre-processing step, the layering part was used to generate layer numbers which are used as a criterion in the identification of the MTF, and also used to remove noise and outliers. In the MTF step, identification and interpolation were carried out in a group of rasterized pyramid levels, and coarse to fine DTMs are generated. In the refining step, the generated rough DTM is refined by nDSM adjustment and used to filter the data into terrain points. An IDW interpolation carries out by these points and produces the final DTM.

Chapter 4. Experiments and Results

In this chapter, the datasets featured in this thesis are listed in Section 4.1. The accuracy evaluation methods are introduced in Section 4.2. The result evaluation and comparison of the proposed method and other existed methods are discussed in Section 4.3. And Section 4.4 describes the parameters applied in the experiments and a sensitivity analysis of these parameters, followed by the conclusion in Section 4.5.

4.1 Data Sources

Three datasets are included in the experiments to verify the proposed filtering algorithm. The first dataset obtained from Stuttgart, Germany was released by ISPRS working group WG III/3, have been made available through the society's web site (www. commission3.isprs.org/wg3/). The second set, required from Toposys GmbH, Germany, is located in Mannheim, Germany. The last dataset generated by Optech covers Waterloo area, southern Ontario in Canada. The first 15 sites from ISPRS are selected to test the performance of the MTF algorithm and compare the results with other methods evaluated by ISPRS (Sithole and Vosselman, 2004). To extend the tests for further verification, two extensive experiments are added to further verify the robustness and stability of the MTF algorithm. Error computation for DTM generation is made, for the accuracy assessment, using several complex terrain samples with dense vegetation.

4.1.1 ISPRS Data

The ISPRS working group WG III/3 has tested a number of algorithms developed in the past (Sithole and Vosselman, 2004). ISPRS collected the data captured by an Optech ALTM scanner. The reference data was manually generated from the data with the reference of the aerial image and landscape. These data are located along seven study sites over the Vaihingen test field and Stuttgart city center. The study cites have varied terrain characteristics and diverse feature content (e.g., open fields, vegetation, buildings, road, railroads, rivers, bridges, power lines, water surface, among others). **Table 4-1** contains a list of the study sites. An aerial image of Forest Site 5 is shown in **Figure 4-1**.

Figure 4-1 Aerial image of ISPRS data Site 5

Table 4-1 Features of the ISPRS dataset (Sithole and Vosselman, 2003)

Test Site	Reference Sample	Number of Points	Terrain Points	Off-terrain Points	Width (m)	Length (m)	Density (points/m^2)
City Site 1	Samp11	38010	21786	16224	133.89	302.73	0.94
	Sampl2	52119	26691	25428	204.38	264.22	0.97
	Features: A mixture of vegetation and buildings on steep hillside, data gap.						
City Site 2	Samp21	12960	10085	2875	123.79	115.19	0.91
	Samp22	32706	22504	10202	187.87	181.23	0.96
	Samp23	25095	13223	11872	146.18	205.9	0.83
	Samp24	7492	5434	2058	121.86	72.44	0.85
	Features: Large buildings, irregularly shaped buildings, road with bridge and small tunnel, data gap						
City Site3	Samp31	28862	15556	13306	174.17	161.94	1.02
	Features: Densely packed buildings with vegetation, data gaps.						
City Site4	Samp41	11231	5602	5629	167.19	104.71	0.64
	Samp42	42470	12443	30027	227.12	202.98	0.92
	Features: Railway station with trains (low density of terrain points), data gaps.						
Forest Site5	Samp51	17845	13950	3895	232.41	429.87	0.18
	Samp52	22474	20112	2362	450.01	301.12	0.17
	Samp53	34378	32989	1389	430.42	472.93	0.17
	Samp54	8608	3983	4625	185.84	267.49	0.17
	Features: Steep slopes with vegetation, quarry, vegetation on river bank, data gaps						
Forest Site6	Samp61	35060	33854	1206	504.23	443.97	0.16
	Features: Large buildings, roads with embankments, data gaps.						
Forest Site7	Samp71	15645	13875	1770	394.83	221.12	0.18
	Features: Bridge, underpass, roads with embankments, data gaps.						

This dataset is widely adopted by the laser scanning researchers (e.g. Meng et al., 2009; Shao and Chen, 2008). It covers many different land features and filtering difficulties. However, it does not contain small woods and residence in urban area. And the reference data is only available for the 15 samples; the reference data for entire site is not available,

which means will limit the algorithm testing for large site. In order to complement these defects, two other datasets are tested.

4.1.2 LAS LiDAR Data in the City of Waterloo

In this thesis, the city of Waterloo is chosen to be a study area. In the city of Waterloo, especially the area close to UW campus, multiple features of the land use can be found. This gives the convenience to test the algorithms in different scenario, such as big buildings in UW campus, residence close to UW campus, as well as some small forest nearby. These three different scenarios will be discussed in this thesis.

In this data, the algorithm and analysis are applied to the LiDAR data in a LAS format which is described before. The raw LiDAR dataset covers the main campus of University of Waterloo (UW), Waterloo, Ontario which was acquired by Optech's Airborne Laser Terrain Mapper (ALTM) on March 11, 2006. The average flying height was 1,200 m above ground level and the flying speed was 66.9 m per second. The scan angle was 20 º. The desired resolution was 0.908 m. The formation of raw data was the point cloud which contains more than seven million points. The dataset is demonstrated in **Figure 4-2**.

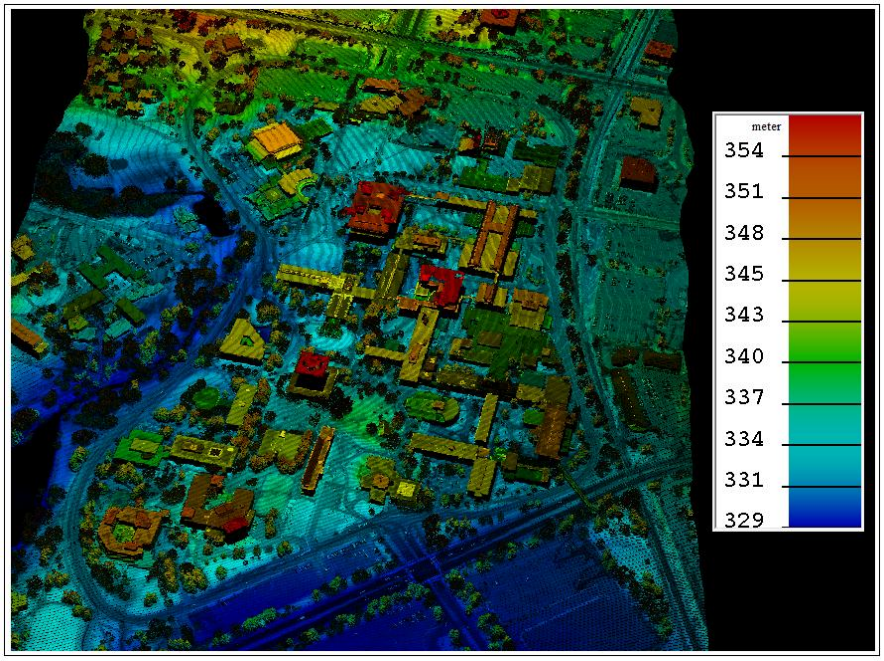


Figure 4-2 Original LiDAR data of the UW campus

The geo-reference of LiDAR point clouds is demonstrated in **Table 4-2**:

Table 4-2 Waterloo Data Specifications

Type	QTC Point Cloud
Comp	Uncompressed
Points	7,997,153
Width	1,587m
Height	7,139m
Scale	0.8403
Density	1.4164 point/m^2

Based on the same LAS LiDAR dataset of Waterloo a manual classification in UW campus is done by a qualified person with the proper knowledge and experience. The ground points are applied in an IDW interpolation for DTM generation. The interpolation is "*using the default 12 points for the "Search Radius Settings" and "Power" of 2… The*

cell size was set to 1" (Lackner, 2010). To somehow avoid the unrealistic interpolation under some buildings "*a 2-meter buffer around the building footprints and calculate the mean elevation from the ground LiDAR points that fall between the 2-meter buffer line and the building footprint*" is created.

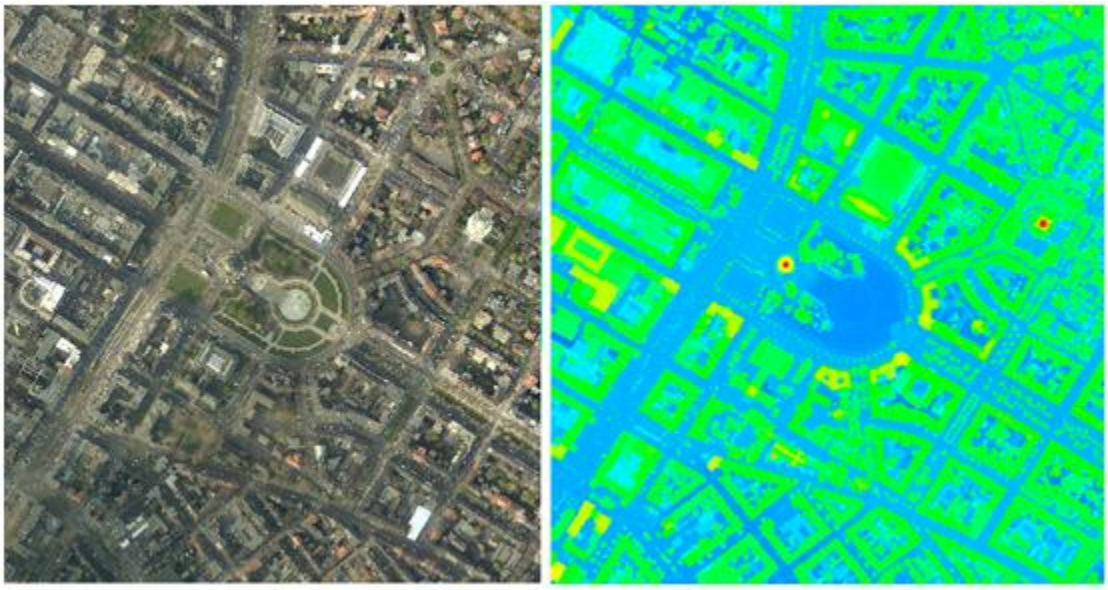
This city of Waterloo dataset contains many different topographic features, such as the university campuses, rivers, forest, parks, farmland, suburban area, residence area, etc. These areas are ready to be experimented and can be shown in a visible format. However its reference data is an interpolated raster TIF data whose resolution is lower than the original point clouds data. A higher error rate of the quantitative evaluation results in the experiments is expected.

4.1.3 TopoSys Demo City Data

Another LiDAR dataset is required from TopoSys GmbH, Germany. This dataset covers the area of Mannheim, Germany. Raw data, CIR-3layer, RGB-3layer, intensity, DSM first return, DSM last return, TIN, Contour lines are all available in this dataset, and the DTM reference data is included as well. The average spacing between points is available in 50cm, 100cm, and 200cm. **Table 4-3** shows the geo-reference of the data used in this research, and **Figure 4-3** shows two images of this dataset.

Table 4-3 Demo City Data Specifications

	DSM-First Echo	DSM-Last Echo	DTM-Interpolated Last Echo
Format	ASCII		
Comp	Uncompressed		
Width	799.50m		
Height	859.50m		
Points	2,748,790	2,738,623	1,399,214
Density	4.0002 point/m^2	3.9854 point/m^2	2.0362 point/m^2



(a) Optic image (b) LiDAR first return points

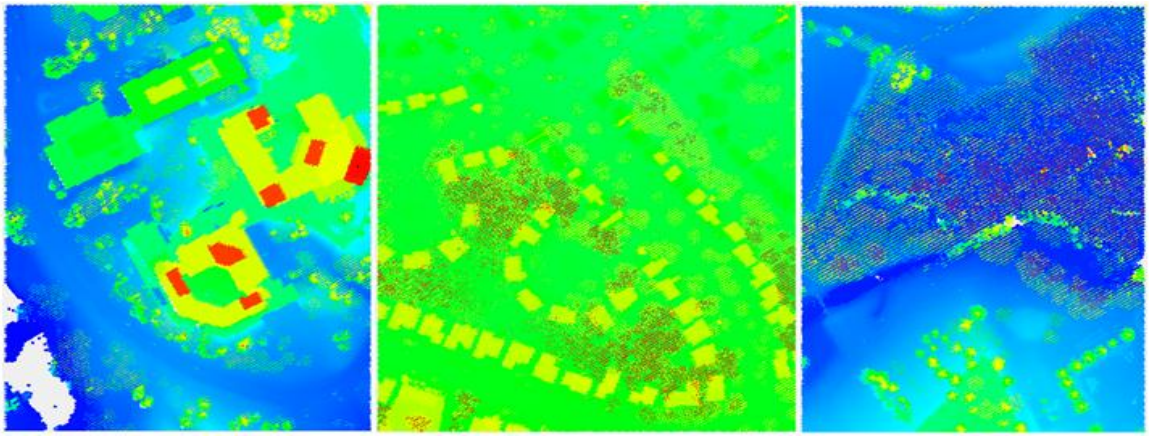
Figure 4-3 TopoSys Demo City data

This dataset is in an ASCII format. The majority of the data is in urban areas, which includes some areas of big buildings and some joined town-house residence areas. Buildings and roads are evenly distributed, and a big square is in the center of the data. Since the reference terrain points for the entire data is available, this data will be used in testing the performance of the proposed method in large site.

4.1.4 Complement Experiment Sites

As the original raw LiDAR data is very large, and the data covers a big area, which contains many different types of land-use. To process and analyze such big and complex data is very difficult and time consuming. Therefore, six experiment sites (additional to the 15 ISPRS samples) are selected from the Waterloo data and the Demo City data. In order to test the performance in sites with different features (complex building area, residence area, forest area and urban city area), the location and size of the test sites are sampled by visually selecting from optic images.

Figure 4-4 (a), (b) and (c) are Sites A, B and C respectively, these three sites are selected from the City of Waterloo data. **Figure 4-4** (d), (e) are Sites D and E, which are from the TopoSys Demo City data. **Figure 4-4** (f) shows the entire Demo City data. The dimension and the some information of experimental sites are listed in **Table 4-4**.



(a) (b) (c)

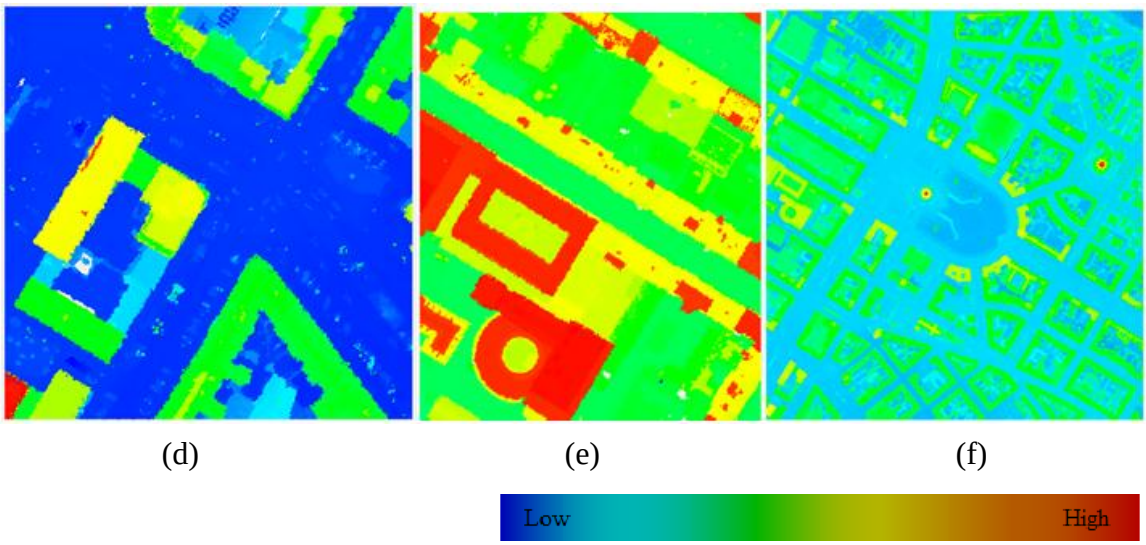


Figure 4-4 Testing sites: a) a corner of UW campus with big buildings; b) a sample of residence area; c) a sample of forest area; d) and e) parts of the Demo City data; f) entire Demo City data. The color represents elevations as the bar shows.

Table 4-4 Experiment Site Specification

Site	Area	Number of Points	Width (m)	Length (m)	Area (m^2)	Density (points/m^2)
A	Campus	87,640	218	285	62,130	1.4106
B	Residence	132,697	248	282	69,936	1.8974
C	forest	84,732	182	255	46,410	1.8257
D	Urban	89411	150	149	22,350	4.0005
E	Urban	135016	169	199	33,631	4.0146
F	Urban	2738623	799	859	686,341	3.9902

Site A is a corner of UW campus, which includes buildings and part of Ring Road, a water body and some trees and other vegetation. The buildings in this site are big and many of them are connected to each other. The big court yards are surrounded by buildings. The vegetation and parking lots are located around the buildings as well. Site B is in a residence area north of UW campus. There are around 50 houses in that site. Some of the houses are close to trees with large canopies which cover some part of the houses.

The landform is that west south is lower than north east. Site C is in the Waterloo Park, which has part of a forest, a little section of road, and lawn in the public park. And there are no buildings in that area. Site D is in urban area which is part of the TopoSys Demo City dataset. Big buildings, court yard surrounded by the buildings, roads, cars on the road, trees along the roads and buildings, and small areas of vegetation are included in this site. Site E is also in urban area in Demo City data. Big buildings, court yard and parking space surrounded by the buildings, roads, cars on the road are included in this site. Consequently six samples with different features are selected to complement the ISPRS data. Since the formats of the reference data in three datasets are different, the following discussion will be around the method evaluating the performance of the proposed method on samples from different datasets.

4.2 Accuracy Evaluation Method

In accuracy assessment methods, visual inspection, random sampling of filtered data, and cross tabulation are three main categories (Meng et al., 2010). Visual inspections are usually utilized when the reference data is not available, and can be used to manually detect obvious errors. However, it is hard to find out the low objects such as bushes, road curb. In this paper, the visual inspection is to analysis type I, type II errors based on visualized cross-matrix. Random sampling of the data works based on an assumption that the errors or bias are evenly distributed. However, the filtering errors usually founded

where the features hard to recognize, which are usually not evenly distributed in data. But in order to assess the generated DTM, a sampling estimation is adopted. The cross tabulation are also adopted to analyze and compare the identified terrain points.

4.2.1 Cross-matrix Analysis

Sithole and Vosselman (2003b) reviewed and compared eight filtering algorithms, and their comparing method and data are frequently cited and applied in many researches of the laser scanning data filtering (Briese, 2010; Pfeifer and Mandlburger, 2008; Meng et al., 2010; Liu, 2008). There are multiple quantitative assessment criteria available, such as Circular Error of 90% (CE90), Linear Error of 90% (LE90), Root Mean Squared Error (RMSE) and 1-Sigma (GeoVAR, 2012). However, in order to compare the results with other algorithms, this paper adopts confusion matrix approach and kappa indices to quantitatively test the performance of the MTF method due to the result availability of other algorithms.

Table 4-5 Cross-matrix

Reference	Filtered	
	Terrain	Off-terrain
Terrain	a	b (type I error)
Off-terrain	c (type II error)	d

$$\text{Accuracy} = (a + d) / (a + b + c + d) \quad (4\text{-}1)$$

$$\text{type I error} = b / (a + b) \quad (4\text{-}2)$$

$$\text{type II error} = c / (c + d) \quad (4\text{-}3)$$

According to Sithole and Vosselman's (2003b) assessment method, the cross-matrices are applied in this research to quantitatively analysis the type I, type II error and their relationship. type I errors are the errors which wrongly identified terrain points as off-terrain points, and type II errors are the errors which wrongly identified off-terrain points as terrain points as shown in **Table 4-5**. The formulas of their calculation are listed on **Equations** (4-1)**,** (4-2) and (4-3) and a, b, c and d refers to **Table 4-5**.

Visualized cross-matrices are also provided to determine the locations where it happens to be type I and type II errors in order to qualitatively analysis the nature of the errors. A typical visualized cross-matrix is shown in **Figure 4-5**.

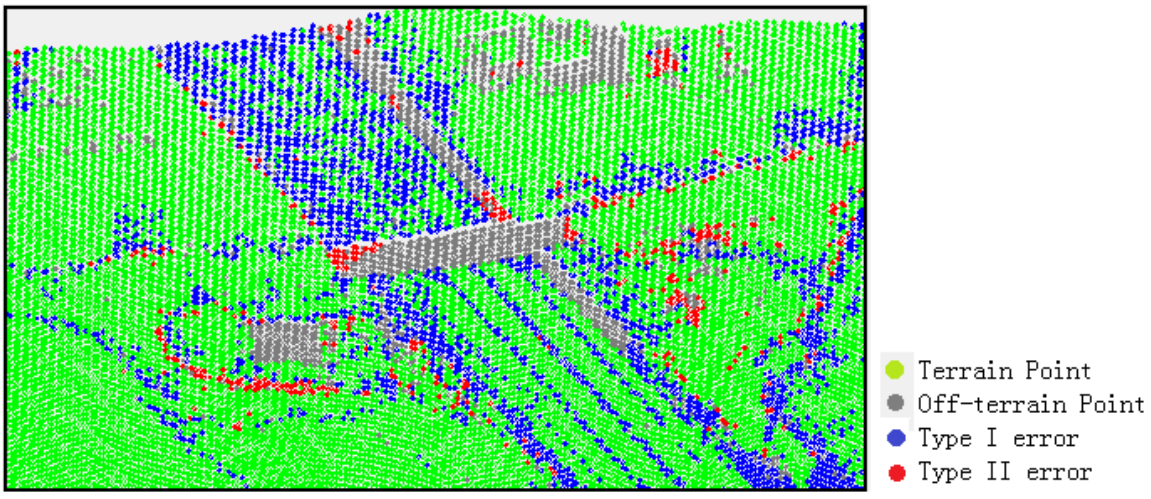


Figure 4-5 An example of visualized cross-matrix (sample 71)

Except type I errors, type II errors and overall accuracies which are mentioned by Sithole and Vosselman (2003b), the kappa Index of Agreement, which takes agreement occurring

by chance into account and is generally considered as a more robust measure than percent agreement (Strijbos et al., 2006) is also adopted in this thesis. And the formulae are listed on **Equations** (4-4), (4-5) and (4-6). And a, b, c and d refers to **Table 4-5**.

$$\text{Kappa} = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{4-4}$$

$$\Pr(a) = \frac{a + d}{a + b + c + d} \tag{4-5}$$

$$\Pr(e) = \frac{a + b}{a + b + c + d} \times \frac{a + c}{a + b + c + d} + \frac{c + d}{a + b + c + d} \times \frac{b + d}{a + b + c + d} \tag{4-6}$$

Where, **Pr(*a*)** is the relative observed agreement among raters. **Pr(*e*)** is chance agreement for the hypothetical probability. It is adopting the observed data to calculate the probabilities of each observer randomly saying (Strijbos et al., 2006); a, b, c and d refers to **Table 4-5**. Kappa measures and considers the agreement between two raters, which can more reflect the performance of the method on both type I, type II errors.

The reference data of the ISPRS data and the Demo City data are in a point cloud format. This format gives the access to compare each point to a reference point, which offers an easier way to check the type I and type II errors and make the cross-matrices. This evaluation method is good to evaluate the performance of filtering method.

4.2.2 Sampling Estimation

The final products of this thesis are DTMs; the accuracy rate of the filtered points cannot

directly reflect the accuracy of the generated DTM. And the reference data of the City of Waterloo data is in a raster format stored in TIFF files, which requires some adjustment to make the cross-matrices. However the choosing of the buffer parameters for the adjustment is not very intuitive. Therefore, in order to directly evaluate the generated DTM and to apply the raster reference, an accuracy evaluation based on the sampling and comparing the interpolated DTMs is adopted as a compliment to the cross-matrix evaluations.

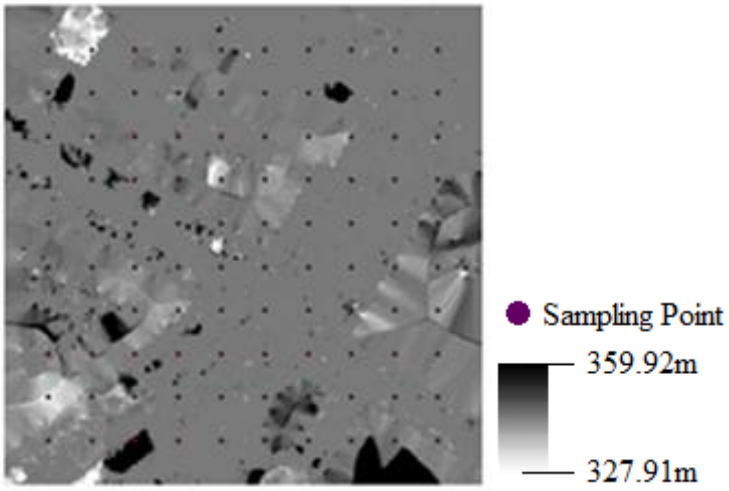


Figure 4-6 A demonstration of sampling point (10x10 points)

The sampling evaluation compares the samples from the result image and the samples with same x, y coordinators from the reference image to assess the accuracy of the result. The selected sampling points are evenly distributed on the study sites, an example is shown in **Figure 4-6**. The number of the samples is determined based on the size and point density of the study site. The sample values of the result are extracted from the DTM interpolated from the identified terrain points, and the sample values of the

reference are extracted from the raster reference data (the raster reference data of the Demo City data is interpolated from its reference terrain points). The offset (difference) between the result value and the reference value extracted on sampling points are calculated for every point. The evaluation results such as average error, standard deviation, and the worst error are calculated based on these offset values. The flowchart is shown on **Figure 4-7**.

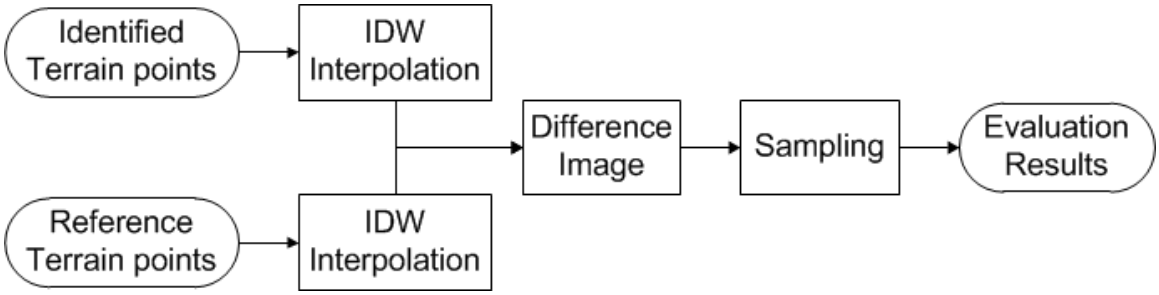


Figure 4-7 Flowchart of the sampling estimation

4.3 Result Evaluation and Analysis

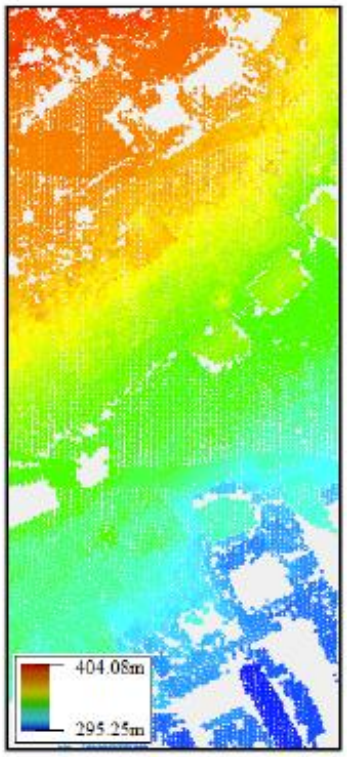
In this section, the proposed MTF algorithm is applied on the samples from three datasets. 15 sample sites from ISPRS are evaluated by the cross-matrix approach; the results are compared to results of eight methods provided by ISPRS. Six sites from the City of Waterloo data and the Demo City data are assessed by the sampling estimation; their results are compared with two methods included in software ALDPAT. In order to clearly demonstrate the process of the experiments, the parameters used in this experiment are listed in **Table 4-6**. And **Figure 4-8** shows the result of the sample sites.

Table 4-6 Descriptions of Parameters

Parameter	Description	Applied Stage
Width	histogram width	Pre-processing Layering
Del	tolerated percentage in histogram layer generation	Pre-processing Layering
Min Layer	minimum number of points in a layer	Pre-Processing Noise Elimination
Min Cell	minimum number of points in a cell	Pre-Processing Noise Elimination
K	basic cell size in rasterized pyramid level generation	ITR Levels Generation
Level num	number of rasterized pyramid levels	ITR Levels Generation
Tanθ	angle threshold	ITR Identification
Identification tolerance	tolerated number of levels in iterative identification	ITR Identification
Classification tolerance	tolerated number of levels in classification	DTM Refining Classification

4.3.1 Quantitative Analysis

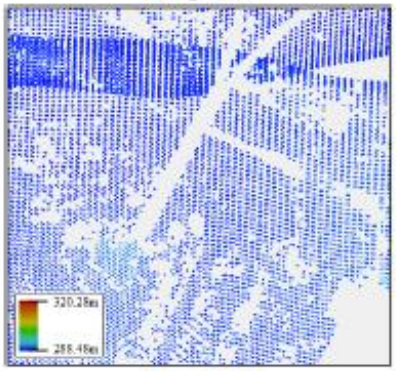
The 15 sample sites acquired from ISPRS are selected on city and forest areas, the ground features such as slope gradient, vegetation density are various. Therefore, Meng et. al. (2010) divided the fifteen ISPRS study sites into three groups. The sites in the first group (Sample 11, 24, 41, 54) have rough slope and dense vegetation; the sites in the second group (Sample 12, 21, 22, 23, 31, 42) are relatively flat urban area; and the sites in the third group (Sample 51, 52, 53, 61, 71) contain rough terrain and discontinuous (e.g. river banks and mining fields). The following discussion will refer to these three groups.



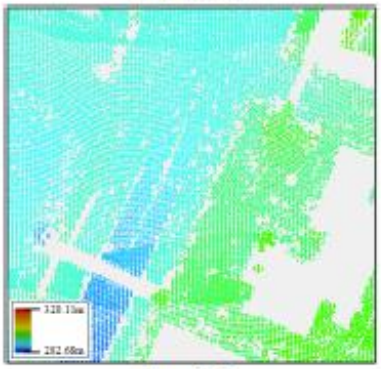
Sample 11

Sample 12

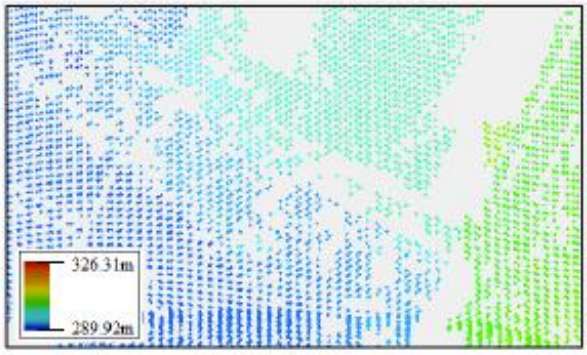
Sample 23



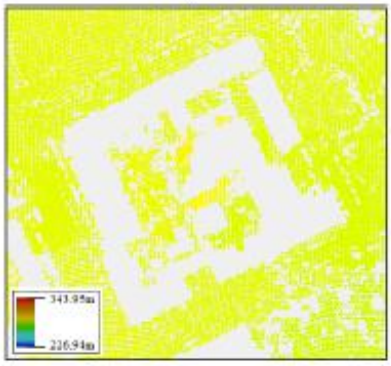
Sample 21



Sample 22



Sample 24



Sample 31

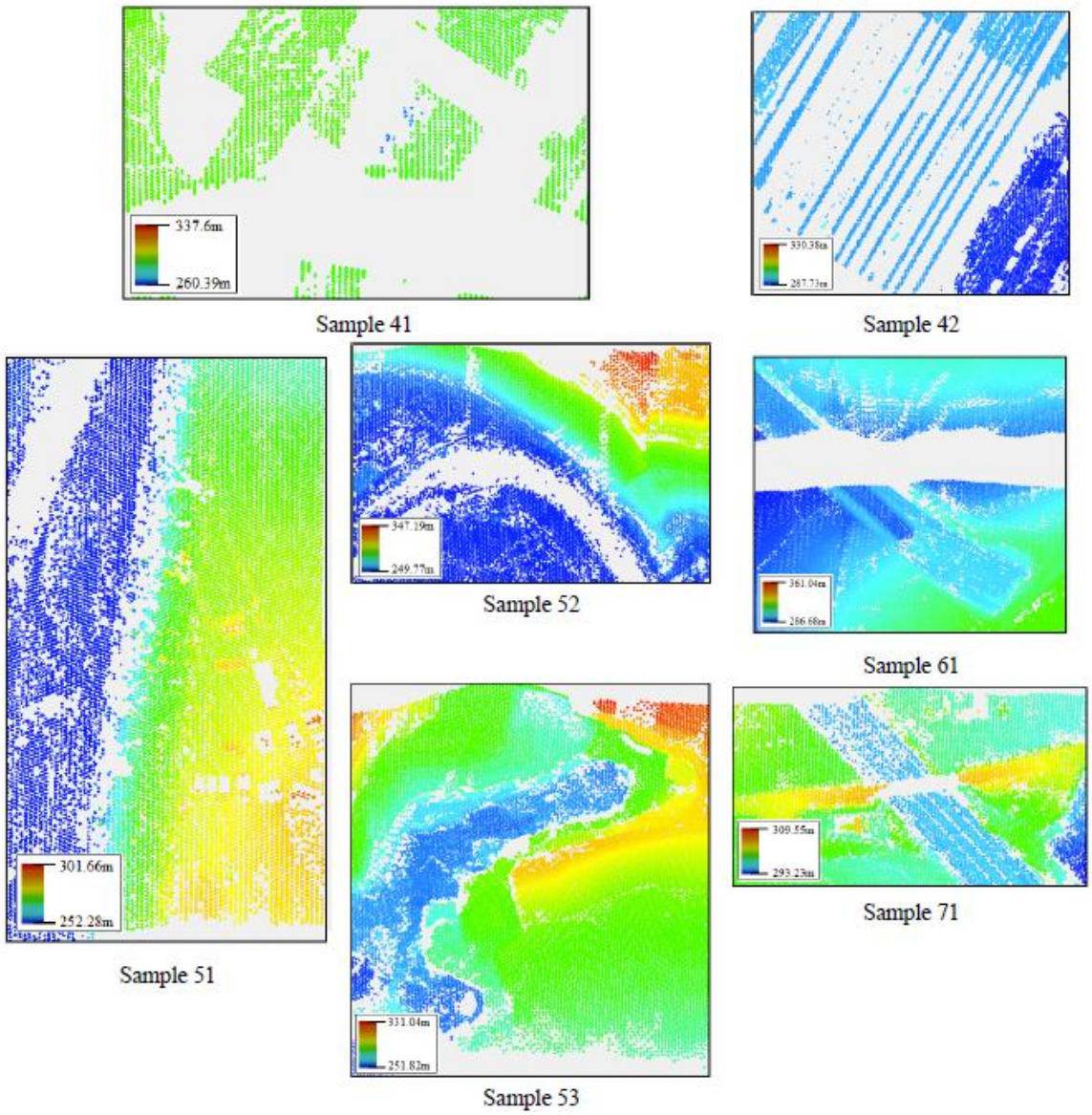


Figure 4-8 Result: identified terrain points in 15 ISPRS sample sites

Figure 4-9 shows the quantitative assessment results of the fifteen sites, while **Table 4-7** lists the parameters used to generate these results. The overall accuracy and kappa coefficient for one site may be required from tests with different combination of parameters, e.g. "samp11k" in **Table 4-7** refers to the parameter combination for the kappa

coefficient value of Sample 11 shown in **Figure 4-9**, the numbers of type I, type II errors and Accuracy for Sample 11 are generated by the parameters listed as “samp11”.

Table 4-7 Parameters of Multi-scale Terrain Filtering method

Sites	Width (m)	Delta	K (m)	Number of Levels	Tanθ	Identification tolerance
samp11	2.1	0.6	1.6	7	0.4	0.5
samp11k	2.6	0.6	4.0	6	0.7	0.5
samp12	0.9	0.6	2.0	7	0.7	0.5
samp21	0.9	0.6	2.0	7	0.7	0.5
samp22	0.9	0.6	2.0	7	0.7	0.5
samp22k	1.5	0.6	1.6	7	1.2	0.5
samp23	0.9	0.6	2.0	7	1.2	0.5
samp24	0.9	0.6	2.0	7	1.2	0.5
samp31	0.9	0.6	2.0	7	0.7	0.5
samp41	0.9	0.6	2.0	7	1.8	0.5
samp42	0.9	0.6	2.0	7	0.7	0.5
samp51	0.9	0.6	2.0	7	0.7	0.5
samp51k	2.0	0.6	2.4	7	0.7	0.1
samp52	0.9	0.6	2.0	7	0.7	0.5
samp52k	4.0	0.5	3.0	6	0.7	0.8
samp53	0.9	0.6	2.0	7	0.7	0.5
samp53k	3.2	1.5	2.4	7	3.0	0.5
samp54	2.7	0.2	2.4	7	0.4	0.5
samp61	0.9	0.6	2.0	7	0.7	0.5
samp71	0.9	0.6	2.0	7	0.7	0.5

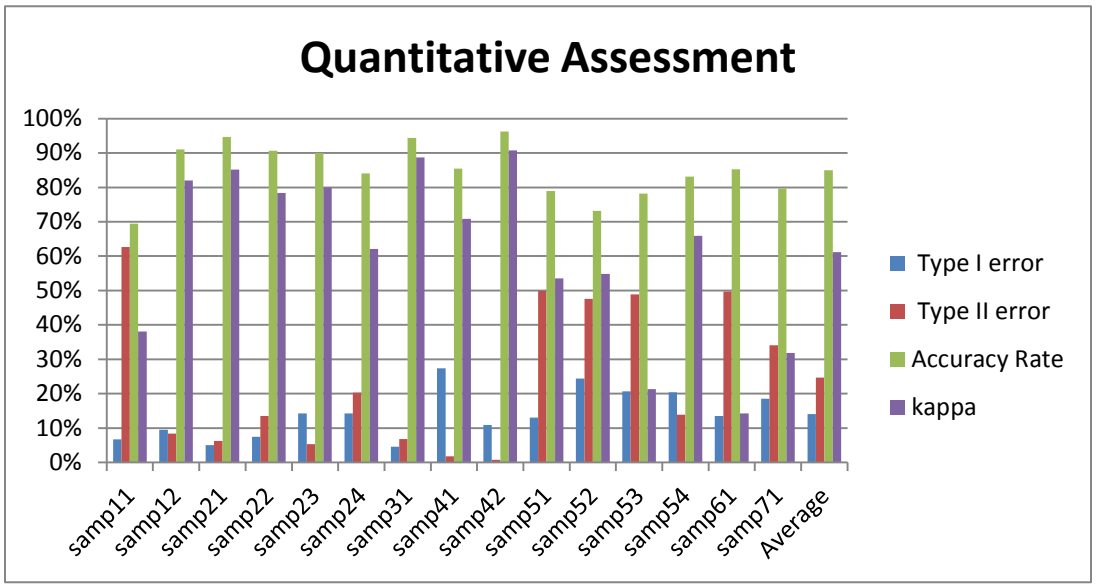


Figure 4-9 Type I errors, type II errors, Accuracy Rates and Kappa coefficients of the 15 sample sites from ISPRS tested by the proposed MTF method

As shown in **Figure 4-9**, the average, best, worst values of the accuracy rate are 85%, 96% and 70% respectively. The standard deviation of the accuracy rate and kappa coefficient in fifteen sites are 8% and 25%, which means the overall accuracy is relatively stable while the kappa coefficient varies depending on the study sites. However, the parameters in the proposed MTF method have to be tweaked to obtain the best results during the finite number of experiments, and the optimal result is not guaranteed in these experiments. In order to analysis the performance of the proposed MTF method on different situations, a series of comparisons are carried out as follows.

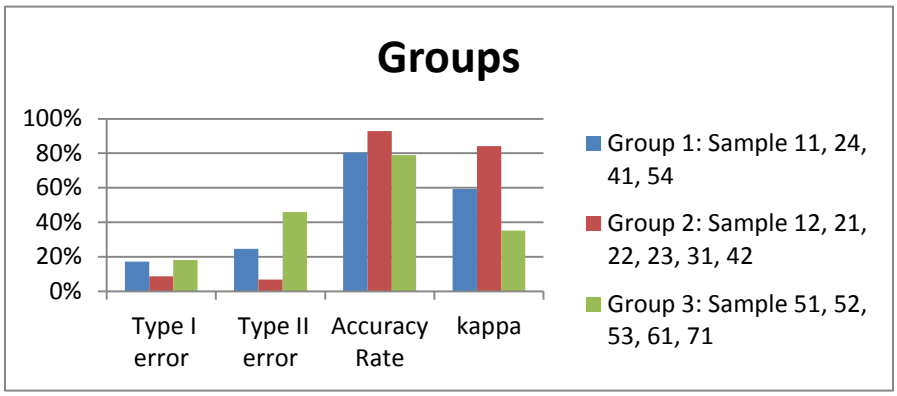


Figure 4-10 Average values of type I, type II errors, Accuracy and Kappa sorted by three groups

Since the fifteen sample sites are divided into three groups, a comparison is shown in **Figure 4-10**. Group 2 sites shows the lowest errors and highest accuracy rate and kappa coefficient, which means that the MTF method can handle Group 2 sites better than the other two groups, this number is also good enough to compare with filters compared by ISPRS (Sithole and Vosselman, 2003b). The performance of the MTF method on Group 1 is average. However, the performance on Group 3 shows a very low kappa coefficient because of the high type II errors. Group 3 sites contain features like steep slope and high percentage of terrain points. Therefore, the MTF method probably has flaw on process this type of areas.

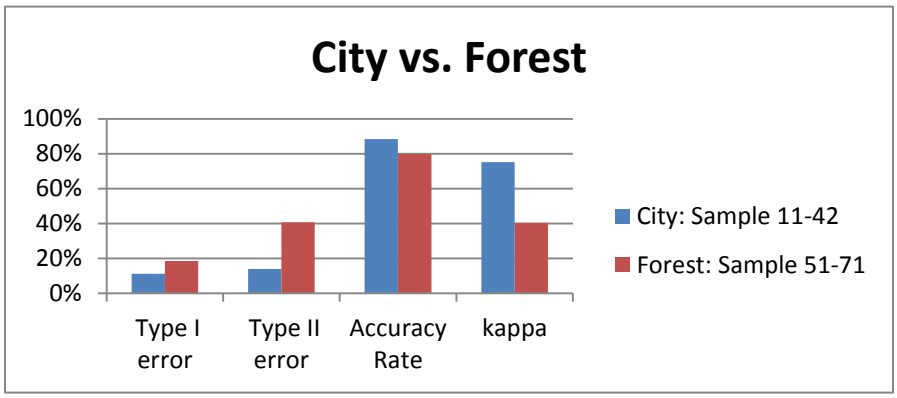


Figure 4-11 Average values of type I, type II errors, Accuracy and Kappa sorted by City Sites and Forest Sites

The ISPRS data is originally sorted as city sites and forest sites, the performance of the MTF method on city sites and forest site are shown in **Figure 4-11**. It is obvious that the performance on city site is better since it has lower type I, II errors and higher accuracy and kappa. The steep slope and the dense vegetation coverage might be the reason why the MTF method has an unsatisfactory result on forest sites. The forest sites are basically overlapping with the Group 3 sites. The high type II error is probably from the buildings on the slope which is a difficulty mentioned by (Sithole and Vosselman, 2003b). It is also the key to improve the value of the kappa coefficient.

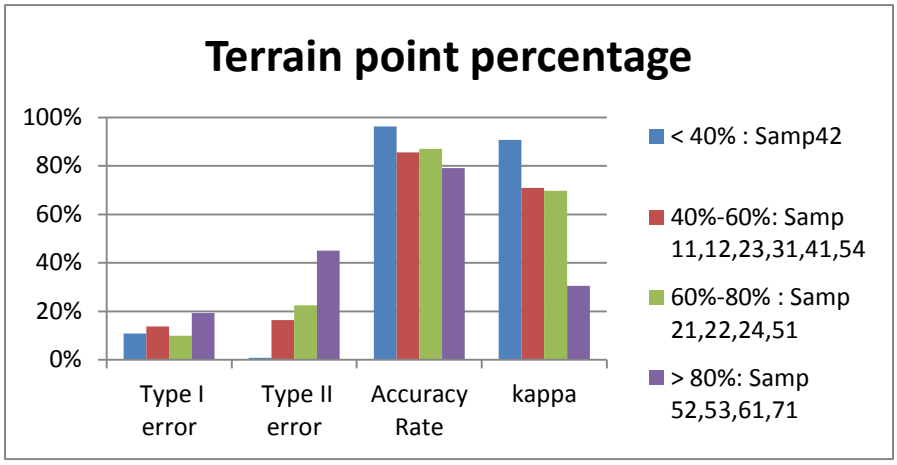


Figure 4-12 Average values of type I, type II errors, Accuracy and Kappa sorted by percentage of terrain point

The MTF is based on layering, which is a global analysis of the data height value. Therefore, the terrain points' portion of all points can affect the result. **Figure 4-12** shows the MTF performance based on different terrain point percentage. It seems along with the growth of the percentage, the errors especially type II error become higher, while the accuracy and kappa become lower. But it needs to be noticed that there is only one sample for the terrain point percentage smaller than 40%, and the sample which have higher than 80% terrain points are all in Group 3. Therefore, the uncertainty of this feature still requires further discussion.

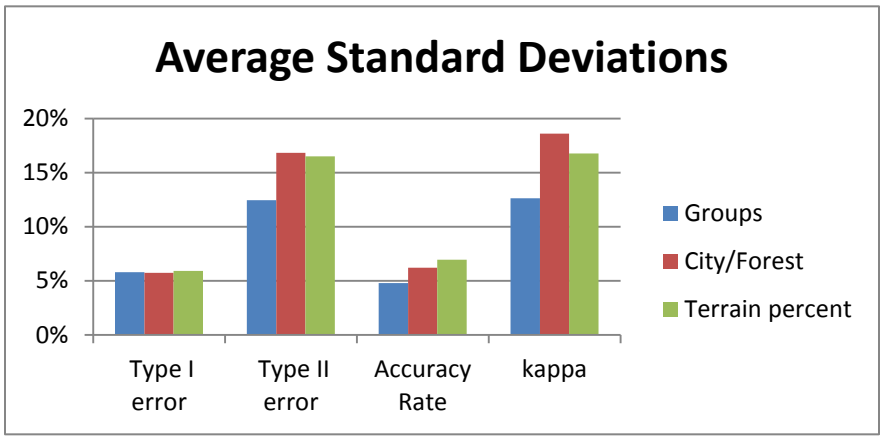


Figure 4-13 Average values of Standard Deviations of type I, type II errors, Accuracy and Kappa in three types of sortation

In order to know which feature of the data has more influence to the result, the average standard deviations of the previous three types of sortation are calculated as shown in **Figure 4-13**. The chart shows that they are all in the same range of each characteristic; however, the group sortation has the lowest average standard deviation among the three type of sortation.

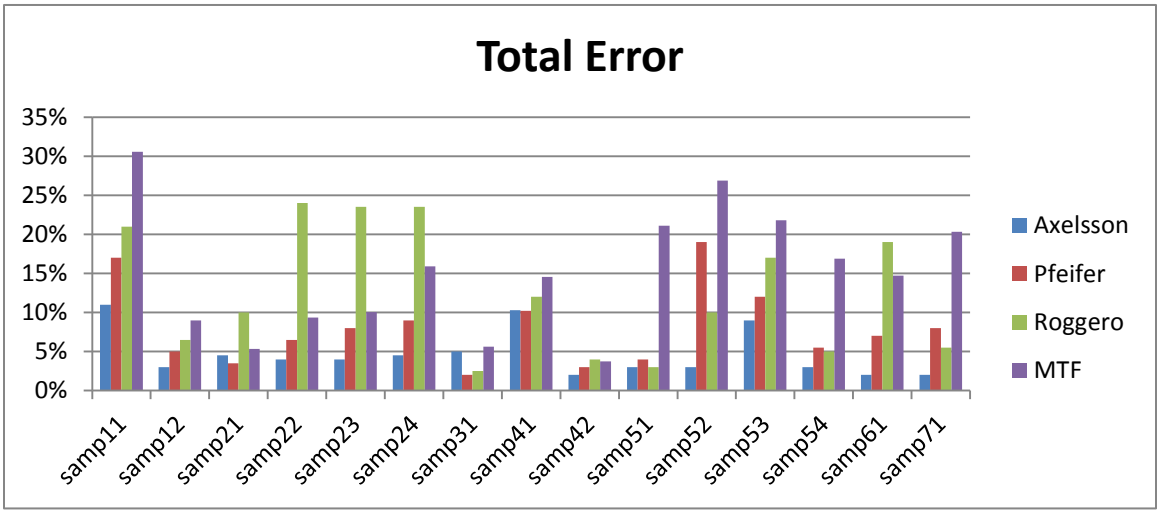


Figure 4-14 Total Error rate of MTF method and three method tested by ISPRS (Shao and Chen, 2010)

To compare with the algorithms analyzed by ISPRS, the error rate and average of kappa

coefficients are shown in **Figure 4-14** and **Figure 4-15** respectively. Unfortunately, the error rates are the worst of the four comparing method in nine of fifteen sites. However, it has better results than Roggero's method in the rest six sites. Similarly, the average of kappa coefficients chart shows a 61.2% of the proposed MTF method, which is the 6th of all 9 methods, only higher than Elmqvist, Brovelli and Sithole's methods (Meng et. al., 2009). Therefore, a further improvement of the MTF method is required.

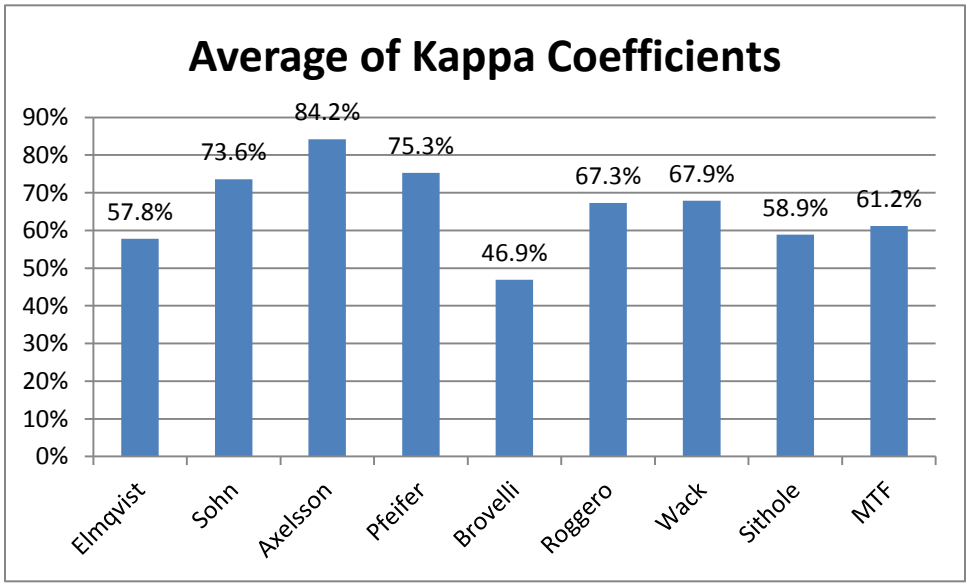
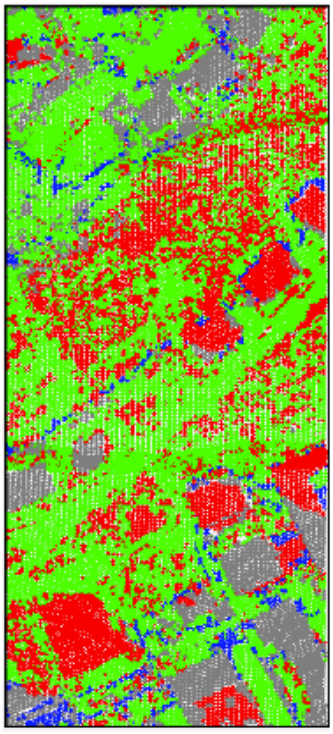


Figure 4-15 Average of Kappa Coefficients in 15 sites of MTF method and eight method tested by ISPRS (Meng et. al., 2009)

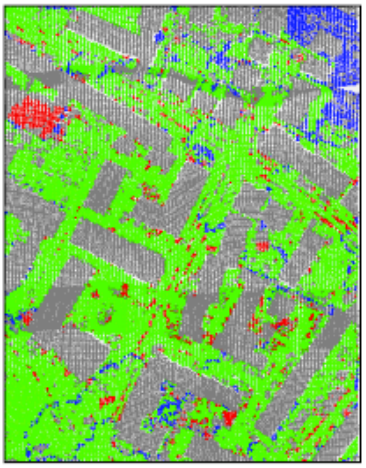
4.3.2 Qualitative Analysis

The misidentified terrain points can be interpolated by its neighbors, while an off-terrain point that is misidentified as terrain point will lead to an inaccuracy of the interpolated DTM. Furthermore, the changing of most parameters in the proposed method will lead to a trade-off between type I error and type II error ((Sithole and Vosselman, 2003b), and will be further analyzed in Section 4.4). Therefore, the proposed method focuses on minimizing the type II error. However, in some sites with very high terrain points' percentage, the minimizing of the type II error will lead to a big drop of the total accuracy because of the decrease of the type I error. Therefore, the total accuracy is the first consideration in these sites. The type I, type II error rates and the total accuracy rate of each site can be seen in **Figure 4-9**.

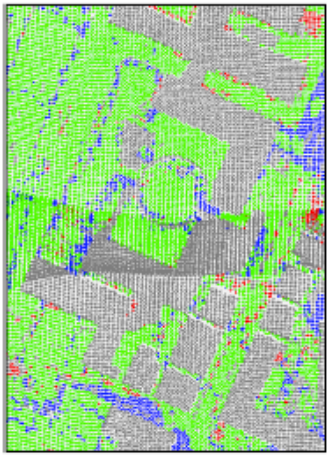
The most errors happens where difficulties exists (six type of filtering difficulties listed in Section 2.3 (Sithole and Vosselman, 2003b)). These six difficulties will be separately discussed based on the visualized cross-matric image in the following paragraphs. The visualized images for the fifteen samples are displayed in **Figure 4-16**.



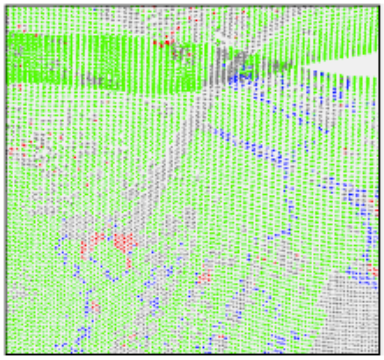
Sample 11



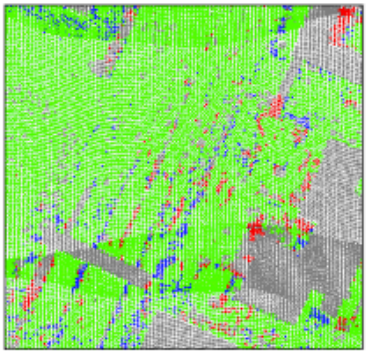
Sample 12



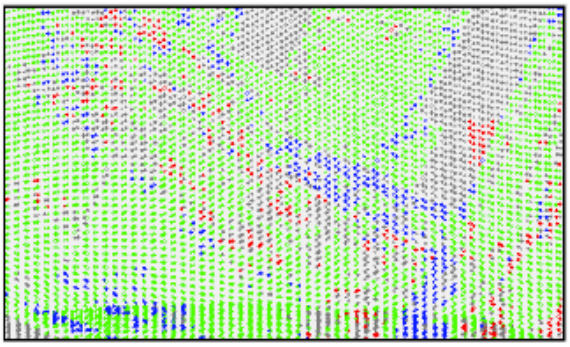
Sample 23



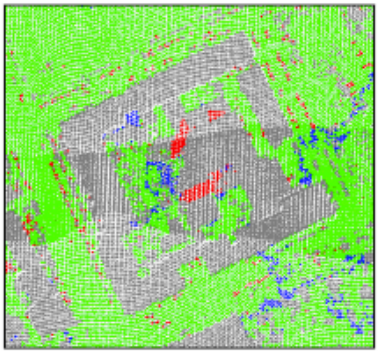
Sample 21



Sample 22



Sample 24



Sample 31

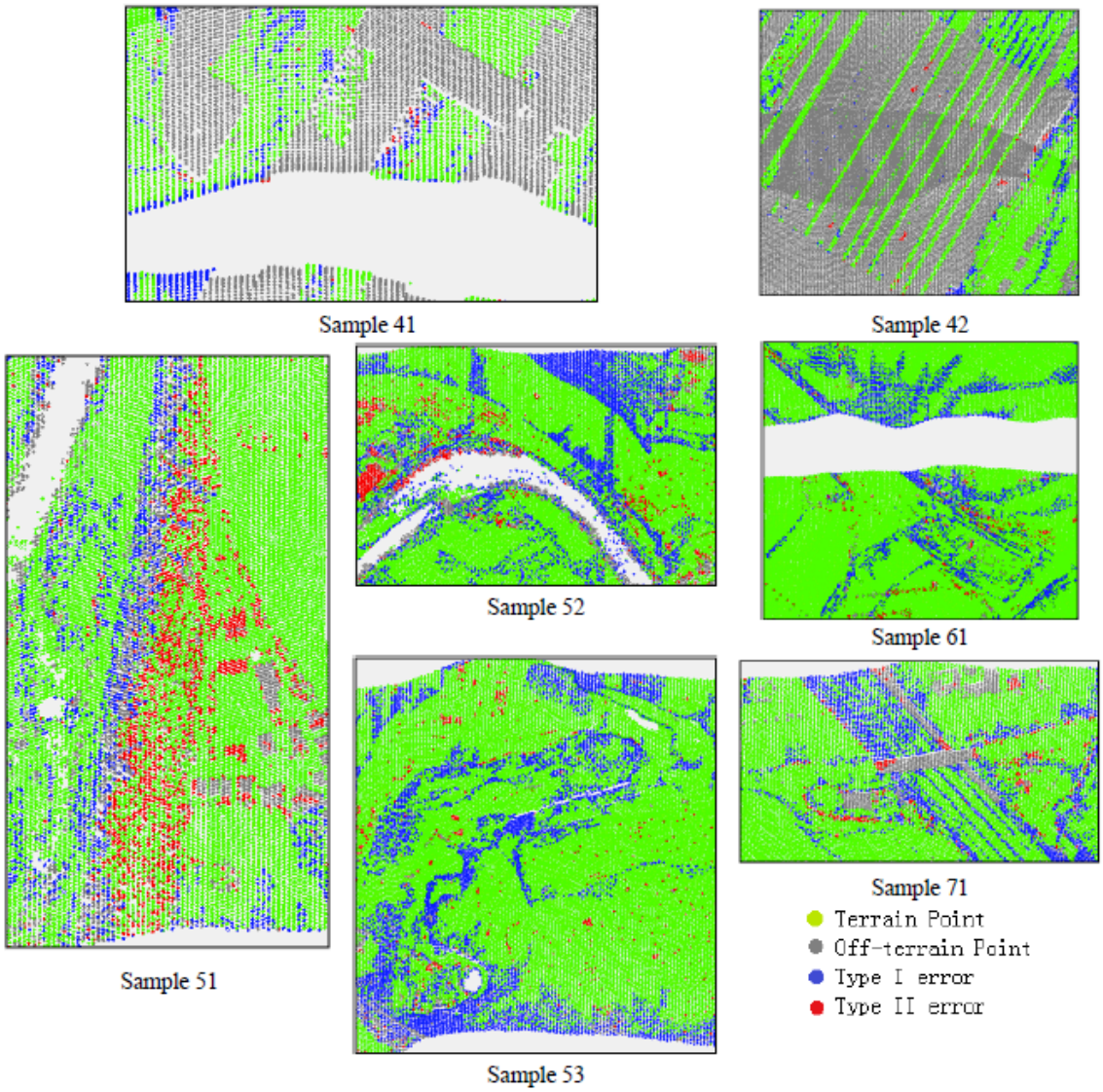


Figure 4-16 Visualized cross-matrices of study sites, each image is displayed at a unique scale.

(1) Steep Slopes

Samples 11, 51, 52 contain steep slopes. Many type II errors can be seen on these images, especially on Sample 11. This is because the steep slope will make the terrain points

distributed all across the horizontal axis in the height histogram of the sites as shown in **Figure 4-17**. The layers generated from this type of histogram can hardly be used to divide terrain and off-terrain points. Since the entire sample 11 is on a steep slope, the errors are all over the place. The steep slope on sample 51 is just a strip in the center, thus the errors are mainly located on that strip as shown in **Figure 4-18**.

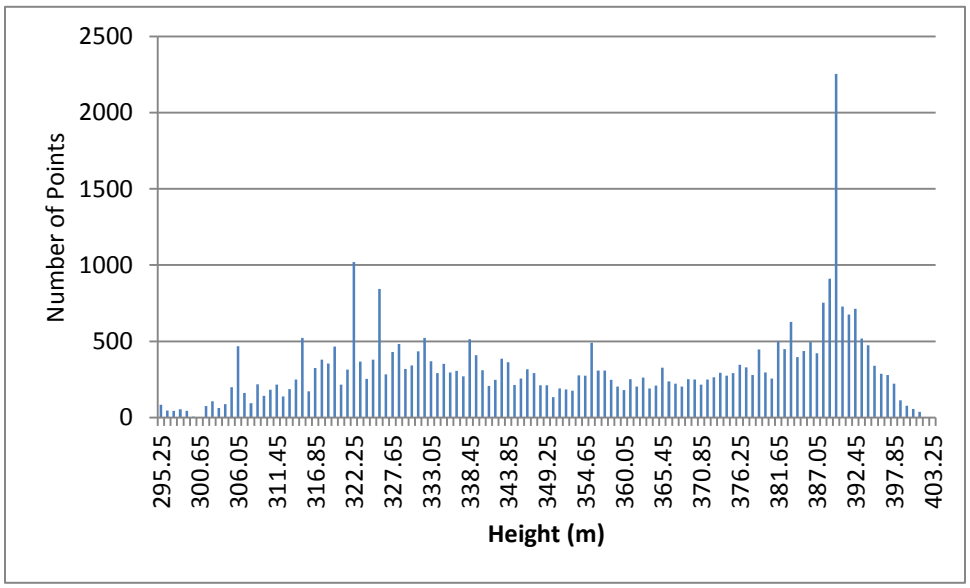
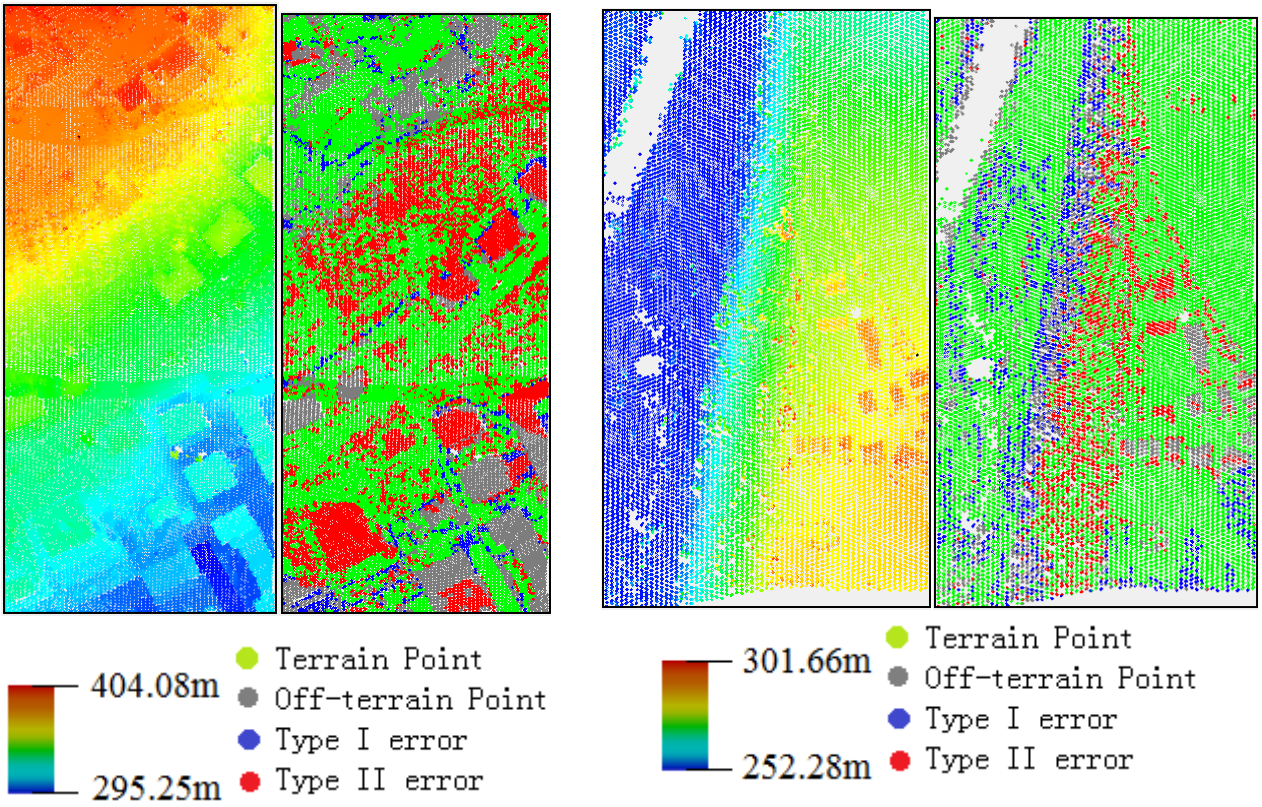


Figure 4-17 Height Histogram of Sample 11



(a) All points on Sample 11 (b) All points on Sample 51

Figure 4-18 Steep slopes on Samples 11 and 51

(2) Discontinuities

Discontinuities happen in samples 22, 23 and 53. The total accuracy of the samples 22 and 23 is around 90% while the kappa coefficients are around 80%, which are good results. The type I error is around 15% on sample 23, and it becomes more significant 49% when comes to sample 53. These errors are from the sharp ridges or valleys as shown in **Figure 4-19**. This is probably because the same reason as the steep slopes. Actually a steep slope can be viewed as a big discontinuity, and it is classified as a type of

discontinuities by Sithole and Vosselman (2003b). However the type I error looks more significant is probably because the number of off-terrain points is relatively small (there are 32989 terrain points and 1389 off-terrain points in sample 53).

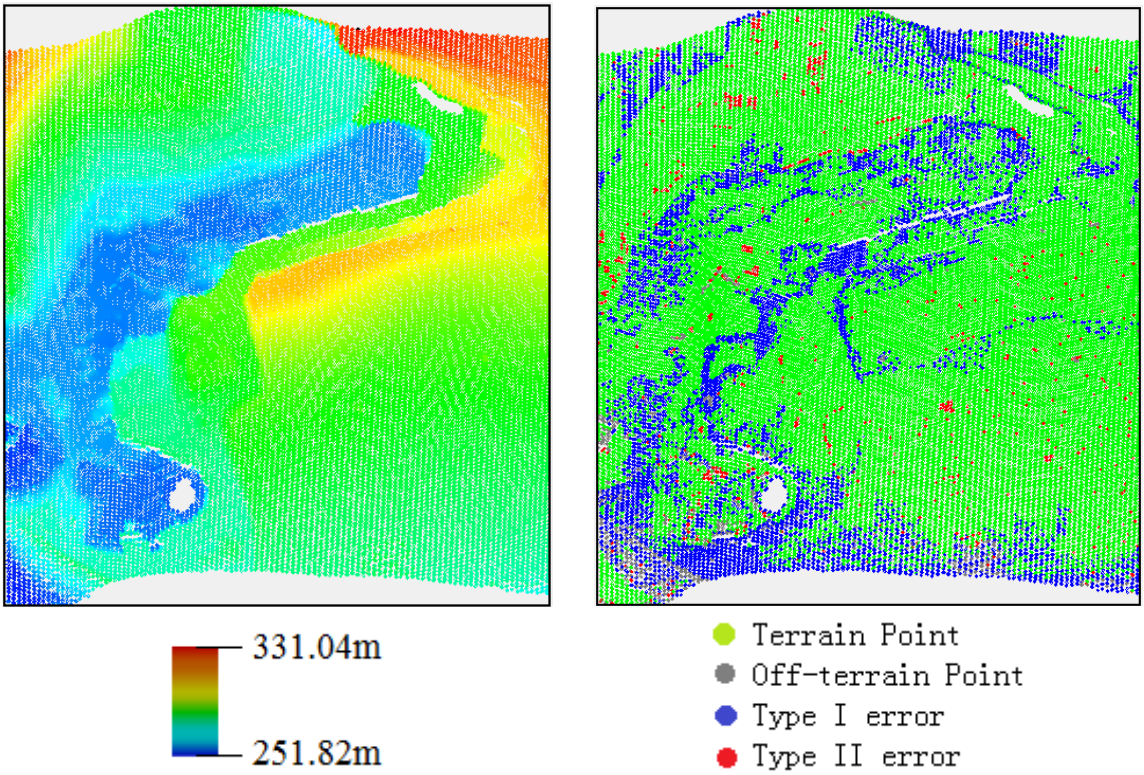
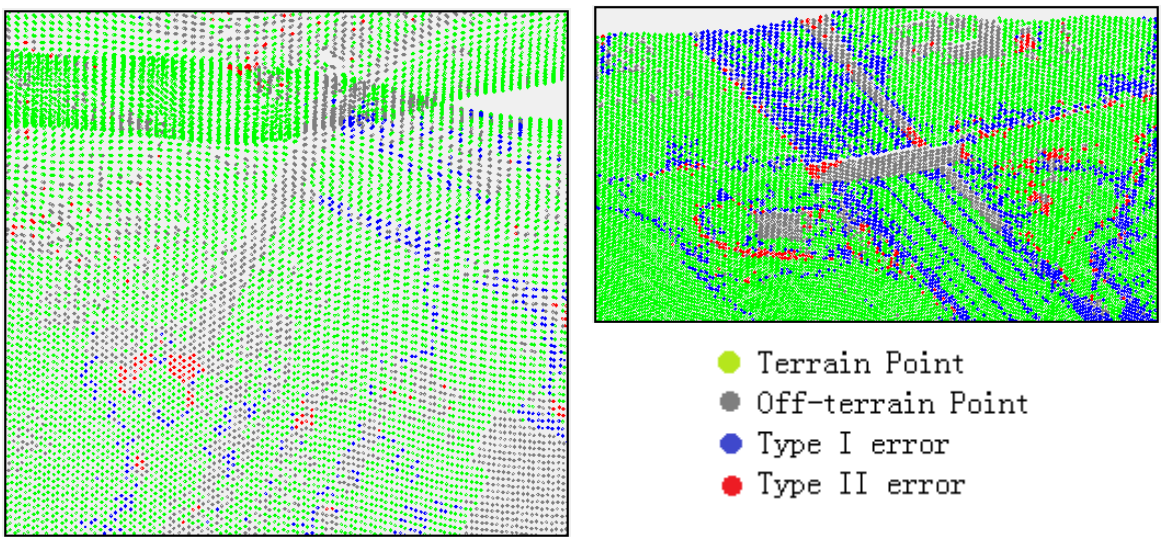


Figure 4-19 Type I errors happens where the ridges or valleys are.

(3) Bridges

As shown in **Figure 4-20**, most points on the bridges are identified as off-terrain objects. However, there are some type II errors at the beginning or end of the bridges, which is similar as some method tested by Sithole and Vosselman (2003b). The reason of this is that the bridges usually smoothly start from the bare earth, while begins and ends of the

bridges are very close to the terrain.



Sample 21 Sample 71

Figure 4-20 Bridges on samples 21 and 71

(4) Complex Scenes

Samples 11, 22, and 23 are tagged as complex scenes by Sithole and Vosselman (2003b). Especially sample 23, a plaza contain structures like three sides of buildings in different shapes, pathway between buildings to the road, stairs from plaza to the road, and even a sunken arcade in the center. The definition of the bare earth here is under the requirement of the availability to walk with no obstruction (Sithole and Vosselman, 2003b). All samples here have been analyzed in previous filtering difficulties. The results of sample 11 need to be improved, while the results of Samples 22 and 23 are good. Therefore, the proposed method can handle the complex scenes on flat terrain surface well, but cannot

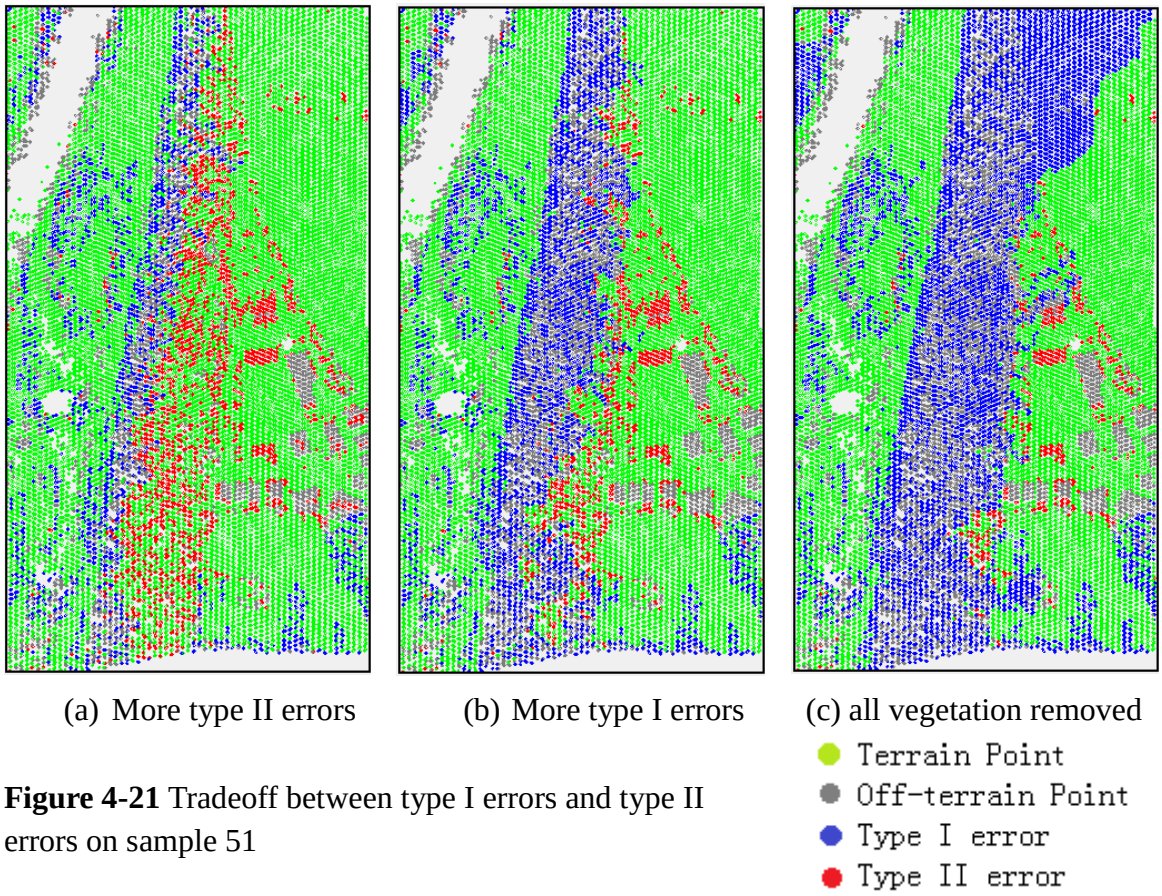
provide a high accuracy result on steep slope areas.

(5) Outliers

Samples 31 and 41 contain outliers. If it is a low outlier on the terrain, a few type I errors will be generated around it. Usually only small number of outliers can be seen on the data, thus the accuracy rate will not decrease significantly because of the outliers. However, they can be removed by the small group of point removal function of the proposed method.

(6) Vegetation on slopes

Samples 51 and 52 are two examples with vegetation on steep slopes. In sample 51, the steep slope and vegetation are mixed together. As can be seen in **Figure 4-21**, the type II errors on the steep slope are generated from the vegetation. These type II errors can be removed by tweaking the parameters. However, this will sacrifice the type I errors rate instead. This problem happens in some of the filters compared by Sithole and Vosselman (2003b) as well.



(a) More type II errors (b) More type I errors (c) all vegetation removed

Figure 4-21 Tradeoff between type I errors and type II errors on sample 51

To conclude, in all the six difficulties listed above, steep slope (discontinuities) is the hardest one to get a high accuracy rate for the proposed MTF method. Relatively small problems happen in the detection of the bridge, bare earth in complex scenes with flat terrain and vegetation, and around 90% total accuracy can be generated in the samples with these difficulties. Outliers can be removed by applying the small group of point removal function.

4.3.3 Results of Other Sites

As previous mentioned (Section 4.1.1), in order to complement some defects of the ISPRS data, two other data (City of Waterloo data from Optech and Demo City data from TopoSys) sets are tested. These two datasets are evaluated by the sampling estimation introduced in Section 4.2.2. The identified terrain points results and the accuracy results are compared between the developed Multi-scale Terrain Filtering method and two comparing methods (Morphological filter and Adaptive TIN filter named by software ALDPAT Version 1.0).

The experiments parameters of the proposed Multi-scale Terrain Filtering method in each site are listed in **Table 4-8**. These parameters combination can produce the lowest average offsets (error) between reference DTM and the DTM generated by the proposed method on sampling points during all experiments. The parameters tested in the two comparing method are listed in **Table 4-9** and **Table 4-10**.

Table 4-8 Parameters of Multi-scale Terrain Filtering method

Parameter	Site A	Site B	Site C	Site D	Site E	Demo City
Width	1m	1m	1m	1m	1m	1m
Del	0.6	0.4	0.4	0.6	0.6	0.6
Min Layer	50	50	50	50	50	50
Min Cell	-	100	100	100	100	1000
K	2m	2m	2m	2m	2m	2m
Level num	5	5	4	5	6	6
Tanθ	-	1	1	1	1	1
Identification tolerance	-	0.3	0.5	0.3	0.5	0.5
Classification tolerance	1	1	1	1	1	1

Table 4-9 Parameters of Morphological filter

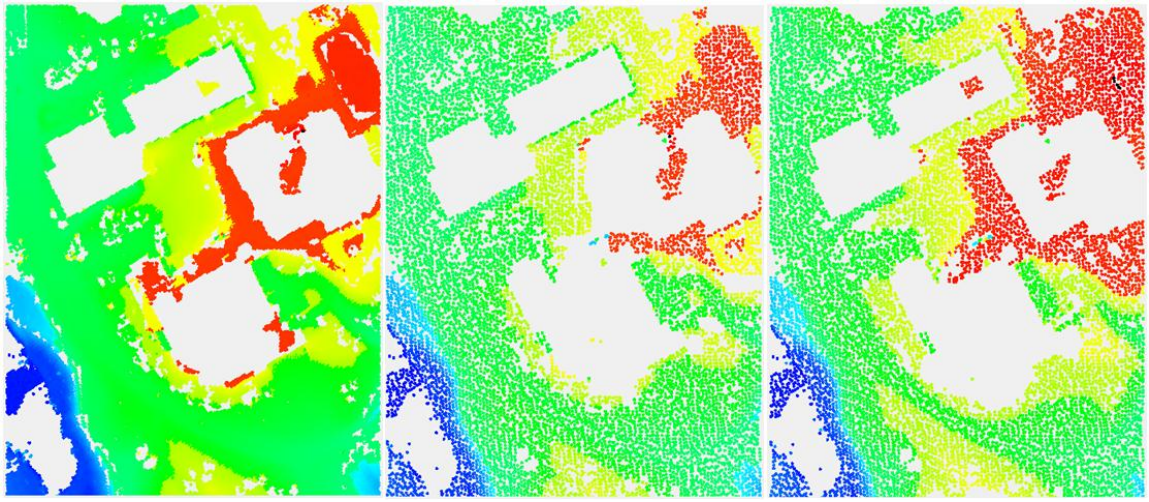
Parameter	Value
Cell Size	2.00 m
Slope	0.08
Init Threshold	0.60 m
Max Threshold	9999.00 m
Window Base	2.00 m
Power Increment	1.00 m
Window Series Length	8 m
Init Radius	1.00 m
Window Series	1,2,4,8,16,32,64,128
Threshold Series	0.60,1.60,2.60,4.60,8.60…
Result Mode	Terrain
Data Mode	Real
Min WndSize	1 m
Direction	X and Y

Table 4-10 Parameters of Adaptive TIN filter

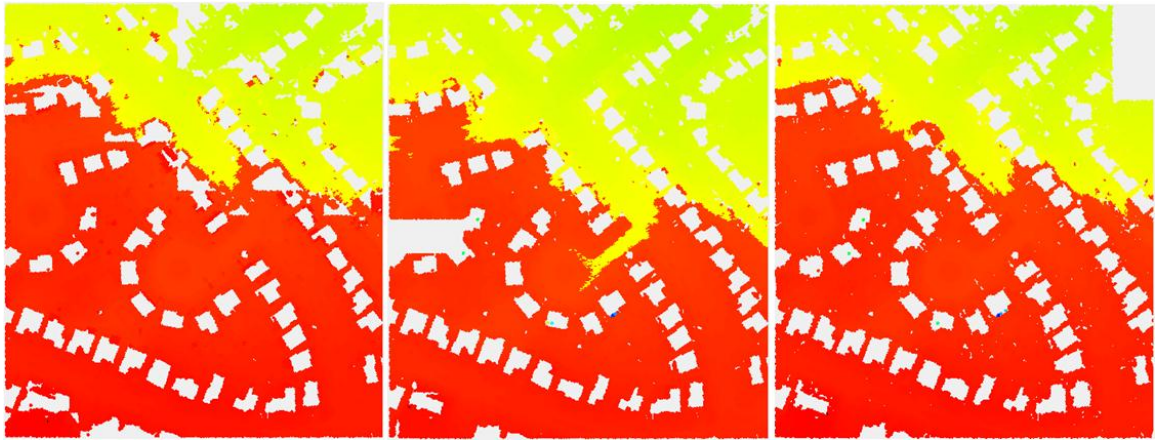
Parameter	Value
Cell Size	2.00 m
Z Difference	0.20 m
Angle Threshold	0.00
Init TriGrid Size	100.00 m
Tile X Width	200.00 m
Tile Y Height	200.00 m
Tile Buffer	20.00 m

The identified terrain points results obtained by two comparison methods (Morphological filter and Adaptive TIN filter) and the developed Multi-scale Terrain Filtering method on two LiDAR datasets are shown in **Figure 4-22**.

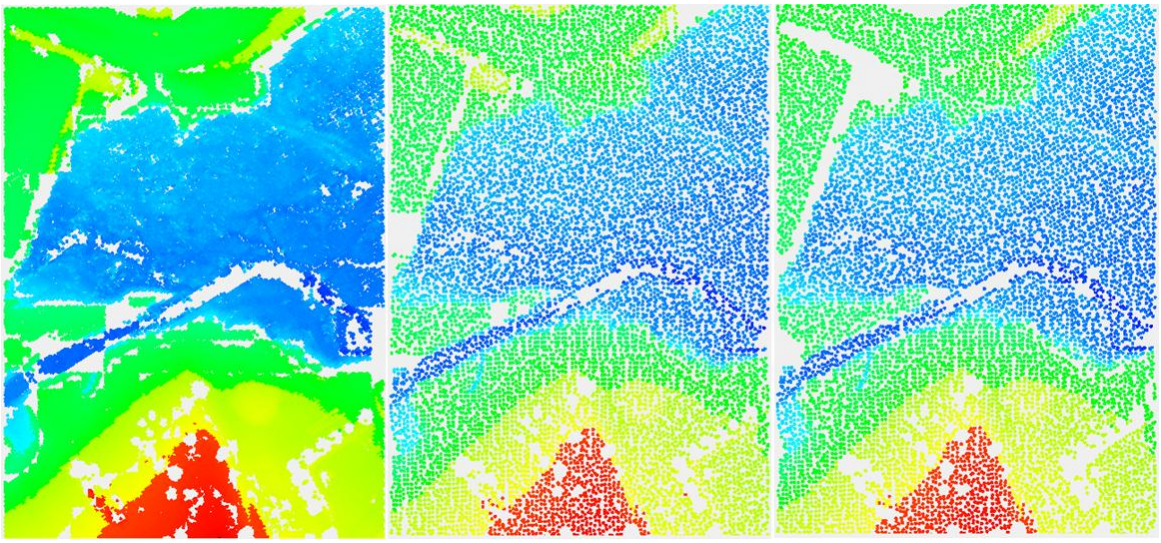
Multi-scale Terrain Filtering Morphological Filter Adaptive TIN filter



Site A: 87640 points

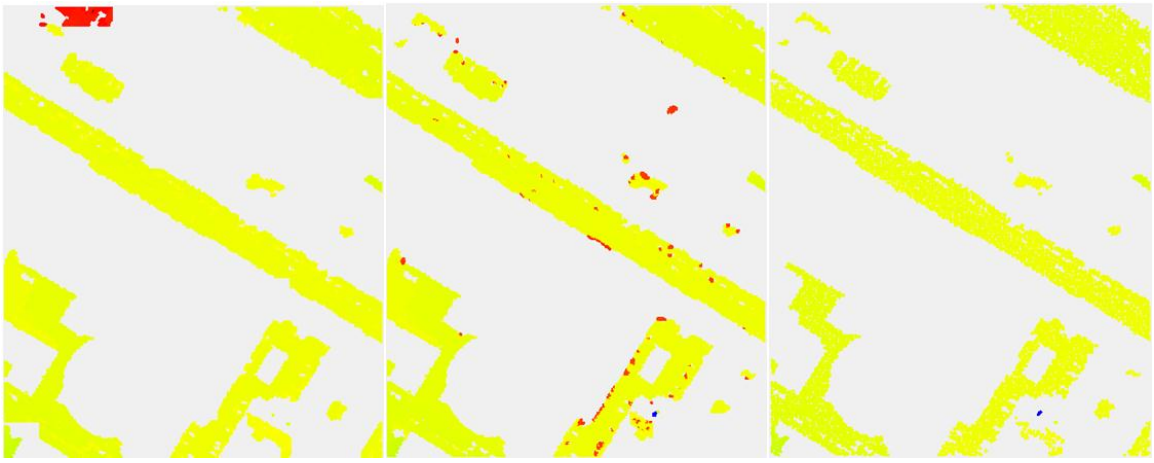


Site B: 132697 points

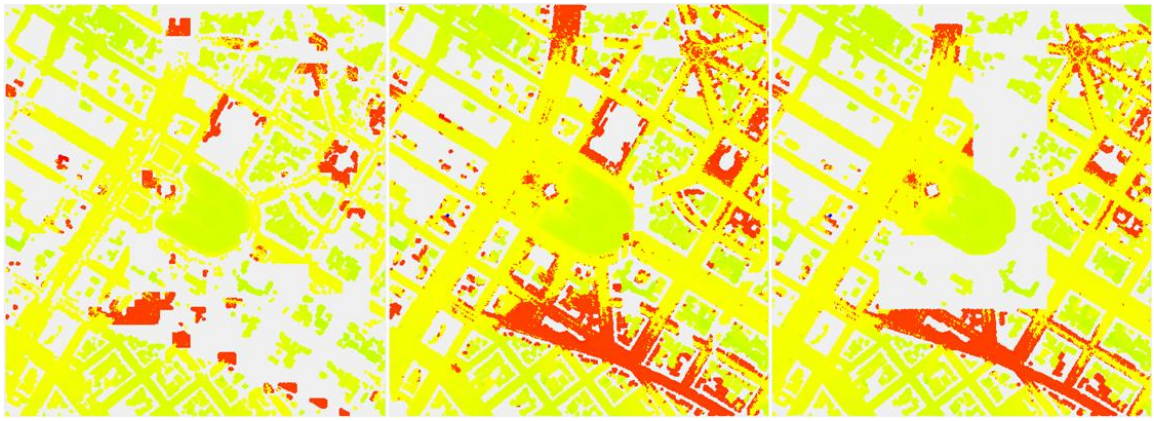


Site C: 84732 points

Site D: 89411 points



Site E: 135016 points



Site F (entire Demo City data): 2738623 points

Figure 4-22 Comparison of terrain points result Images generated by MTF method, Morphological filter, and Adaptive TIN filter

The identification results obtained by the three methods show the advantage of each one. However all three methods have some defects. The result using the morphological filter is missing the whole section of the terrain points in some area probably due to the setting of the threshold. Adaptive TIN filter performed better than the other two in most cases, while the terrain points in some large rectangular area are missing in Site B and the Site F. The proposed method generating some unexpected small empty rectangle is caused by some low outliers in the wrongly identified area. The performance of layering becomes lower when handling a large area as shown in **Figure 4-22** Site F (entire Demo City data). This is because in a large area with many types of objects mixed, the points of the objects and the terrain cannot be divided clearly on the height histogram.

Figure 4-22 shows the identified terrain points, which are intermediate results of the whole process. The final interpolation will make the DTMs covered the whole area. **Table 4-11** shows the evaluation results (average errors, standard deviations, and the worst errors) of the DTMs generated by the proposed method and two comparing method. The best results among the three methods for each criterion are highlighted. Except the processing time, the proposed method has six best results in all 18 results, while the morphological filter has seven and adaptive TIN has five. However, the result numbers of the three methods are similar. Furthermore, the evaluation results by proposed method in Site D, E are obviously better than the two comparing filters. But it fails in processing the

entire TopoSys Demo City data which the reason has been discussed previously.

Table 4-11 Evaluation Results on six sites generated by three methods

Number of Points	Criteria	Developed Method	Morphological filter	Adaptive TIN filter
Site A	processing time	0.9s	5s	15s
87640	Average offset (m)	-0.237	0.148	0.105
	STDEV of offset (m)	0.819	0.639	0.689
	Worst error (m)	-4.273	3.247	3.360
Site B	processing time	1.3s	11s	27s
132697	Average offset (m)	0.445	0.569	0.656
	STDEV of offset (m)	1.096	0.907	0.916
	Worst error (m)	-3.810	2.742	2.813
Site C	processing time	0.7s	4s	21s
84732	Average offset (m)	-0.593	-0.506	-0.495
	STDEV of offset (m)	1.008	0.987	0.959
	Worst error (m)	-3.168	-2.991	-2.667
Site D	processing time	1s	3s	3s
89411	Average offset (m)	0.005	0.018	0.034
	STDEV of offset (m)	0.080	0.105	0.335
	Worst error (m)	-0.579	-0.643	-0.915
Site E	processing time	1.4s	6s	5s
135016	Average offset (m)	-0.092	-0.336	0.061
	STDEV of offset (m)	0.170	0.630	0.210
	Worst error (m)	-0.969	-2.936	-1.514
Site F	processing time	26.8s	100s	240s
2738623	Average offset (m)	-0.159	0.032	0.355
	STDEV of offset (m)	0.502	0.390	0.826
	Worst error (m)	-3.709	-3.704	4.314

As can be seen, **Table 4-11** also presents the processing time of each method. A great amount of data is used in the field of geological information processing. The processing time is an important role in the transfer of the lab work to the real world commercial

applications. In this aspect, thanks to the terrain point's identification of the proposed method is based on the comparison between the layers numbers, the Multi-scale Terrain Filtering method will consume less processing time than the other two methods. For example for the experiment of Site C, the consumed time for the proposed method is 0.7 seconds while the morphological filter and adaptive TIN filter took 4 seconds and 21 seconds, respectively to complete the same identification task. The proposed Multi-scale Terrain Filtering method generally cost less than one third of the time compared with the morphological filters cost and even less than the adaptive TIN filter, while the identification results and the evaluation results were similar as the other two methods.

In conclusion, the proposed Multi-scale Terrain Filtering (MTF) method can identify terrain points in flat terrain areas with around 90% total accuracy. This accuracy rate drops when handling steep slope (discontinuities) areas. The average of kappa coefficients of the proposed methods tested on fifteen ISPRS sample sites is 61.2%, which is higher than three methods tested by ISPRS, but lower than the other five. The proposed method can solve or partially solve the difficulties like bridge, complex scenes, outliers and vegetation. However, whenever the test sites contain steep slopes, the result and performance becomes lower. The compliment tests show that the proposed method has difficulties in processing very large data as well. The good thing is the processing time is promising comparing to the other two methods.

The proposed method is developed on C# by Visual Studio 2008. The morphological filter and adaptive TIN filter compared in this research are included in the ALDPAT Version 1.0, which was developed by the International Hurricane Research Center, Florida International University in 2007. The final IDW interpolation and accuracy evaluation are process on ArcGIS 10. The processor of the computer is equipped with Intel Core2 Duo CPU T5800 @ 2.00 GHz and 4 GB RAM.

4.4 Sensitivity Analysis of the Parameters

As previous experiments demonstrated, the test results are based on the repeated experiments and the adjusting of the parameters. Therefore, choosing the parameters is critical to achieve a result with higher accuracy. And the sensitivity of the parameters, i.e. how the changing of the parameters influences the method output, will be a serious topic to discuss in the following section.

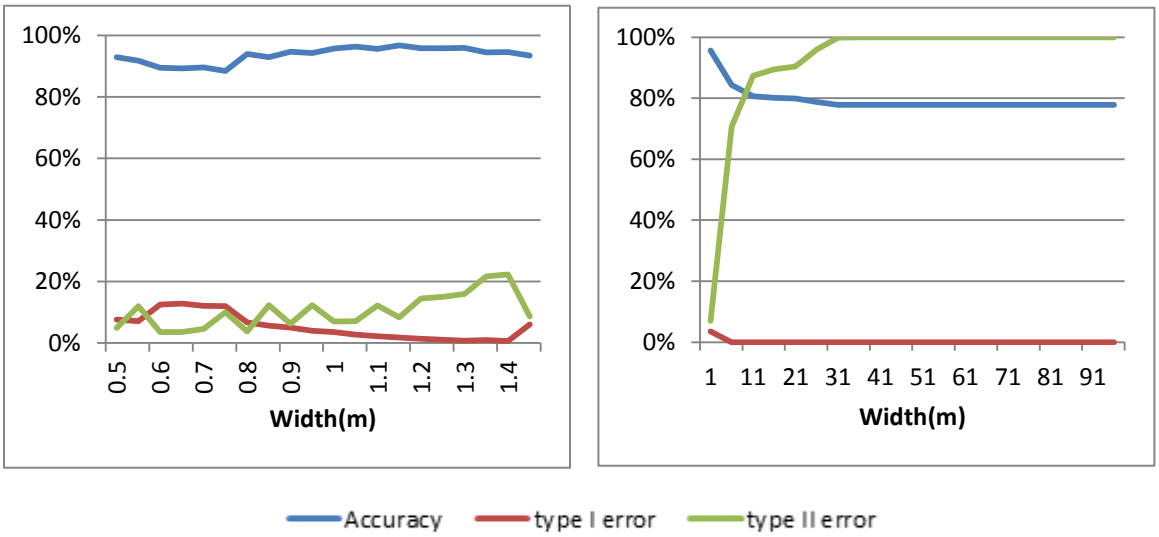
There are nine parameters involved the proposed method, they are *Width, Delta , Min Layer, Min Cell, K, Number of Levels, Tanθ, Identification tolerance and Classification tolerance*. Parameters *Width* and *Delta* are two very "close" parameters which decide how the layers will be generated together; therefore they will be discussed in a group. Other parameter will be discussed separately. Based on the previous introduced analysis method and the experiment results, the sensitivity of the proposed MTF method is analyzed by

parameters as follows.

The following discussion is tweaking one parameter each time. During a discussion of one parameter, the other parameters will keep the same. The setting of the other parameters can be found in **Table 4-7**.

4.4.1 Width and Delta

Width and *Delta* (percentage) are two parameters which are used in the generation of layers. *Width* is used to define the unit height difference in the generation of the height histogram. When *Width* becomes larger and larger, usually there will be fewer layers generated.

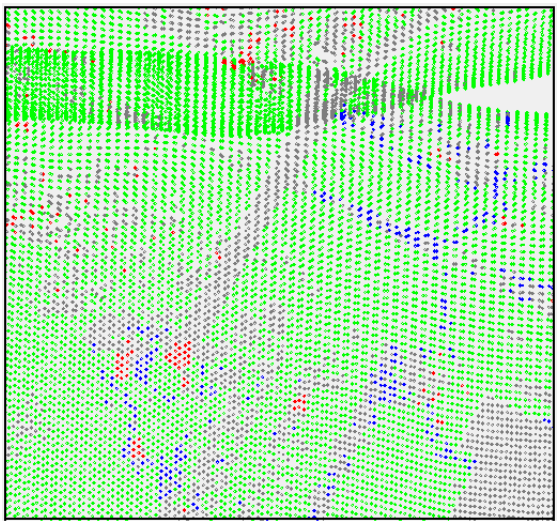


(a) Width: 0.5 ~ 1.5 m (b) Width: 0.1 ~ 100 m

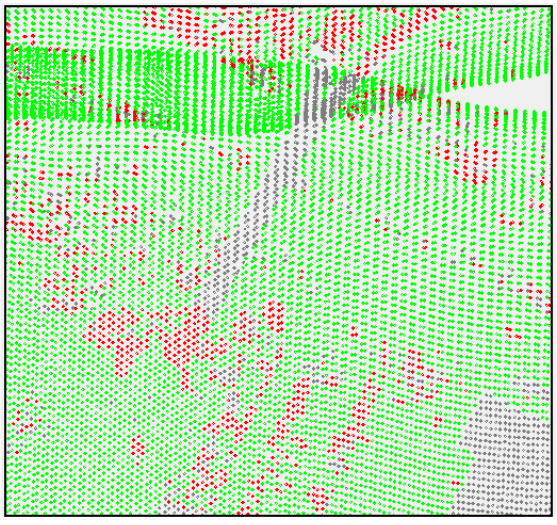
Figure 4-23 Width changes in different ranges (Sample 21)

Theoretically, the range of parameter *Width* can be any number between vertical precision

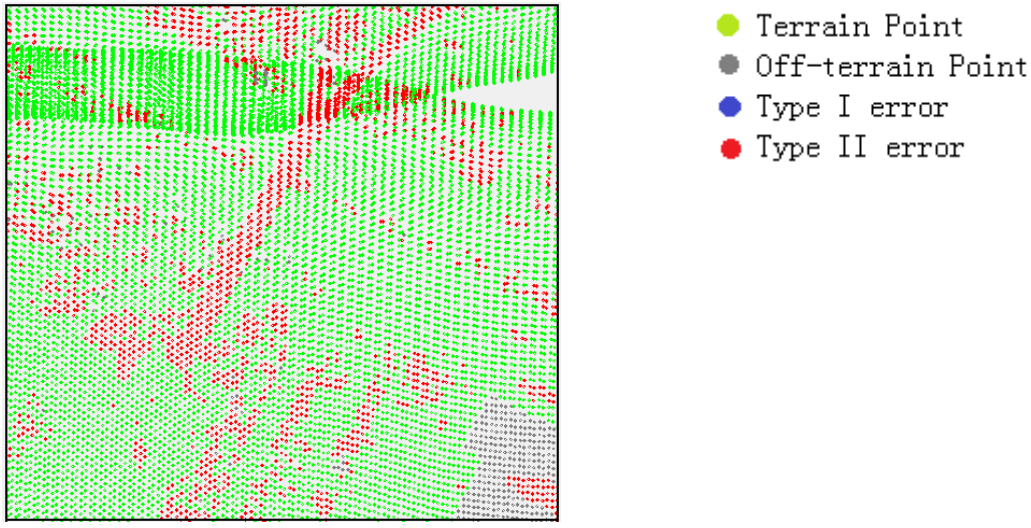
of the system and the elevation value range of the data. However, usually type II error will increase along with Width as shown in **Figure 4-23** (b). This is because when the width of the histogram bin becomes larger, the possibility for a histogram bin to include both terrain and off-terrain points becomes higher. The extreme situation is that one histogram bin includes every point in the data, and all points will be identified as terrain point, which means 100% type II error and 0% type I error, as shown in **Figure 4-24**. But as shown in **Figure 4-23** (a), this parameter is stable when set to a relatively small range. Therefore, the parameter Width should be chosen from vertical precision of the system to a low object height (e.g. one storey of a building).



Width = 1



Width = 5



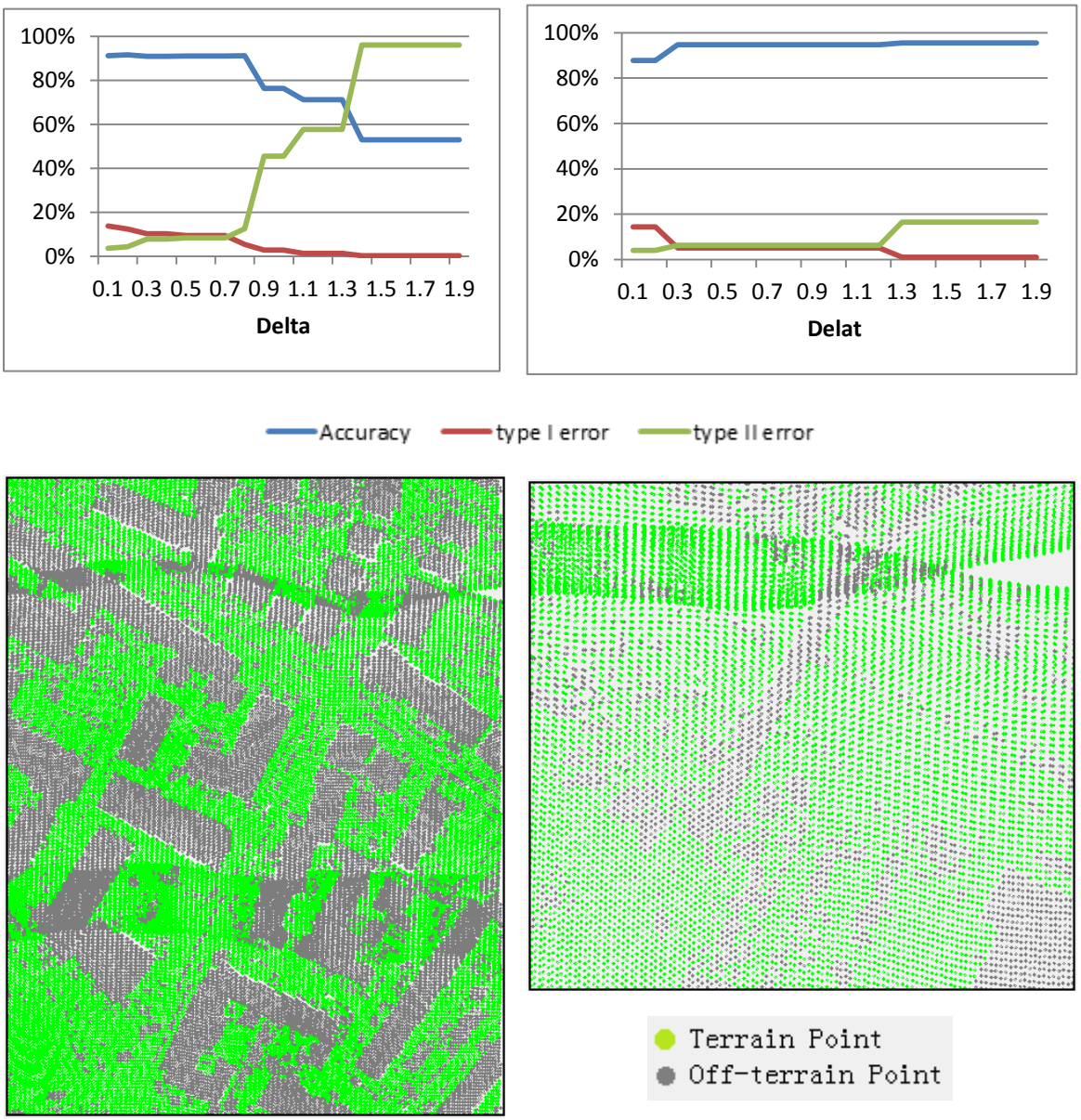
Width = 10

Figure 4-24 Visualized cross-matrix with different *Width* value

Delta defines the percentage threshold to separate the histogram into different layers. The layer generation is based on comparing the number of points between adjacent height units. If difference of the numbers of points between two neighbor units of the histogram is over the given *Delta*, the border of the two units will become the border of two layers. If the difference is under the *Delta* percentage, the two units will be grouped into same layer. This parameter will affect the number of layers as well. When it getting larger, the required difference is larger and harder to reach, therefore fewer layers will be generated. The extreme situation is same as *Width*, the requirement is too hard to generate two layers, and all points will be identified as terrain point.

However in the certain range of the testing, the stability is related to the number of terrain points and off-terrain points. As shown in **Figure 4-25** (a), in the situation that the number of terrain points is similar to the off-terrain points, the sensitivity of *Delta* is high, while it

becomes more stable when the difference between the numbers are big as shown in **Figure 4-25** (b). Because then the *Delta* is relatively small to make the two groups of points into the same layer, which can lead to the correct layer separation of two types of points.



(a) Terrain points = 26691, Off-terrain points = 25428 (Sample 12)

(b) Terrain points = 10085, Off-terrain points = 2875 (Sample 21)

Figure 4-25 Sensitivity of *Delta* varies in different data structure

Therefore, the proposed layering method works more stable in the situation like sample

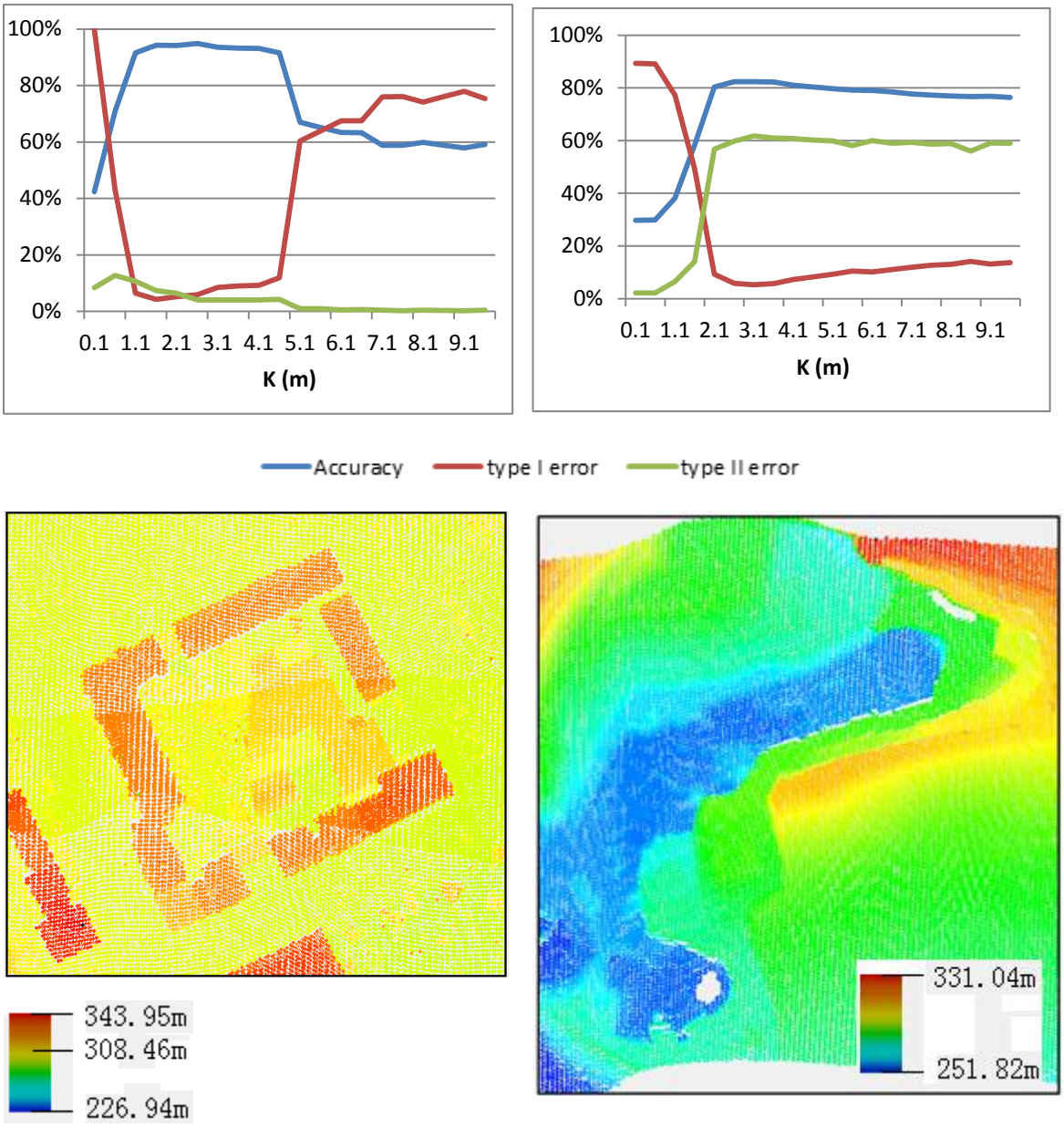
21. The trade-off of the type I error and type II error is also noticeable and happens in most of the experimental sites, which means the changing of the parameters will affect both type I and type II errors.

4.4.2 Unit Cell Size: K

K is the unit cell size, as well as the side length of a cell of the first level, and cell sizes of every pyramid level are calculated by *K* and its *Number of Levels*. The influence of a low point is decided by *K* and *Number of Levels*.

As shown in **Figure 4-26**, type I errors decline sharply from almost 100% to less than 10% when *K* reaches the points resolution (average distance between points). Because if the unit cell size is K smaller than the resolution, there will be many empty cells in the bottom level which are going to be interpolated by the low point, this will lead to a high type I error rate.

But normally, when *K* increases, there will be more points in a cell of the bottom pyramid level. Since the last turn of identification and interpolation is performed in the bottom level, no further identification will be processed inside its cells. Therefore, the increase of *K* (when *K* is higher than the resolution) will lead to the growing of type I error, which means more terrain points will be wrongly identified as off-terrain points. The fast increase in **Figure 4-26** (a) happens when the unit cell is larger than the short edge of the building. However this will also reduce the type II error.



(a) resolution: 0.99m (Sample 31) (b) resolution: 2.43m (Sample 53)

Figure 4-26 Type I errors are relatively highly effected by *K*

As can be seen from **Figure 4-26**, type II error is almost not affected by the changing of *K*. (The initial sharp increase of type II error in **Figure 4-26**(b) is because almost every point is identified as off-terrain points when *K* is close to zero. And the high type II error rate in **Figure 4-26**(b) is a problem when the proposed MTF method handling steep slope data.) It

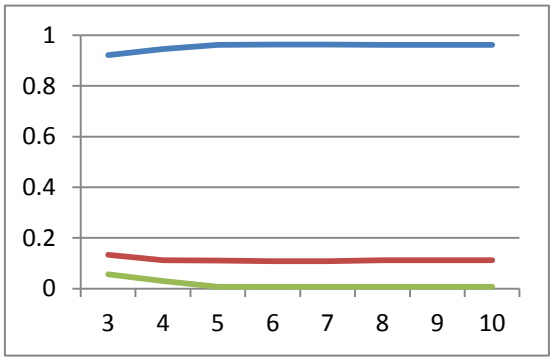
slowly declines along with the increase of *K*.

To conclude, the change of *K* will influence type I error more than type II error, and the sensitivity is low when *K* is higher than the data resolution and smaller than the size of the building.

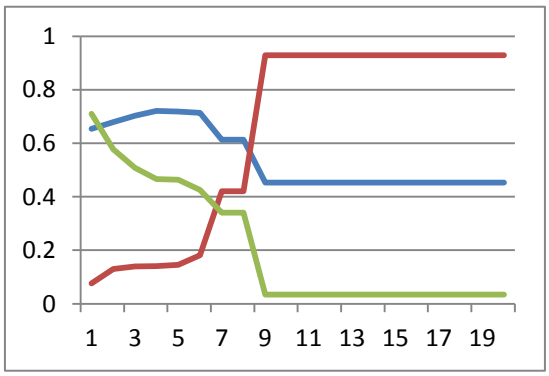
4.4.3 Number of Levels

Number of Levels defines the number of rasterized pyramid levels to be generated in the multi-scale terrain filtering process. Since the developed method is basically processing the data in each divided cells separately, this parameter determine the range of the biggest processing area which sharing a same lowest reference point. The increasing of the *Number of Levels* will lead to a decline of the number of cells in the top level, and bigger influence areas of the lowest points in these cells.

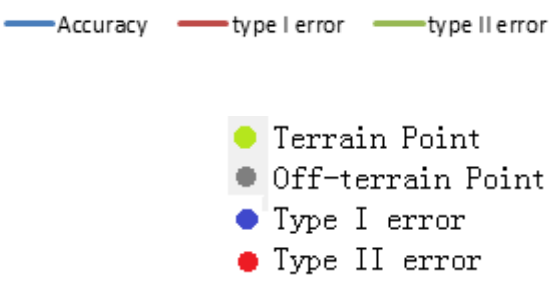
According to the experiments results, the parameter *Number of Levels* is very stable in 14 of all 15 samples. No matter what is the number of the type I or type II error rate, they almost stay the same (the difference between highest and lowest rate is usually less than 1 %.) when the *Number of Levels* is bigger than 4 or 5. As shown in **Figure 4-27** (a), type II error rate declines a little along with the increasing of the Number of Levels if it is smaller than 4. Because if there are only a few levels, the off-terrain points have more chance to be included as the lowest points of the top level.

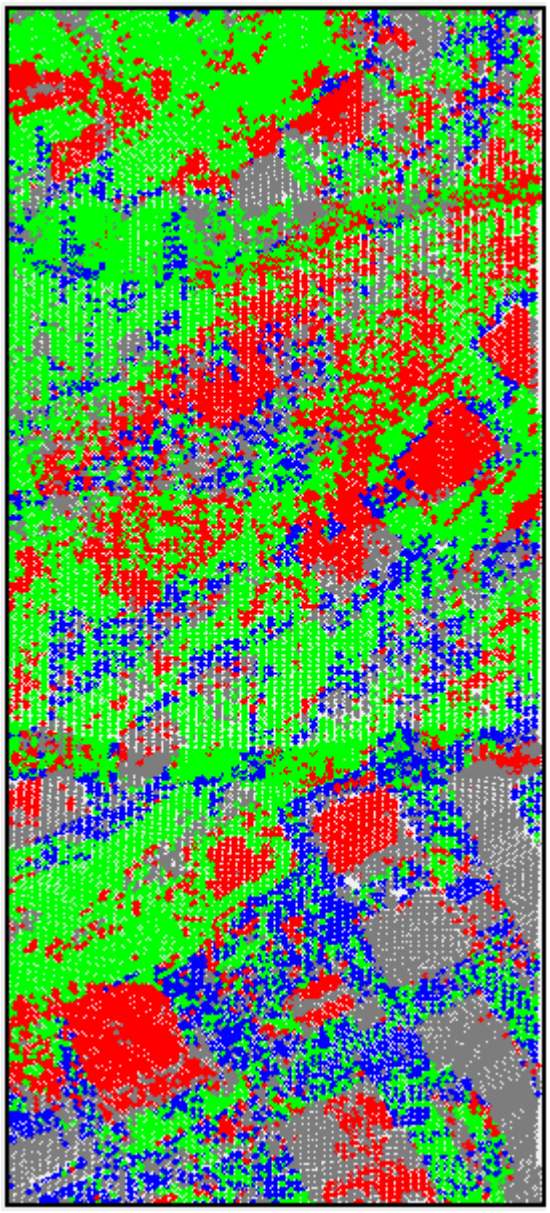


(a) Number of Levels is very stable in most samples (14 of 15) (Sample 42)



(b) Special case (Sample 11)





(c) Visualized Cross-Matric of Sample 11

Figure 4-27 Sensitivity of *Number of Levels* is low

A special case for the stability of the *Number of Levels* happens in Sample 11. An explanation is that this sample is on a steep slope, the points are evenly distributed in every height histogram, and it is hard to correctly layer the data. It also has similar number of terrain and off-terrain points, which make the identification even harder by the proposed method.

4.4.4 Slope Gradient: Tanθ

This parameter is used as an additional verification of the on-terrain verdict after the comparison of the layer numbers. As shown in **Figure 4-28**, the *Tanθ* represents a threshold to determine the acceptability of an identifying point.

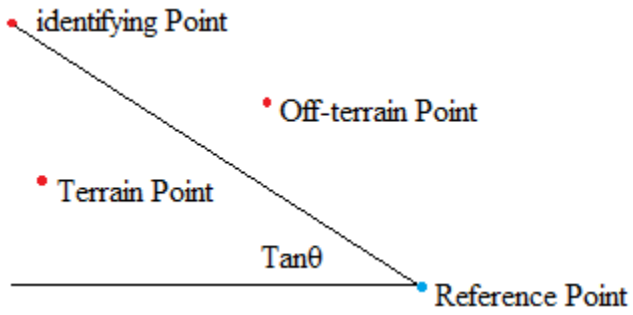
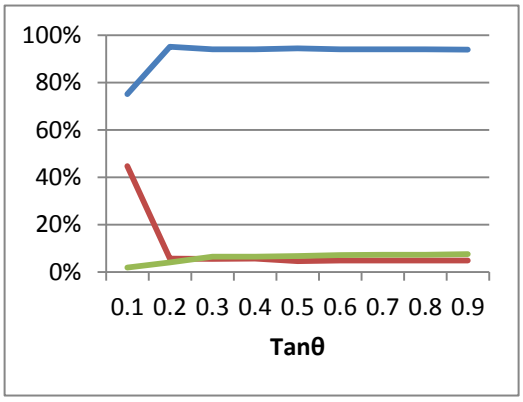


Figure 4-28 Tanθ

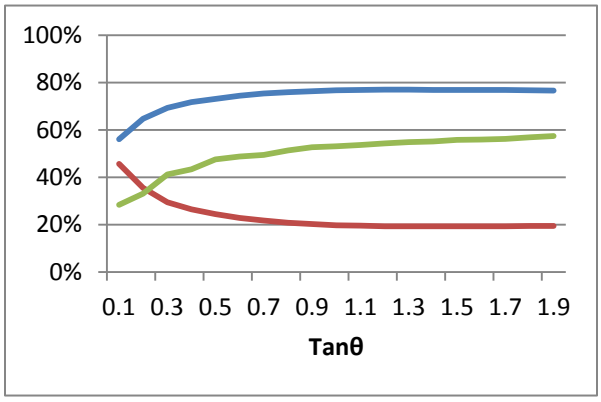
It is obvious that when the *Tanθ* becomes larger, there will be more points (restricted to the points meet the layer number requirement) identified as terrain points. The range of this parameter is determined by the slope gradient. Usually the slope gradient of the test sites is from 0 degree to 45 degree, therefore, the range of *Tanθ* can be chosen from 0 to 1. But for the testing purpose, the experiment range of *Tanθ* is from 0 to 2 since the slope gradient is usually less than 60 degree.

As shown in **Figure 4-29**, the sensitivity of *Tanθ* varies in different scenarios. Flat terrain and gentle slope as shown in **Figure 4-29** (a) and (d) have a similar situation, which is type I error drops along with the increase of *Tanθ*. This is because when *Tanθ* is too small, the requirement is to restrict to accept even real terrain points. However, type II error rate is

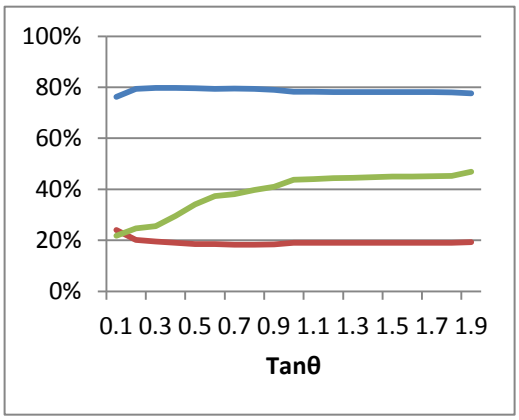
stable. The difference between flat terrain and gentle slope is that after *Tanθ* reaches a certain value, type I error rate stops dropping in flat terrain as *Tanθ* is high enough to cover most terrain points. This value varies in different sites, and it can be very small to be neglected.



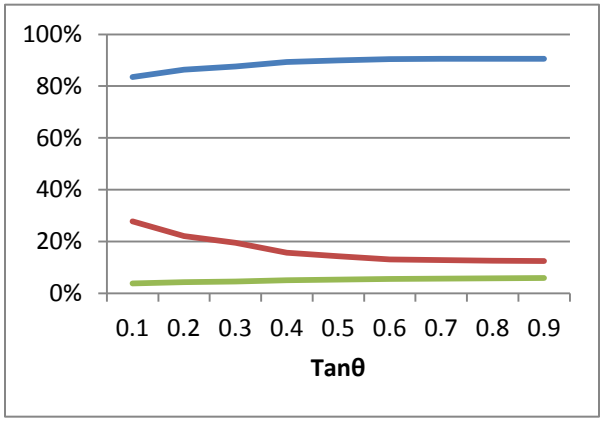
(a) flat terrain (Sample 31)



(b) Steep Slope (Sample 52)



(c) Flat terrain with low objects (Sample 71)



(d) Gentle Slope (Sample 23)

Figure 4-29 Sensitivity of Tanθ in different situation

When it comes to steep slope as shown in **Figure 4-29** (b), type I error drops the same as on gentle slope, while type II error rate increases considerably with *Tanθ*, which means the off-terrain object on the slope is sensitive to this parameter. Another situation is flat terrain

with low objects as shown in **Figure 4-29** (c). Since low objects have a relatively small difference to the terrain, they are easy to be wrongly identified as terrain when *Tanθ* increases, which leads to an unstable status of type II error. To conclude, *Tanθ* is stable on flat terrain, and it is sensitive to the steep slope and low objects.

4.4.5 Identification Tolerance

Identification tolerance is defined as the highest acceptable layer difference in terrain points' identification during the multi-scale terrain filtering. In the high levels (levels close to the top level), the identifying points and reference point may have a long horizontal distance. Thus, the terrain points may have higher layer numbers than the reference points have. If the identification is always based on the same layer number, many terrain points may be identified as off-terrain points, and these wrongly identified points will influence their neighboring points in the following identifications as well. Parameter *Identification Tolerance* is set to fix this problem. This parameter is applied as **Equation** (4-7). The identification tolerance varies according to the *Number of Levels*.

Acceptable Layer Number = INT (level number * Identification Tolerance) (4-7)

The range selecting of this parameter is based on the number of levels and number of layers generated. For example, if 7 levels and 20 layers generated, the *Identification Tolerance* should definitely smaller than 20/7. And in the top level, there should at least have two

layers accepted, which means the *Identification Tolerance* should bigger than 2/7.

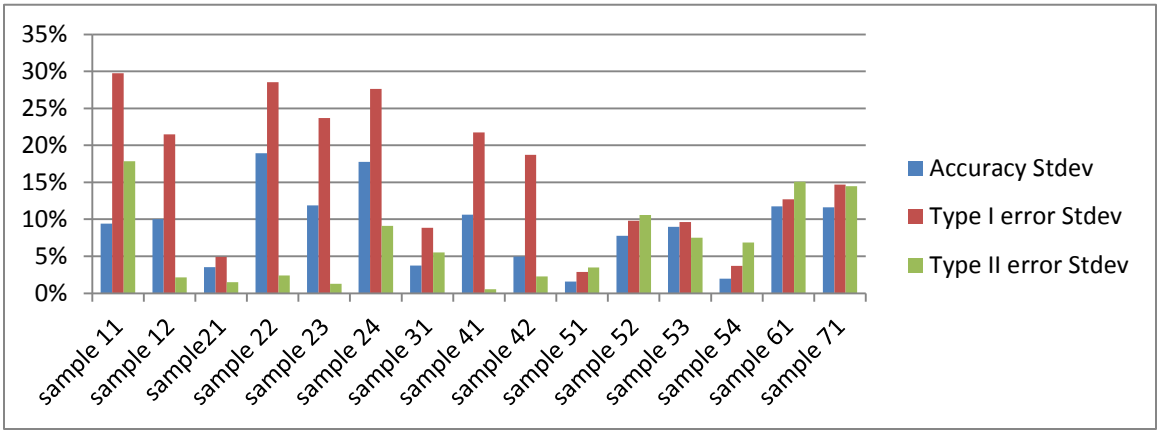


Figure 4-30 Standard Deviations of overall accuracy, type I and type II error rates by testing different *Identification Tolerance* in 15 Samples Sites

Same as *Tanθ*, the increase of this parameter will allow more points identified as terrain points, which leads to a drop of type I error and an increase of type II error. However as shown in **Figure 4-30**, the sensitivities of *Identification Tolerance* in different sample sites varies. For example, type I error is quite stable in sample 21, 51 and 54, but in sample 11, 22, and 24 the standard deviation of type I error rate is very high. type II error also has stable samples like sample 12, 21, 41 etc. and unstable sites such as sample 11, 61 and 71. Therefore, overall the performance of parameter Identification Tolerance is unfortunately not reliable. However the change of the results becomes a more visible when tweaking this parameter.

4.4.6 Other Parameters

There are three more parameters which are *Minimum Layer*, *Minimum Cell* and

Classification Tolerance. *Minimum Layer* and *Minimum Cell* are used in pre-processing to eliminate outliers and single points. These two parameters are not applied in every experimental site; they are only applied and tweaked in those sites where the outliers and single points affect the performance of the proposed method. This means the function of this parameter is very specific to the certain sites, therefore the sensitivities analysis are not applied to them. The parameter *Classification Tolerance* does not apply in any of the experimental site, which is only a debug tool in the developing period. Therefore, no sensitivity analysis applied as well.

For these parameters, some of them perform very stable in every sample site such as *Number of Levels*. More parameters like unit cell size *K*, slope gradient $Tan\theta$ are stable in certain range of value and certain sample sites. There is only one parameter *Identification Tolerance* which is hard to predict the performance.

4.5 Chapter Summary

This chapter gives the results and the analysis of the results of the proposed Multi-scale Terrain Filtering (MTF) method. A Dataset with fifteen samples from ISPRS and two complement datasets are used in the experiments. A cross-matrices analysis method is used to estimate the identified terrain points on the ISPRS data, while a sampling estimation method is adopted in analyzing the generated DTMs. The results show that the proposed

method perform as well as other comparing methods when handling flat terrain or terrain with gentle slopes. Steep slopes and very large data are two major difficulties of the proposed method. However, the processing time is very fast compared to the method provided by the software ALDPAT Version 1.0. To evaluate the stability of the results, a sensitivity analysis is carried out as well, which come to a conclusion that the sensitivity of the result to the parameters are usually stable in certain ranges, but the results are sensitive to some parameters like *Width*, *Delta* , cell size *K*, and *Identification Tolerance* even when these parameters are in rational ranges.

Chapter 5. Conclusions and Recommendations

This chapter gives the conclusions including the summary of the proposed method in Section 5.1 and recommendations for the future work in Section 5.2.

5.1 Conclusions

In this thesis, an automatic method has been developed to generate DTM from last return points of high resolution airborne LiDAR point clouds data. This method is based on an algorithm called Multi-scale Terrain Filtering. It achieved good results in flat terrain areas in terms of evaluation accuracy and promising computational efficiency.

Usually filtering and interpolation are the two steps of DTM generation. The difficulties and research frequently lie on the filtering of terrain points, especially on the filtering in complex situation of the study area, such as outliers, complex objects, vegetation, etc. To overcome these difficulties, several methods were applied in the proposed approach. Since most multiple returns occur in the vegetation area and the edge of the buildings, the method utilizes last return to eliminate some vegetation points. In order to eliminate outlier and noises, applying minimum point number in cells and layers was motivated by the facts that the outlier and noises usually appear by themselves. Another fact is that the fixed filter window size cannot satisfy both large building situations and big slope

situations. Therefore, the proposed method applied Multi-scale Terrain Filtering to handle complex objects.

The developed method consists of three main steps. In the first step pre-processing, outliers and noise points are eliminated. The method separates the point clouds into several layers based on the distribution of the elevation value of the points, and layer numbers are assigned to the points. The second step is the Multi-scale Terrain Filtering. Rasterized pyramid levels are generated from the lowest points in each cell. Then a series of iterative identifications and interpolations are processed to generate a rough DTM. The identification is comparing the layer numbers of the points with the reference points. The interpolation replaces the layer number and height value of off-terrain cells by the average value of their neighbors. The last step is to refine the DTM. The terrain points are adjusted by comparing with the nDSM and then separated from the original data by the generated rough DTM. By using these identified terrain points, an IDW interpolation is processed to produce a final DTM.

To verify the effectiveness of the developed method, two groups of experiments are carried out. The first group of tests is using ISPRS datasets with eight study sites and fifteen samples. ISPRS also provided the results of eight existing algorithms for this dataset. The result of the proposed MTF method indicates that it works as well as other filters in the flat terrain or terrain with gentle slopes. The proposed method can also

overcome the difficulties like bridges, complex scenes, outliers and vegetation. However, the performance of the proposed method drops very much when handling the steep slope or discontinuities of the terrain. The total accuracy of the proposed method can be higher than 90% in some samples; however, it can be as low as around 65% in one study site. And the average kappa coefficient in all fifteen study site is 61.2%, which is low than average performance of all tested algorithms.

A UW Campus LiDAR dataset and the TopoSys Demo City data were applied in the second group of experiments. The data cover many different scenarios, for the convenience of the research, the data were cut into smaller pieces and typical sites such as campus areas, forest areas, residential areas, and urban areas were selected. A comparison between the proposed method and two existing filters, morphological filter and adaptive TIN filter, indicates that identified terrain point images of all the methods have some missing areas, especially in the entire Demo City data. The missing areas of the developed method are usually around the off-terrain points in a rectangular shape, which had limited effect on the result.

The evaluation results of the developed method were in the same level as the other two filters in processing UW campus LiDAR data, and better results were generated from the Demo City experimental sites. A good result is achieved in an urban site, and the average error, standard deviation, and worst error are 0.5 cm, 8 cm, and 57.87 cm respectively.

Since the comparison of the layer numbers the identification is very fast, the calculating speed of the developed method is three times more than the other two filters. This is a very important factor in transferring the algorithm into commercial applications.

The sensitivity analysis of the parameter of the proposed method is carried out as well. The result shows that the variation of most parameters will change the result in a certain range. Some parameters like *Number of Levels*, slope gradient *Tanθ* are pretty stable in most study sites. However, parameters like *width, Delta, Identification Tolerance* have more influence to the result. And sensitivity to the type I and type II error also varies according to each parameter and feature of the study sites.

Overall, this study has developed a method of generating DTM from airborne LiDAR point clouds. However, the proposed method has difficulties in processing steep slope area, very large data. Noticeably, the developed method has a faster speed with a similar level of errors than the two existing algorithms. But the result of the proposed method is not very stable. Although this study did not retrieve higher accuracy result than some of the existing algorithms, a global pre-analysis of the data (layering) is tested and applied in the proposed method and generate similar accuracy result as other algorithms in flat terrain. It proves the utility value of layer feature and the feasibility of statistical classification the LiDAR data.

5.2 Recommendations for Future Research

The proposed method produces good results in some flat terrain areas. However, it does not work perfectly in many other cases, e.g. steep slope, very large data, unimodal height histogram. Therefore, improvements are still need to be work on to develop a good algorithm.

First, the parameters of each site are different, which means the DTM generation still requires manual interaction. The adjustment of the parameters is required to get better results. However, the uncertainty of the parameter combinations reduces the algorithm robustness, and the sensitivity analysis shows the results are sensitive to some parameters. Therefore, an automatic generation of the parameters will be a great feature to be developed in the future. Some of the parameters can be further analyzed to find the relation between optimal value and the dataset. For example, the unit cell size K is relevant to the point density and the size of the biggest off-terrain objects. To change a layering method might stabilize the result sensitivity to the parameter *Width* and *Delta*.

Second, in the experiments handling the entire Demo City data, the evaluation result is not as good as it is in other subsections of dataset. This is because in the large image, the height histogram will represent a large area, and more points will be counted, which will decrease the representativeness of the histogram by mixing too many points. This will

eliminate the representativeness of the features in the image. This means the robustness of the developed method is challenged by dealing with large data and complicated scenarios. Adjusting the parameters by a pre-analysis of the trend of the data and dividing the image into small pieces to process are two potential approaches to overcome this problem.

Third, the steep slope problem makes the result of many ISPRS samples hardly acceptable. The reason of that is the proposed method is based on an assumption that terrain and off-terrain objects are well separated in the space. The layering method is also designed based on that. Therefore, in order to overcome this filtering difficulty, some progressive densification filter and slope based filtering method can be good reference.

Fourth, besides the reference points generated from lowest points in cells, other reference points can be generated by using the intensity data. It should be noted that pulses reflected from asphalt surfaces, which are most commonly roads, will have a different intensity than other data. This will allow easy determination of road points. Since roads are considered ground points it would consequently determine additional reference ground points based on the certainty of the road points.

In addition, using the reference layer number to replace the identified off-terrain points in interpolations will cause a problem. The visual rectangular missing area of the terrain

point images is a display of this problem. Recalculating the layers of the data and fitting these interpolated points into a new layer might be a solution to this problem.

In conclusion, based on the Multi-scale Terrain Filtering algorithm, the developed method has proven to be efficient. The accuracy is depending on the study sites and the parameter setting. The modifications listed above, could be a trial to improve the result of the developed method.

References

Akel, N. A., and Kremeike, K. (2005). Dense DTM generalization aided by roads extracted from LiDAR data. *ISPRS WG III/3, III/4, V/3 Workshop "Laser Scanning 2005"*, Enschede, the Netherlands, 54-59.

Akel, N. A., and Zilberstein, O. (2003). Automatic DTM extraction from dense raw lidar data. *Proceeding of FIG Working Week,* Paris, France, 1-10.

Akel, N. A., and Zilberstein, O. (2004). A robust method used with orthogonal polynomials and road network for automatic terrain surface extraction from LiDAR data in urban areas. *International Archives of Photogrammetry and Remote Sensing,* Vol. 35, ISPRS, 274-279

Ali, T.A., (2010). Building of Robust multi-scale representation of Lidar-based digital terrain model based on scale space theory. *Optics and Lasers in Engineering*, 48:316-319

Anderson, S. (2010). An evaluation of spatial interpolation methods on air temperature in phoenix, az. Retrieved from http://www.cobblestoneconcepts.com/ucgis2summer/anderson/anderson.htm

Axelsson, P. (2000). DTM generation from laser scanner data using adaptive TIN models. *Int. Arch. Photogram. Remote Sens*., pt. B3, vol. 33, pp. 85–92

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

Baltsavias, E. P. (1999). A comparison between photogrammetry and laser scanning. *ISPRS Journal of Photogrammetry & Remote Sensing*, (54), 83–94.

BC-CARMS. (2006). *LiDAR – overview of technology, applications, market features & industry.* Unpublished University of Victoria, Victoria, BC.

Beraldin, J., Blais , F., and Lohr, U. (2010). Laser scanning technology. In G. Vosselman, and H. Maas (Eds.), *Airborne and Terrestrial Laser Scanning*, CRC Press, pp.1-39

Briese, C. (2010). Extraction of digital terrain models. In G. Vosselman, and H. Maas (Eds.), *Airborne and Terrestrial Laser Scanning*. Dunbeath: Whittles Publishing. pp. 147-150

Bunting, P., Labrosse, F. and Lucas, R., (2010). A multi-resolution area-based technique for automatic multi-modal image registration, *Image and Vision Computing*, 28:1203-1219.

Burrough, P. A., and McDonnell, R. A. (1998). *Principles of Geographical Information Systems*. Oxford: Oxford University Press.

Boulaassal,H., Landes, T. and Grussenmeyer, P., (2008). Automatic Extraction of Planar Clusters and their Contours on Building Fa çades Recorded by Terrestrial Laser Scanner, *International Journal of Architectural Computing*, 1(7):1-20.

Chen, Q., and Peng, G. (2007). Filtering airborne laser scanning data with morphological methods. *Photogrammetric Engineering & Remote Sensing Journal of the American Society For Photogrammetry And Remote Sensing*, 73(2): 175-185.

Doneus, M., and Briese, C. (2006). Digital terrain modeling for archaeological interpretation within forested areas using full-waveform laser scanning. *The 7th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST*,Zypern, 569-579

Elmqvist, M. (2002). Ground surface estimation from airborne laser scanner data using active shape models. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, XXXIV, Part 3A: 114-118.

Forlani, G., and Nardinocchi, C. (2007). Adaptive filtering of aerial laser scanning data. *ISPRS Workshop on Laser Scanning 2007 and SilviLaser 2007*, Finland, 130-135.

GeoVAR. (2012). *Geo-referencing, ortho-rectification, accuracy statistics, and mapscale equivalencies*. GeoVAR LLC, Wyoming, USA, Retrieved from http://www.geovar.com/orthorectification.htm

Gonzalez, R. C. and Woods, R. E. (2002). *Digital Image Processing*, 2nd ed., Prentice Hall, Upper Saddle River, NJ. 598-600

Gonz âlez, R. C., Woods, R., and Eddins, S. L. (2004). *Digital image processing using matlab*. Pearson Prentice Hall. 255-262.

Guan, H. (2011). Position relations between IMU, GPS and laser scanner. *Unpublished manuscript, University of Waterloo, Waterloo, ON.*

Guan, H., Li, J. and Chapman, M.A. (2011). Urban thematic mapping by integrating LiDAR point cloud with colour imagery, GEOMATICA, 65(4): 375-385.

Guo, Q., Li, W., Yu, H., and Alvarez, O. (2010). Effects of topographic variability and lidar sampling density on several DEM interpolation methods. *Photogrammetric Engineering and Remote Sensing*, *76*(6), 701-712.

Hu, Y. (2003). Automated extraction of digital terrain models, roads and buildings using airborne lidar data. *Unpublished Ph.D.*, The University of Calgary.

Hu, Y., and Tao, C. V. (2005). Hierarchical recovery of digital terrain models from single and multiple return lidar data. *Photogrammetric Engineering & Remote Sensing, 71*(4), 425-433.

Jiang, J., Ming Y., Zhang, Z., Zhang, J. (2005). Point-based 3D Surface Representation from Lidar Point Clouds. *The 4th ISPRS Workshop on Dynamic and Multi-dimensional GIS. September 6-8, 2005, Wales, UK, 1-4*

Keller, Y. and Averbuch, A., (2006). Multisensor image registration via implicit similarity, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):794–801.

Kilian, J., Haala, N., and Englich, M. (1996). Capture and evaluation of airborne laser scanning data. IAPRS XXXI 3, Vienna, 383-388.

Kobler, A., Pfeifer, N., Ogrinc, P., Todorovski, L., Ostir, K., and Dzeroski, 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.

Korupa, O., Montgomeryb, D. R., and Hewittc, K. (March 2010). Glacier and landslide feedbacks to topographic relief in the himalayan syntaxes. *Proceedings of the National Academy of Sciences of the United States of America, 107(12), 5317-5322*.

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*(4), 193-203.

Kraus, K., and Pfeifer, N. (2001). Advanced DTM generation from LIDAR data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* (XXXIV(3/W4): 23-30)

Kraus, K., and Rieger, W. (1999). Processing of laser scanning data for wooded areas. Wichmann Verlag, Heidelberg, 221-231.

Krzystek, P. (2003). Filtering of laser scanning data in forest areas using finite elements. ISPRS Workshop'3-D reconstruction from airborne laserscanner and InSAR data', TU Dresden, Germany, 8-10.

Lackner, M. (2010). *UW LiDAR LAS and derivatives* UW MAD LAB.

Li, Z., Zhu, Q., and Gold, C. (2005). *Digital terrain modeling: Principles and methodology*. Boca Raton, FL: CRC Press.

Liadsky, J. (2007). Recent advancements in commercial lidar mapping and imaging systems. *Informally published manuscript, Optech Incorporate, Available from NPS*

Lidar Workshop. Retrieved from http://www.nps.edu/academics/Centers/RemoteSensing/Presentations/Lidar Presentations/RecentAdvancements.pdf

Liu, X., Zhang, Z., Peterson, J., and Chandra, S. (2007). LiDAR-derived high quality ground control information and DEM for image orthorectification. *GeoInformatica, 11*(1), 37-53.

Liu, X. (2008). Airborne LiDAR for DTM generation: Some critical issues. *Progress in Physical Geography, 32*(1), 31-49.

Mantis. (2010). History of lidar. *Retrieved from http://xipar.info/historylidar.html*

Masaharu, H., and Ohtsubo, K. (2002). A filtering method of airborne laser scanner data for complex terrain. *International Archives of Photogrammetry and Remote Sensing Commission III,* (Working Group III/3), pp. 175-185

Meng, X., Currit, N., and Zhao, K. (2010). Ground filtering algorithms for airborne LiDAR data: A review of critical issues. *Remote Sensing,* (2), 833-860.

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

Petzold, B., Reiss, P., and Stossel, W. (1999). Laser scanning-surveying and mapping agencies are using a new technique for the derivation of digital terrain models. *ISPRS Journal of Photogrammetry & Remote Sensing, 54*, 95–104.

Pfeifer, N. (2001). Derivation of digital terrain models in the scope environment. *Proceedings of OEEPE Workshop on Airborne Laser Scanning and Interferometric SAR for Detailed Detailed Digital Elevation Models,* Stockholm, Sweden. (Official Publication No. 40.)

Pfeifer, N. (2011). Lidar: Theory, point cloud, DTM. *Informally published manuscript, Fayetteville State University, North Carolina*.

Pfeifer, N., and Mandlburger, G. (2008). LiDAR data filtering and DTM generation. In S. Jie, and Charles K. Toth (Eds.), *Topographic laser ranging and scanning: Principles and processing,* CRC Press, NW, pp. 308-331

Pike, R. J. (1988). The geometric signature: Quantifying landslide-terrain types from digital elevation models. *Earth and Environmental Science Mathematical Geology, 20*(5), 491-511.

Roggero, M. (2001). Airborne laser scanning: clustering in raw data. *International Archives of Photogrammetry and Reomte Sensing*, Volume XXXIV-3/W4 Annapolis, MD, 22-24

Roggero, M. (2002). Object segmentation with region growing and principal component analysis. *in Proc. ISPRS Commission III, Symposium 2002,* Graz, Austria, September 2002, pp. A-289-294..

Rottensteiner, F., Trinder, J., and Clode, S. (2005) Data acquisition for 3D city models from LIDAR extracting buildings and roads. *IGARSS 2005. IEEE International Geoscience and Remote Sensing Symposium, 25-29 July 2005, Seoul, South Korea.* pp. 521-524.

Schuckman, K., and King, B. (2011). Geography 497d: Lidar technology and applications. Retrieved from https://www.e-education.psu.edu/lidar/book/export/html

SCOP++ (2007). The software package SCOP++, http://www.inpho.de/ (accessed 30 March 2012)

Shao, Y., and Chen, L. (2008) Automated Searching of Ground Points from Airborne Lidar Data Using a Climbing and Sliding Method. *Photogrammetric Engineering & Remote Sensing*, Vol. 74, No. 5, May 2008, pp. 625–635.

Shi, W., Li, B., and Li, Q. (2005). Distance image segmentation method based on the projection of the laser scanning point density. *Journal of Surveying and Mapping, 34*(2), 95–100.

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. IAPRS, XXXIV, 3/W4, Annapolis, MD, 203-210.

Sithole, G., and Vosselman, G. (2003a) Automatic structure detection in a point-cloud of an urban landscape. *2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion Over Urban Areas. URBAN 2003 (Cat. no.03EX646); 2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion Over Urban Areas. URBAN 2003, 22-23 May 2003, Berlin, Germany. IEEE; Eur. Space Agency; Eurimage S.p.A.; Geosyst. GmbH; PCI Geomatics Group; FPK Ingenieurgesellschaft mbH,* 67-71.

Sithole, G., and Vosselman G., (2003b). Comparison of filtering algorithms, *Proceedings of the ISPRS Working Group III/3 Workshop: 3-D Reconstruction from Airborne Laserscanner and InSAR data, October, Dresden, Germany, URL:http://www.isprs.org/commission3/wg3/workshop_laserscanning/*

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*(1-2), 85-101.

Sithole, G., and Vosselman, G. (2005). Filtering of airborne laser scanner data based on segmented point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(Part 3/W19), 66-71

Sithole, G., and Vosselman, G. (2006). Bridge detection in airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing, 61*(1), 33-46.

Sohn, G. and Dowman I. (2002). Terrain surface reconstruction by the use of tetrahedron model with the MDL criterion. *Proceedings of the Photogrammetric Computer Vision, ISPRS Commission III, Symposium 2002.* 336–244

Sohn, G., and Dowman, I. (2007). Data fusion of high-resolution satellite imagery and LiDAR data for automatic building extraction. *ISPRS Journal of Photogrammetry and Remote Sensing,* (62): 43-63

Strijbos, J.; Martens, R.; Prins, F.; Jochems, W. (2006). Content analysis: What are they talking about?. *Computers & Education* 46: 29–48.

Toutin, T. (2008). ASTER DEMs for geomatic and geoscientific applications: A review. *International Journal of Remote Sensing Archive, 29(7),* 1855-1875

Tóvári, D., and Pfeifer, N. (2005). Segmentation based robust interpolation – a new approach to laser data filtering. *ISPRS WG III/3, III/4, V/3 Workshop "Laser Scanning 2005",* Enschede, the Netherlands, 79-84.

Vosselman, G. (2000). Slope based filtering of laser altimetry data. IAPRS XXXIII, B3/2, Amsterdam, 203-210.

Wack, R., and Wimmer, A. (2002). Digital terrain models from airborne laser scanner data - a grid based approach. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* XXXIV(3B): 293-296

Wand, M.P; Jones, M.C. (1995). *Kernel Smoothing*. London: Chapman & Hall/CRC. ISBN 0-412-55270-1.

Zhang, K. (2003). A progressive morphological filter for removing non-ground measurements from airborne LIDAR data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(4), 872-882