Automated Extraction of 3D Building Windows from Mobile LiDAR Data

by

Menglan Zhou

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, 2016

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

The three-dimensional (3D) city models have gained more and more attentions because of their considerable potential applications at present. In particular, the demands for Level of Detail (LoD) building models become urgent. Mobile Laser Scanning (MLS) has supplied a brand-new technology in the acquisition and update of 3D information in urban off-terrain features, particularly for building façade details. Accordingly, generating LoD3 building models from MLS point clouds becomes a new trend in recent studies.

As a consequence, a method that can accurately and automatically extract 3D windows from raw MLS point clouds is presented in this thesis. To provide solid and credible information for LoD3 building models, this automated method endeavors to identify window frames on building facades from MLS point clouds. This algorithm can typically be regarded as a stepwise procedure to interpret MLS point clouds as semantic features. A voxel-based upward-growing method is firstly applied to distinguish non-ground points from ground points. Noise is then filtered out from non-ground points by statistical analysis. In order to segment out the building facades, all the remaining non-ground points are clustered based on conditional Euclidean clustering algorithm; clusters whose density and width are over a given threshold will be designated as points for building facades. After a building façade is successfully extracted, a volumetric box is created to contain façade points so that neighbours of each point can be operated. A manipulator is finally applied according to the structural characteristics of window frames to extract the potential window points.

The experimental results demonstrate that the proposed algorithm can successfully extract the rectangular or curved windows in the test datasets with promising accuracies. The 2D validation and 3D validation were both conducted in this study. In the 2D validation, the lowest *F-measure* of the test datasets is 0.740, and the highest can be 0.977. While in the 3D validation, the lowest correctness of the test dataset is 79.58%, and the highest can be 97.96%. After further analysis of the experimental results, it was found that, for those windows concave on walls or with curtains drawn, the performance of the proposed method was influenced. Furthermore, big holes caused by system errors in raw point clouds also had negative impacts on the proposed method.

In conclusion, this thesis makes a considerable contribution to extracting 3D rectangular, irregular and arc-rounded windows from noisy MLS point clouds with high accuracy and high efficiency. It has supplied a promising method for generating LoD3 building models.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Jonathan Li, who offers me not only professional research backgrounds but also life guidance in my master program. Also, thanks to my thesis committee members: Dr. Mike Chapman, Dr. Peter Deadman and Dr. David Clausi; thank you for reviewing my dissertation and providing me the precious comments on my study. Especially, I need to express my thanks for Dr. Peter Deadman, who is on sabbatical leave but still willing to be my thesis reader.

In addition, I need to thank TOLLUCH Engineering, for providing me the raw MLS point clouds and digital images in this study.

I would like to thank my friends and group mates: Haocheng Zhang, who provided me relevant literatures and backgrounds of the building façade segmentation; Weikai Tan, who offered me strong support in generating orthophotos in ArcGIS 10.2.2; Dr. Xiaoliang Zou, who discussed with me about structuring this thesis; and all other friends who support me in life here and in China.

I also would like to thank my family in China: my dear parents, thanks for your financial support, and for encouraging me whenever I need you.

Table of Contents

List of Abbreviations

2D	Two-dimensional
3D	Three-dimensional
ALS	Airborne Laser Scanning
CCD	Charge Coupled Device
CSV	Comma Separated Value
CTP	Conventional Terrestrial Pole
DMI	Distance Measurement Indicator
DOD	US Department of Defense
DTM	Digital Terrain Model
GIS	Geographic Information System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
MLS	Mobile Laser Scanning
IMU	Inertial Measurement Unit
FOV	Field of View
LiDAR	Light Detection and Ranging
LoD	Level of Detail
PCA	Principal Component Analysis
PCL	Point Cloud Library
RANSAC	Random Sample Consensus
TIN	Triangulated Irregular Network
TLS	Terrestrial Laser Scanning

TOF	Time of Flight
POS	Position and Orientation System
WGS84	World Geodetic System 1984
UTM	Universal Transverse Mercator projection

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

1.1.1 Requirements for LoD3 Building Models

With the continual development of three-dimensional (3D) spatial information technology and the proposed concepts of Digital Earth, Smart City, Virtual Reality, Intelligent Transportation Systems, experts' research focus has been gradually shifting from two-dimensional (2D) planes to 3D space (Lemmens et al., 2011). The 3D city model has proved useful for a variety of applications such as telecommunication, urban planning, environmental simulation, cartography, virtual tourism, and mobile navigation (Yang et al, 2013). Currently, considerable achievements have been made in related works on the 3D city modeling, and the most high-profile product of which is Google Earth. This system incorporates the global digital terrain model (DTM), global satellite images, 3D building models of some metropolitan areas, and spatial information on roads, addresses, administrative borders as well as special attractions, which come together to create the Digital Earth (Google, 2016). One of the most fundamental databases of Digital Earth is 3D building models.

Buildings are major urbanized bodies. Reconstructing building models has become more and more urgent with the continuously enhanced requirements of achieving more detailed features on building facades in user scenes (Cheng et al., 2011). Commercial enterprises such as Google, Sony, Apple and Nokia have invested enormous manpower and technical resources into realistic 3D city modeling (Zhu & Hyyppa, 2014). Ample texture information and complex 3D geometric structures of buildings result in contradictions between the realistic presentation of 3D scenes and computer performances (Haala & Kada, 2010). Therefore, the ability to

reconstruct building models of large scenes efficiently becomes urgent. The details of 3D building moedels can be described by the Level of Detail (LoD). A significant technical feature in the digital GIS is to support the concept of multiple levels of feature details (Over et al., 2010). The concept of LoD was firstly put forward by Clark in 1976 (Guercke et al., 2011). As shown in Figure 1-1, the 3D LoD building model can be basically separated into four levels: LoD1 describes block models or roof structures, LoD2 describes building models with texture and roof models, LoD3 describes detailed architecture models with external doors and windows, and LoD4 describes interior models including rooms, stairs, windows, doors, furnishings, and so on (Over et al., 2010).

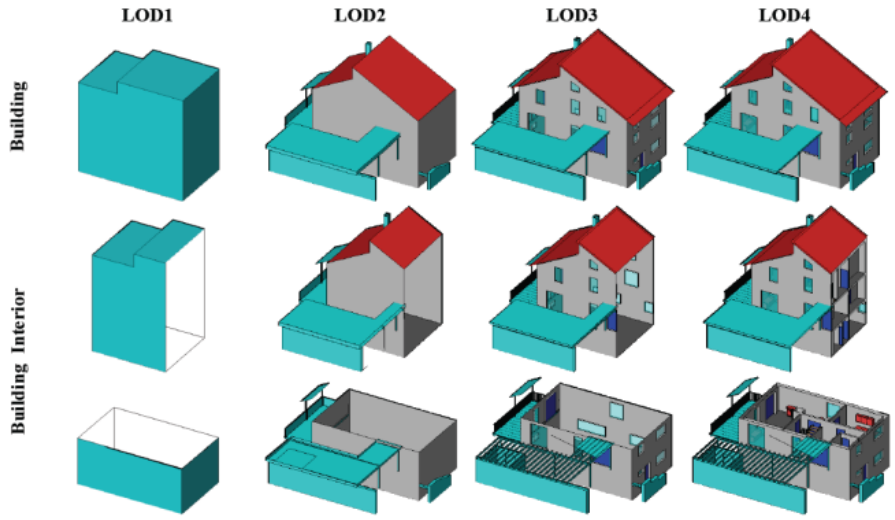


Figure 1-1 LoD building models (SimStadt, 2016)

1.1.2 Advantages of MLS Point Cloud in LoD3 Building Modeling

In traditional aerial photogrammetry, building rooftops are almost the only part of a building, which are visible in aerial photographs. Therefore, the algorithms for the 3D building modeling start with the extraction of the roof's geometric characteristics and then extend them by integrating the walls to form complete 3D building models. One of the most widely applied

approach of 3D building modeling at present is to use aerial digital images to identify footprints and corners of buildings. This approach can acquire accurate 3D geometric information of buildings, however, it still demands abundant manual work to ensure the accuracy. (Sahin et al., 2012; Demir & Baltsavias, 2012). In sum, because of the high complexity and diversity of building geometric properties, the building modeling depending on aerial images is not a fully automatic process so far.

In recent years, with the continuous promotions of laser scanning as well as integrated position and orientation technology, the quick acquisition to high-quality 3D information of realistic scenes is capable of being realized (Yu et al., 2015). Currently, precise 3D point clouds can be acquired through airborne laser scanning (ALS) systems, mobile laser scanning (MLS) systems and static terrestrial laser scanning (TLS) systems (Hyyppä et al., 2013). Each laser point not only contains factual geographic coordinate information, but also records laser reflective intensity of feature surfaces. Laser imaging technology uses active non-contact measurements to directly get access to the information of the scanned 3D surfaces, the data acquisition depends little on illuminations. It also has the capability of recording real-time point cloud data at a high scanning speed precisely and actively (Pu & Vosselman, 2009; Zhu & Hyyppa, 2014). However, 3D building models generated from ALS point clouds can only be regarded as LoD2 models; detailed architecture models such as windows and doors cannot be built from ALS point clouds. Therefore, TLS point clouds and MLS point clouds become the optimal choices in generating LoD3 building models.

Nowadays, MLS point clouds are more applicable than TLS point clouds in a cosmopolitan extent due to its higher flexibility and acquisition rate in large-scaled complex scenes (Guan et

al., 2015). In addition, an MLS system can capture point clouds and close-range images along with the movements of vehicles, which enriches the data sources. In contrast, instruments in TLS systems should be fixed in one position, resulting in the stationary resolution and density of laser scanning point clouds (Jochem et al., 2011; Yang et al., 2012; Aijazi et al., 2014). The whole process to capture a completed street scene is rather time-consuming and demands a great cost in manpower, which highly reduces the adaptability of TLS point clouds (Williams et al., 2013).

1.1.3 Challenges of Window Extraction from MLS Point Cloud

Points of building facades are in high density, and texture information of building facades cannot be directly acquired from 3D point clouds (Puente et al., 2013). Thus, point clouds should be interpreted as various features before LoD3 building modeling. The most challenging work of the LoD3 building modeling from the noisy point cloud is to extract detailed features on facades, such as windows, doors, and remove holes caused by systematic errors (Haala & Kada, 2010). Windows are indispensable frames on building facades, nevertheless, there are no laser points for window glasses due to the very low reflectivity. In addition, because of aesthetic requirements and cultural diversities, windows are designed in multitudinous shapes (Pu & Vosselman, 2007; Wang et al., 2011). Accordingly, generating a reliable method to extract windows while retain their geometry, semantic and coordinate information precisely from the noisy MLS point clouds has become a colossal challenge in recent years.

This thesis will focus on establishing an effective rationale and developing an automatic algorithm for 3D window extraction from an MLS point cloud according to theories and

methods in geomatics, computer vision, numerical analysis and computational geometry. The method proposed in the thesis is to enable an intelligent processing and modeling of an MLS point cloud with the minimal manual intervention. This study will improve the level of automation in the intelligent interpretation and the 3D window extraction from MLS point clouds. It will also extend engineering applications and promote practicalities of MLS data in mobile mapping, digital heritage management, surveying, and geomatics.

1.2 Objectives

The objective of this thesis is to develop a semi-automatic approach for the creation of the LOD3 building models in typical Canadian residential houses, in particular, to extract building windows from large-volume mixed-density, unstructured MLS point clouds.

The specific objectives of this thesis are:

1) To develop a semi-automatic algorithm for building facades segmentation from a noisy MLS point cloud to solve challenging problems in feature classifications and volume data processing of MLS point clouds.

2) To develop a semi-automatic algorithm for 3D window extraction by semantically and topologically analyzing characteristics of segmented building facades, as well as to recover topological and geometric relationships of detailed features (including windows, doors, and façade outlines) on facades.

3) To develop a solid accuracy assessment mechanism to validate the extracted window points and provide fundamental proofs for applications of the proposed method.

1.3 Structure of the Thesis

This thesis is extended from the successive chapters:

Chapter 2 presents a literature review about current related works on MLS systems. The chapter starts by introducing the working principle of MLS systems. The main components and the properties of current commercial MLS systems are described, and a working principle about how to generate geo-referencing 3D point clouds together with digital images in MLS systems is presented. The chapter continues by introducing the concepts of the WGS84 coordinate systems, the UTM projection, and feature extraction. This chapter ends with introducing historical related works on building façade extraction and window extraction from MLS point clouds.

Chapter 3 elaborately describes the methodology applied in this thesis. This chapter extends into three main sections. Firstly, the study area, test datasets, the programming platform, the related software, and the workflow are presented. This chapter continues by introducing the stepwise data processing algorithms in detail, including the voxel-based upward-growing algorithm, the statistical analysis, the conditional Euclidean clustering, the density/width analysis, and the hole detection algorithm. Finally, an accuracy assessment mechanism is described to validate the proposed method.

In Chapter 4, the experimental results are presented and discussed.

And finally, Chapter 5 summarizes the contributions and limitations of this thesis; also, recommendations of future works are put forward.

Chapter 2 Background and Related Work

This chapter comprises a literature review of backgrounds and related works of this study. Firstly, a brief introduction about mobile laser scanning technology is presented, and basic components of an MLS system are described. Then detailed information of RIEGL VMX-450 system is introduced. Furthermore, this chapter reviews the most state-of-art semi-automatic and automatic algorithms in building façade reconstruction and window extraction. By introducing a brief information of the MLS system and reviewing the historical related technology of building façade reconstruction and window extraction, this chapter elaborately offers an appropriate theoretical background in understanding algorithms of the methodology in this study.

2.1 Introduction to Mobile Laser Scanning

2.1.1 What is Mobile Laser Scanning

Mobile laser scanning is defined as a mapping technology used to obtain 3D information of features by laser scanners mounted on mobile vehicles (Haala et al., 2008; Marshall, 2011). MLS systems utilize near-infrared laser spectrums to scan surfaces of objects by constantly emitting laser beams. They also record 3D coordinates and intensity properties of scanned points simultaneously when the laser energy returns from the features (Lemmens et al., 2011; Olsen, 2013; GIM, 2013). Compared with conventional optical imaging systems, MLS systems can access highly qualified 3D information of wide-range scenes, and record the information in high-density and high-precision point clouds (Brenner, 2009; Yen et al., 2011). In addition, active laser measurements are applied in MLS systems so that data acquisition will not be

influenced by any ambient lighting conditions; thus, field work can be undertaken in both the day and the night (Zhu et al., 2011). The main characteristics MLS point clouds can be summarized as follows:

(1) Point density. The MLS system can obtain 3D point clouds with high accuracy, high density and rich details. For instance, RIEGL VMX -450 system can collect 7000-8000 pts/m^2 (Schrock, 2013);

(2) Noise. MLS systems usually generate measuring disturbances in different directions caused by acceleration, deceleration, direction shift and road undulation. Additionally, mechanical perturbations of vehicles and disturbances of the MLS platform also occur in the measurements.

(3) Intensity information. Most current commercial MLS systems can record energy of laser pulses reflected from the scanned objects. Therefore, intensity information of the scanned surfaces can be stored in 3D point clouds.

(4) Scene complexity. MLS systems is capable of obtaining high-density point cloud in urban areas. The acquired 3D point clouds contain ample features, such as low shrubs, fences, poles, traffic signs, bus stops, vehicles, pedestrians, road curbs, flower beds, and buildings. Therefore, the complex scenes and diverse objects in 3D point cloud lead to difficulties in automatic feature classifications and recognitions.

(5) Occlusions. Due to the complex scenes in urban areas, detailed information on building facades is often occluded by objects, such as trees, poles, traffic signs, etc. Occlusions often lead to incomplete data acquisition on buildings and loss of facade details.

2.1.2 Components of MLS Systems

MLS systems are usually carried by various vehicles and in synchronized control of multiple sensors. Typically, an MLS system consists of six main portions (as shown in Figure 2-1): laser scanners, optical cameras, a GNSS, an IMU, a DMI, and a control system for data synchronization. The GNSS, the IMU, and the DMI compose a position and orientation system (POS).

Figure 2-1 Basic components of an MLS system

1) Laser Scanner

There are distinguishable quantities and configurations of laser scanners in different MLS systems. Laser scanners scan surfaces of features by intermittently emitting continuous waves or laser pulses. Two principle techniques are applied in the plurality of MLS systems for range measurements: time-of-flight (TOF) and phase shift (Pu et al., 2011; Puente et al., 2013).

A TOF scanner first launches a short laser pulse to a target, and the distance (d) from the target to the MLS system can be determined by calculating the time interval between pulse

transmission and reception when the pulse is reflected back from the target (Puente et al., 2013). The distance *d* can be computed by the following expression:

$$d = \frac{1}{2} c\Delta t \qquad 2\text{-}1$$

in which, *c* is the speed of light and Δt is the time interval between pulse transmission and reception. At present, the majority of MLS systems are equipped with TOF scanners since the TOF scanner can reach a further range over phase-based scanners (Brenner, 2009). On the contrary, phase-based scanners transmit a continuous amplitude-modulated wave to a target at first, and the distance (*d*) from the target to the MLS system can be achieved by calculating the phase shift between signal transmission and reception (Puente et al, 2013). The distance *d* can be computed by the following expression:

$$d = \frac{\Delta\varphi}{2\pi}\frac{\lambda}{2} + \frac{\lambda}{2} n \qquad 2\text{-}2$$

where φ is the phase shift, λ is the modulated amplitude, *n* is the number of transmitted waves. Phase-based scanners can achieve higher ranging accuracy but reach shorter range than the TOF scanners; the range of a phase-based scanner in recent commercial MLS systems is about 100 meters (Lichti, 2010).

2) Optical Digital Camera

Different MLS systems are equipped with different numbers and types of optical cameras to capture auxiliary texture information in various scenes (Zhou & Vosselman, 2012). For example, a RIEGL VMX -450 MLS system carries four digital cameras simultaneously. The original 3D point cloud lacks sophisticated texture information; nevertheless, texture information obtained by digital cameras can be superimposed upon the 3D point cloud to

generate a color point cloud. The generated color point cloud can not only provide an ampler visual effect, but also broadens the scope of point cloud applications, such as 3D games and elaborated 3D object reconstruction.

However, digital images captured by optical cameras have perspective distortions. Perspective distortion refers to a deformation that is caused by changes in relative proportions of distance, resulting that an object and its surrounding areas look completely different in standard lens and in reality (Valente & Soatto, 2015). Perspective distortion is caused by changes in shooting distances and imaging angles, it is indispensable in MLS systems since digital images are captured while the vehicle is moving.

3) Position and Orientation System

As mentioned before, a position and orientation system is composed of GNSS, IMU and DMI. GNSS mainly provides three observations: time, location and speed. Despite the fact that GNSS can afford highly accurate positioning information in broad vision, its positioning accuracy will decline when satellite signals are blocked by tall trees, buildings or other objects. IMU can offer orientation information on three directions, including acceleration and rotation angles from three axes, and its working process will not be influenced by magnitude of satellite signals. However, with the accumulation of time, time drift will appear in IMU that the positioning and orienting accuracy will decrease (Puente et al., 2013). Accordingly, the combination of GNSS and IMU is able to provide more accurate position and orientation information for MLS systems. Positioning accuracy of GNSS can be enhanced by IMU when GNSS is under poor satellite signals, and time drift of IMU can be compensated by positioning information of GNSS. Typically, DMI is fixed on the left rear wheel of a car to measure

travelling distances of the vehicle by recording the number of rotations of the wheels. DMI is able to give constraints to time drift of IMU especially when satellite signals are poor.

4) Central Control

The main role of a central system is to synchronize and store data acquired by sensors (laser scanners and digital cameras), as well as signals acquired by position and orientation systems (GNSS, IMU, DMI).

2.1.3 Principle of MLS Systems

(1) Working Principle

In RIEGL VMX-450, when the vehicle is moving, the GPS receiver constantly measures instant real-time geodetic coordinate of GPS antenna phase centre. The IMU records instant attitude angles (including direction angle, ie: roll, pitch, and yaw). The laser scanners transmit and receive laser beams by line scanning mode, and keep track of scanning angles and distances between the scanning centre and the scanned points. The CCD cameras or the panoramic cameras get direct access to texture information of features at a certain frequency. By integrating scanning frequency and field of view (FOV) of laser scanners, with scanning angles and distances from the scanning centre to the scanned points, coordinates of scanning points in the laser scanner coordinate system can be computed. Then through the positional relationship between GPS, IMU and laser scanners, coordinates of scanning points in World Geodetic System 84 (WGS 84) can be obtained through coordinate transformation (Lichti, 2010; Pu et al., 2011; Zhou & Vosselman, 2012; Puente et al., 2013). Therefore, real-time 3D information of roads and off-road features can be accessed.

(2) Direct geo-referencing and scanned parameters

Figure 2-2 shows the principle of the direct geo-referencing of MLS systems. The scanning angle α and the scanning range d of a scanned point P are measured by laser scanners. By coordinate system transformation, the coordinate of point P in the mapping frame can be calculated from the centre of the GPS antenna.

Parameters used in the calculation are listed in Table 1, thus the coordinate of point P can be derived by the following equation (Guan, 2013):

$$\begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix} = R_M^{IMU}(\omega, \varphi, \kappa) \cdot (R_{IMU}^S(\Delta\omega, \Delta\varphi, \Delta\kappa) \cdot r_P^S(\alpha\ d) + \begin{bmatrix} l_X \\ l_Y \\ l_Z \end{bmatrix} + \begin{bmatrix} L_X \\ L_Y \\ L_Z \end{bmatrix}) + \begin{bmatrix} X_{GPS} \\ Y_{GPS} \\ Z_{GPS} \end{bmatrix} \quad 2\text{-}1$$

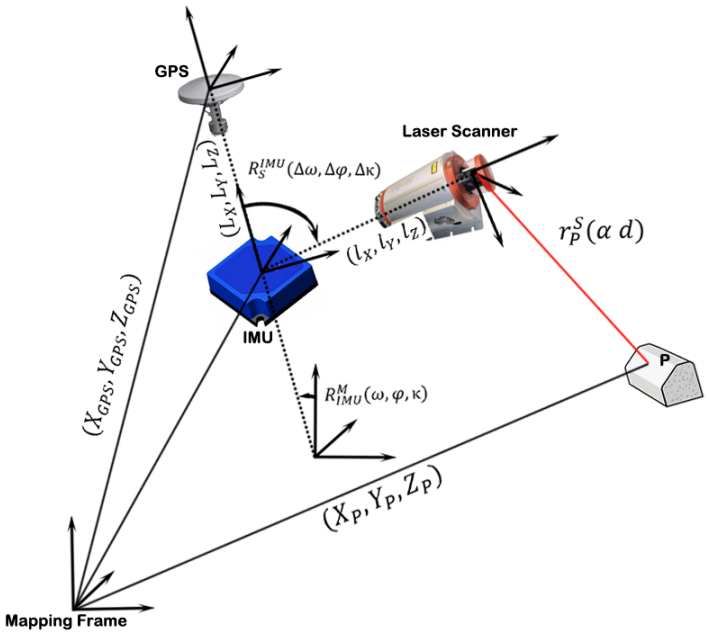


Figure 2-2 Principle of direct geo-referencing (Zhang, 2016)

Table 1 Parameters used in direct geo-referencing

Parameters	Description
$\boldsymbol{X_P, Y_P, Z_P}$	Coordinate of point P in the mapping frame
$\boldsymbol{X_{GPS}, Y_{GPS}, Z_{GPS}}$	Coordinate of the phase centre of the GPS antenna in the mapping frame
$\boldsymbol{R_M^{IMU}(\omega, \varphi, \kappa)}$	Rotation matrix from IMU coordinate system to the mapping frame. ω, φ, κ are roll, pitch and yaw of IMU with respect to the mapping frame
$\boldsymbol{r_P^S(\alpha\ d)}$	Relative position of point P in the laser scanner coordinate system
$\boldsymbol{L_X, L_Y, L_Z}$	The offsets from the origin of GPS to the origin of IMU
$\boldsymbol{l_X, l_Y, l_Z}$	The offsets from the origin of IMU to the origin of the laser scanner coordinate system
$\boldsymbol{R_{IMU}^S(\Delta\omega, \Delta\varphi, \Delta\kappa)}$	Rotation matrix from the laser scanner coordinate system to the IMU coordinate system. $\Delta\omega, \Delta\varphi, \Delta\kappa$ are bore sight angles which align the scanners with the IMU

2.1.4 RIEGL VMX-450

Test datasets used in this thesis were acquired by a RIEGL VMX-450 system from Tulloch Engineering, an Ottawa based geomatics company. A RIEGL VMX-450 system consists of two fully integrated and calibrated RIEGL VQ-450 laser scanners, four RIEGL VMX-450-CS6 optical cameras with pixel array is 2452H×2056 V, one Global Positioning System (GPS) receiver, an IMU and a DMI. The pulse repetition of two laser scanners can be 1.1 MHz for high-resolution mobile laser scanning in urban areas, 600 kHz for medium range applications, and 300 kHz for long range applications. The maximum range of a single laser scanner is 800 m, and the highest precision of this system can achieve 5 mm (1σ). As shown in Figure 2-1,

two laser scanners of RIEGL VMX-450 are fixed on both sides of the principal axis at an elevation of 45˚, and one single laser scanner can provide 360˚ profiles at a pulse frequency of 550,000 measurements/sec (1,1 million measurements/sec for two laser scanners). According to RIEGL Datasheet Introduction, the speed of the vehicle was about 20-30 km/h and the pulse repetition rate was about 1.1 MHz in this study (RIEGL, 2015).

RIEGL VMX-450 has its own data processing software (RiPROCESS) to transform laser scanner raw data to 3D point clouds in WGS84 or UTM (Universal Transverse Mercator). The workflow of RiPROCESS is shown in Figure 2-3. Two different kinds of raw laser scanner data are separately input into the software modules, RiANALYZE and SDCImport, to be transformed into laser scan data in the scanner's own coordinate system. Then the processed laser scan data and the mounting orientation are input into the software module, RiWORLD. Then by combing information in the laser scan data and the trajectory data, laser point clouds in WGS84 or UTM can be generated.

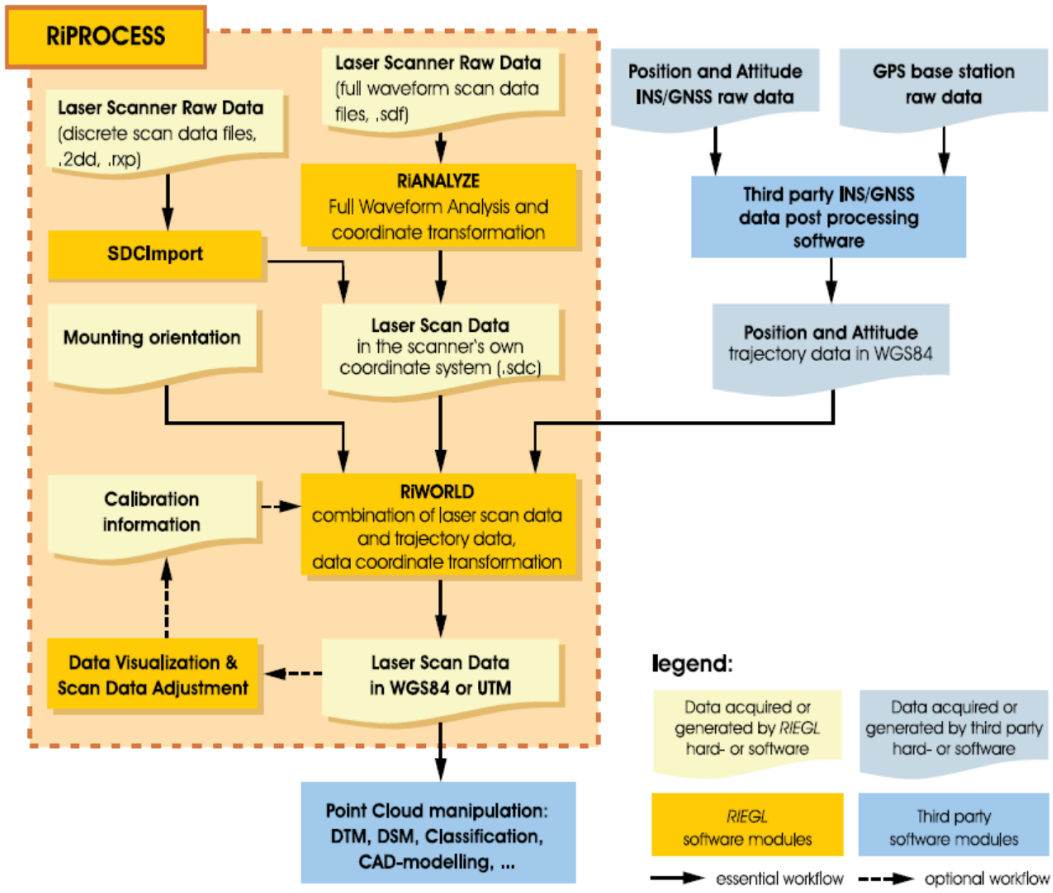


Figure 2-3 Workflow of RiPROCESS (RIEGL, 2014)

2.2 Related Concepts

2.2.1 WGS84 and UTM

The standard used in the GPS of RIEGL VMX-450 is WGS84, which is a protocol reference system. The origin of the WGS84 is the earth's centre of mass. Its Z-axis points to Conventional Terrestrial Pole (CTP), which was determined by the International Time Bureau (BIH) in 1984. Its X-axis points to the intersection of the BIH Zero Meridian (epoch 1954.0) and the equator corresponding to CTP. And its Y-axis completes a right-handed coordinate system with the X-axis and Z- axis. WGS84 coordinate system was originally established by

the US Department of Defense (DOD). The WGS84 was firstly regarded as the coordinate reference datum of GPS satellite broadcast ephemeris in January, 1987 (Leick et al., 2015).

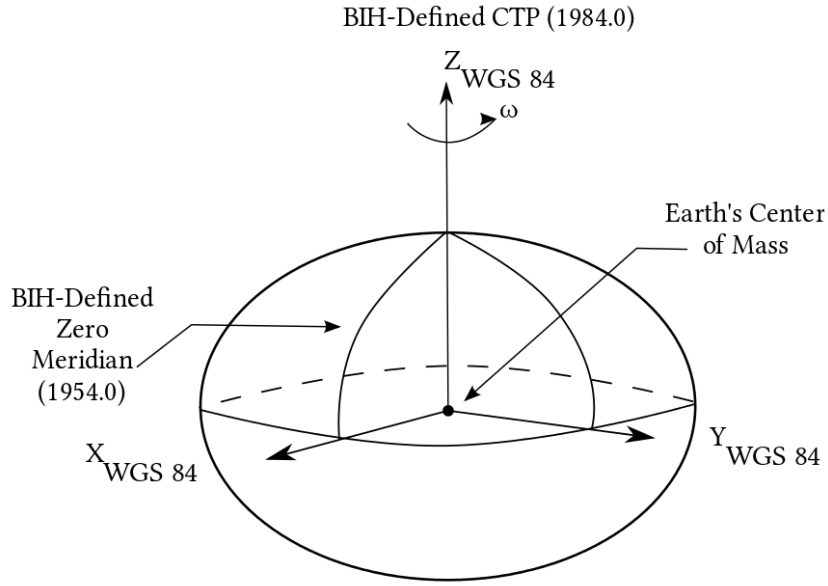


Figure 2-4 WGS4 reference frame (Defense Mapping Agency, 2015)

The plane projection method of 3D point clouds in this thesis is UTM. The UTM projection is a conformal cylindrical projection method, which causes almost no distortion when the projection is completed. UTM is a horizontal position representation which uses a 2D Cartesian coordinate system to specify exact point locations on the surface of Earth. It distributes the Earth into 60 UTM zones, while each zone utilizes a transverse Mercator projection to map a region. The coordinates in each zone can be expressed by X, Y, and Z.

As mentioned in Section 2.1.4, 3D point clouds and trajectory data generated by RIEGL VMX-450 are in WGS84 and UTM. In addition, coordinates in Google Earth are also expressed in WGS84 and UTM (Google, 2016). One of the main objectives of this thesis is to extract 3D window points which can be used in practical applications. Therefore, the object

space coordinate system used in thesis should be WGS84 and UTM in order to guarantee that output 3D window points can be aligned with geo-referenced information in Google Earth.

2.2.2 2D and 3D Feature Extraction

2D feature extraction is a concept from computer vision and image processing. It originally refers to using computer technologies to determine whether an image point belongs to a specific image feature. The result of the 2D feature extraction is to divide image points into different subsets, while these subsets always belong to isolated points, continuous curves and continuous areas (Nixon & Aguado, 2012). The concept of feature extraction is extended to 3D objects nowadays in fields of photogrammetry and remote sensing. Primarily, 3D feature extraction refers to determining whether a point with geospatial information (x, y, z) in 3D data belongs to a certain object in reality (Steder et al., 2011). However, it is more difficult to create and manipulate 3D information than 2D information. Some experts still prefer to solving problems of feature extraction on images by projecting 3D datasets onto 2D planes (XoY plane, YoZ plane or XoZ plane), while the projection usually results in information loss of a certain dimension and accuracy loss of final feature extraction (Pu et al., 2011).

2.3 Building Façade Extraction Using MLS Point Cloud

2.3.1 2D Building Façade Extraction

Some experts have tried to convert 3D point clouds into 2D planes to extract building facades based on the generated geo-referenced images. Arachchige et al. (2012) applied an automatic processing algorithm to detect building facades by choosing intelligent seed points. This method firstly used the random sample consensus (RANSAC) algorithm to achieve

segmented planar patches, and then introduced intelligent seed points and growing rules to detect important specific building façade features, finally, a rule based partitioning tree was applied to recognize the designated features on building facades (Arachchige et al., 2012). In most cases, the large size of MLS point clouds increases the difficulties to be processed automatically. This algorithm decreases the complexity for computing abundant data by separately processing the test point clouds step by step; this advantage of the method remarkably prompts the workflow for handling large MLS data; nonetheless, this algorithm requires abundant façade prior knowledge of geometry and the building façade extraction is completed in a 2D plane. In addition, Yang et al. (2012) proposed an algorithm for automatically extracting building façades from 3D point cloud by generating geo-referenced images. They firstly employed planer division to grid raw point clouds to calculate the weights of points in each cell based on spatial distribution analysis. Then the weighted points were projected onto a geo-referenced image to extract building facades using inverse-distance-weighted interpolation method (Yang et al., 2012). This algorithm highly enhanced the computational efficiency; however, the process of generating geo-referenced images produced some accuracy loss.

Similarly, Zhu et al. (2011) developed an automated algorithm to extract points of buildings in the binary images. The algorithm firstly transformed all the points contained in the top-view in each overlap into binary images and compared the differences between X and Y coordinates in each overlap in the selected cutoff-boxes. The cutoff-boxes were designed to decrease the influences caused by trees close to the buildings. Then a step of standard image processing was introduced to remove all the off-building points on the binary images; afterwards, the

processed binary images were re-transformed into point cloud files (Zhu et al., 2011). This algorithm can be applied not only in extraction of building façades, but also other binary feature extraction. However, this algorithm needs a manual check in the final step to ensure its accuracy. The accuracy relies heavily on different illuminations between images and influences caused by the nearby features, such as trees and pole-like objects. Thus, the degree of automation of this algorithm still needs to be improved.

2.3.2 3D Building Façade Extraction

Points on buildings have heights, thus most methods can use height information to segment buildings out of the whole points cloud. However, the detailed information on buildings is hard to detect and reconstruct. Furthermore, trees or some pole-like objects also contain height information; thus, using the latter steps to distinguish features on buildings from trees or other features containing height information becomes a hot issue. Yang et al. (2013) introduced a semi-automated algorithm to extract building façade footprints by using random sample consensus technique. The method projected all the points on buildings to different planes, and then by using Principal Component Analysis (PCA) algorithm, points on planes were selected to compute their scanning beam angle, start and ending point to each corresponding façade footprint. And finally, the footprints on building facades could be determined. This method can be applied to most detections of building facades; it can effectively decrease the influences caused by noise or other features, but the method does not enable one to achieve desirable accuracy for those buildings that have more complex structures (Yang et al., 2013).

Jochem et al. (2011) also introduced an algorithm based on the coordinate information contained in geo-referenced 3D point clouds. This algorithm used information in a 3D point

cloud to compute the 3D horizons of all the points, and then by computing different slope and aspect angle corresponding to each point, the building facades could be successfully classified. This algorithm also developed a workflow to detect the interior wall and exterior wall of building facades, which could also help to compute the solar energy in the future (Jochem et a., 2011). However, building facades which are not influenced by shadows can be successfully detected, for those buildings under shadows of trees, this algorithm cannot work as expected.

Furthermore, some algorithms are developed not only to detect but also to reconstruct building facades in urban areas. Significant semantic information such as size, positions, topology and shapes, are always contained in features (windows, doors, rooftops, walls, etc.) on facades, and these semantic features can also be utilized to reconstruct building facades. Pu & Vosselman (2009) first proposed an algorithm to reconstruct building facades in a TLS point cloud. And then in 2011, Pu et al. (2011) put forward a semi-automated algorithm to reconstruct building façades by fusing MLS data and images. The method was based on a prior semantic knowledge of features on building facades, such as windows and doors, and Pu set a constraint for each corresponding feature on building facades. The large-sized point cloud was firstly segmented by a growing segmentation algorithm to planar planes. And each segment was processed by the defined constraints to determine the most likely feature (Pu et al., 2011). This method can work well for most features on building facades except for windows, since usually there were no sufficient laser points for windows. Additionally, since this method uses convex hull polygons and concave polygons to extract building facades, which requires ample prior-knowledge about buildings, this algorithm failed in extracting building facades with complex structures, such as curved walls and non-vertical walls.

2.4 Window Extraction Using MLS Point Cloud

Building façade reconstruction is used to establish an outer contour wireframe model according to characteristics of building façades, and then to obtain facets (rooftops, walls, windows and doors) point cloud data after divisions, and finally to extract boundaries (including inner and outer boundaries) on these facets and sketch boundary lines. However, boundaries are often not smooth and coplanar, also they always have confused topological relationships because of potential missing parts in point cloud and inaccurate segmentations (over-segmentation or fake segmentation); therefore, applying relevantly accurate algorithms to adjust extract semantic information (windows, doors, crossbars, etc.) on facades becomes essential. It is always difficult to obtain promised results in high-level detailed featured extraction, especially for windows. Since windows have a low laser reflection rate, there are usually no laser points or few laser points representing windows in 3D point clouds, meaning that effective 3D information on windows cannot be directly obtained without human intervention. To conquer the problems induced by insufficient raw point cloud data, a series of explorations have been conducted by different experts in recent years.

2.4.1 2D Window Extraction

The window extraction is usually the subsequent procedure of the building façade extraction. Extracted building facades are always planar structures in the point cloud, meaning that most researches in window extraction are conducted in 2D plane or 2D images. Aijazi et al. (2014) proposed an automatic window detection method from TLS datasets. The extracted building façade was firstly projected into a 2D plane and a watertight boundary of this façade is generated to build a sealed environment. Then a point inversion algorithm was applied,

resulting that all holes including windows are surfaced out (Aijazi et al., 2014). The standard accuracy of this method can be over 90%; however, the extracted building façade was projected to a 2D plane first before window extraction, which will inevitably lead to some accuracy loss. Similarly, Mesolongitis & Stamos (2012) developed an algorithm under the assumption that windows were periodic structures on facades. The algorithm firstly converted 3D point cloud into a 2D binary map, then by iteratively sorting alignment and peak location analysis, windows were finalized with the result that 4,744 windows were successfully detected out of 6,614 testing samples. Nevertheless, this method can only detect window existences on building facades rather than exact window frames, and the completeness of the result is also low (Mesolongitis & Stamos, 2012).

Window edges are usually linear features on walls, which also enlightens researchers to inspect windows by detecting linear structures on building facades. Nguatem et al. (2014) developed a method to localize windows and doors in a 3D point cloud. This method depended on Monte Carlo Simulation (MC-Simulation). Templates of windows and doors were firstly generated while a parametric B-spline curve was used to interpolate windows and doors as continuous curves, and then 2D shape-space method was introduced to match similar shapes of the templates in the point cloud (Nguatem et al., 2014). This method can achieve high completeness in window detection. However, it can only be used in rectangular window detections.

2.4.2 3D Window Extraction

As mentioned before, windows are usually holes on building facades; insufficient raw points for windows on building façade enhance the difficulties in window detection. However,

this characteristic provides some experts inspirations that hole-detection algorithms can be applied in window extraction. Pu & Vosselman (2007) put forward an approach of window extraction based on hole detections in a TLS point cloud. A TIN (triangulated irregular network) was firstly produced to contain all points in extracted building facades. And then boundary points of the building façade could be successfully inspected since long TIN edges only appear at outer boundaries. Then points belonging to the same holes were clustered by boundary tracing, and holes could be detected in the TIN mesh at last. This method offers a promising approach in window recognition and it can achieve relatively high accuracy in the end.

Similarly, Wang et al. (2011) proposed an algorithm based on hole detections in an MLS point cloud. They mentioned that except for some rare cases of extraordinary architecture, edge of building structures such as windows, doors, are usually aligned along horizontal and vertical directions. They generated a pattern that divided potential window frames into four categories: upper borders, lower borders, left borders and right borders. Then a volumetric grid was manipulated to obtain more explicit neighborhood relationships among sparse points. By detecting these frames along with holes, windows can be successfully extracted (Wang et al., 2011). This approach can achieve a 100% correctness rate in experiments; however, prior semantic knowledge about sizes of windows and intervals between windows is essentially needed.

2.5 Chapter Summary

This chapter firstly reviewed related backgrounds about MLS systems, including MLS technology, components and the working principle of MLS systems. Especially, basic

information of RIEGL VMX-450 system used in this thesis is introduced. Then the chapter continued by introducing the coordinate system used in the thesis and the concept of feature extraction.

In addition, the most state-of-the-art building façade extraction algorithms were reviewed in this chapter. It can be summarized that large-sized data volume, unevenly distributed point density, occlusions, and diverse façade structures of MLS point clouds bring great difficulties to building facade segmentations. So far, there is still no cosmopolitan-recognized method to automatically extract 3D building facades from wide-range scenes with complex environments. In allusion to the problems in existed studies, a new algorithm of building façade extraction will be proposed in Chapter 3.

Furthermore, algorithms of window extraction in recent studies were also reviewed. By comparing the results of the reviewed 2D window extraction algorithms and 3D window extraction algorithms, it can be concluded that Wang's method (Wang et al., 2011) can achieve a promising result so far, except that prior knowledge of window sizes and intervals between adjacent windows should be provided first. Therefore, in Chapter 3, a revised algorithm of Wang's method will be proposed.

Chapter 3 Semi-automated Extraction of 3D Windows

This chapter provides a detailed introduction of the methodology of semi-automated LOD3 feature extraction in this study, namely, 3D window extraction from MLS point clouds. The study area and the test datasets are described first. Then information about the software and the programming platform are summarized. Furthermore, a stepwise workflow of the methodology is presented, including the ground point removal using voxel-based upward-growing algorithm, the noise removal using statistical analysis, the classification using conditional Euclidean clustering algorithm, the building façade extraction using density/width analysis, and the window extraction using hole detection algorithm. Finally, an accuracy assessment mechanism is introduced.

3.1 Workflow

To provide solid and credible information for building reconstruction in a LoD3 building models, this semi-automated 3D window extraction method endeavors to identify window frames on building facades from MLS point clouds. This algorithm is a stepwise procedure to interpret the MLS point cloud into sematic features. Figure 3-1 shows the workflow of the methodology.

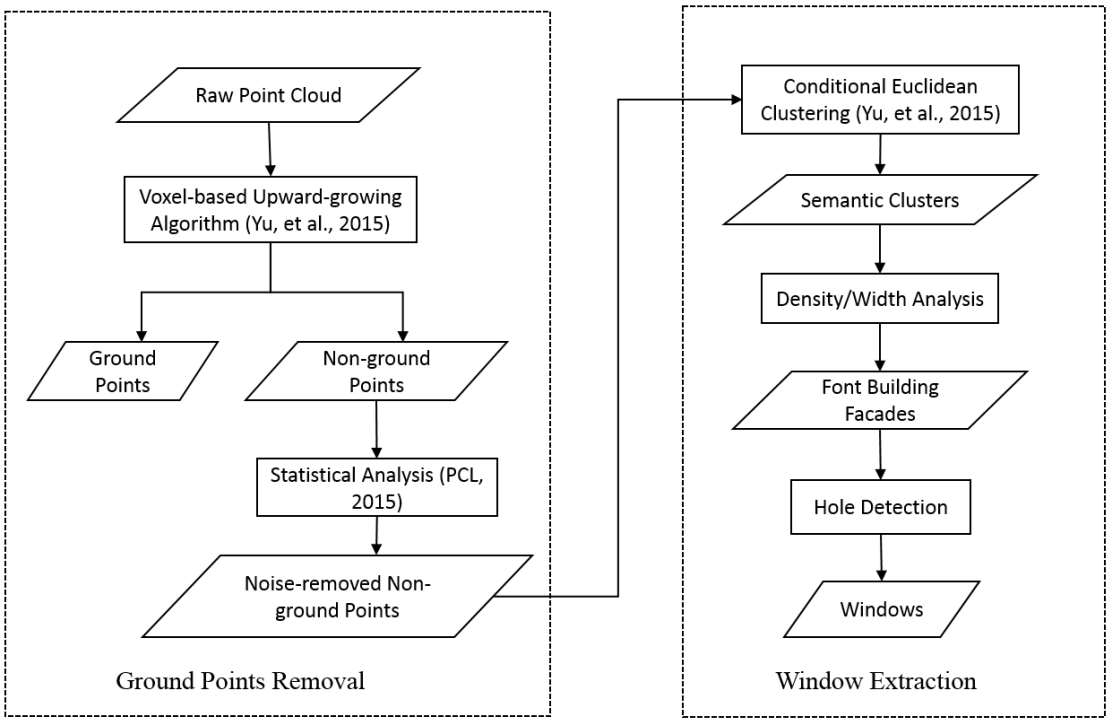


Figure 3-1 Workflow of the proposed method

The raw point cloud is firstly allocated into a 3D grid, then a voxel-based upward-growing filtering method is applied to distinguish non-ground points from ground points. Then noise is filtered out from the raw MLS point cloud by computing and analyzing the distances from each point to its neighbours. By assuming that the computed distances of each point should behave according to Gaussian distribution with a certain mean and a standard deviation, points whose mean distances to their neighbours are beyond a given interval are taken as noise. In order to segment the building facades, all the remaining non-ground points are clustered based on conditional Euclidean clustering algorithm. Then all the clustered point clouds are projected onto a pseudo 2D plane (XoY) ignoring z values (3D information of the point cloud are still marked and retained). Under the hypothesis that points of building facades are near-linear

features on the XoY plane, then the projected clusters whose density and width are over a threshold will be designated as points for building facades. After a building façade is successfully extracted, a volumetric box is created to contain façade points so that neighbours of each point can be operated. A manipulator is applied according to the structural characteristics of window frames to extract the potential window points. The window frames are divided into left, right, upper and lower crossbars. By the fact that points only existed above upper crossbars, the upper crossbars can be extracted, and the left, right and lower crossbars can be detected accordingly. Thus, the windows and outliers on building facades can be extracted.

The programming platform of Microsoft Visual Studio 2010 and third-party programming library Point Cloud Library (PCL) are utilized to realize statistical analysis, voxel-based upward-growing filtering method, conditional Euclidean clustering, density/width analysis and hole-detection algorithm. Cloud Compare v2.6.2 is used to display and analyze the tested data and experimental results, and ArcGIS 10.2.2 is employed to generate orthophotos which are captured by optical cameras along with the point clouds in the study area.

3.2 Ground Points Removal

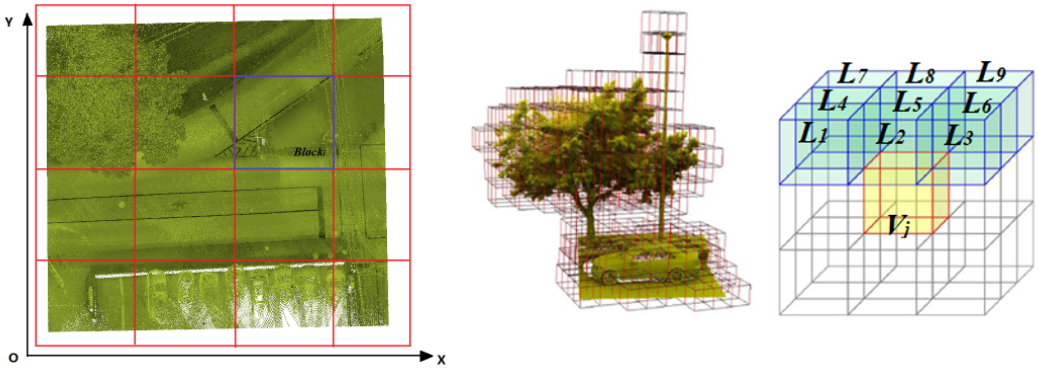
3.2.1 Voxel-based Upward Growing Algorithm

Typically, MLS systems have a more straightforward perspective to the ground points, therefore, 3D point clouds collected by MLS systems always have high-density and large-volume ground points. Nevertheless, these high-density ground points will undoubtedly increase the complexity and efficiency of non-ground object detection algorithms. Thus,

applying fast and effective pretreating method to filter out ground points in 3D scene point clouds is of great significance in enhancing reducing data search range and improving the performance of 3D non-ground object detection algorithms. In this section, a voxel-based upward-growing algorithm (Yu et al., 2015) is described to filter out ground points from the raw MLS point cloud.

Grounds are always undulating in the 3D point cloud scenario. If the entire tested 3D point cloud dataset was processed simultaneously, the running time of voxel-based upward-growing method will increase, additionally, the performance of the proposed algorithm is not promising. Therefore, in order to quickly and effectively process the undulating ground points, the entire point cloud will be segmented first, and ground-point filters will be applied in each block step by step. By using this strategy, the ground undulation in each block will be alleviated, and the filtering result will be more promising.

As shown in Figure 3-2 (a), the entire 3D point cloud will be perpendicularly and parallelly segmented into a series of local point blocks ($Block_i$, $i=1, 2, 3, ..., N_b$) according to a certain width (w_b, which is determined by the size of tested datasets) in the XY plane, N_b is the total number of the generated local point blocks, and these local point blocks will be individually processed to filter out ground points. Secondly, as shown in Figure 3-2 (b), each $Block_i$ ($i=1, 2, 3, ..., N_b$) will be divided into a series of spatially continuous point cloud voxels (v_j, $j=1, 2, ..., N_v$) based on a certain width (w_v) by using Octree Spatial Index, where w_v is determined by w_b and average point density, and N_v is the total number of generated voxels. Additionally, as shown in Figure 3-2 (c), each voxel has 26 adjacent voxels. Based on the unique octree index structure, a voxel-based upward-growing method is proposed and described as follows:



(a) (b) (c)

Figure 3-2 Principle of voxel-based upward-growing algorithm

(a) Segmented raw point cloud; (b) Octree Spatial Index in a local block (Yu et al., 2015);

(c) Voxel-based upward-growing pattern

Figure 3-2 (c) displays the detailed procedure of voxel-based upward-growing method. For each voxel v_j ($j=1, 2, \ldots, N_v$), the upward-growing process firstly grows along with its '9 neighbours', where '9 neighbours' of a certain voxel is defined as voxels adjacent to the voxel v_j and locating above the voxel v_j. For instance, the '9 neighbours of v_j in Figure 3-2 (c) are *voxel* L_1, *voxel* L_2... *voxel* L_9. Next, the upward-growing process regards the '9 neighbours' (L_1, L_2... L_9) of voxel v_j as the starting points, and continues to grow up at the same pattern. And by this analogy, the rest of the upward-growing processes regard the generated '9 neighbours' voxels as starting points to grow up. This recursive upward-growing process will stop only when all grown voxels have no '9 neighbours' to grow up. Then, voxel v_h with the maximum local height value, which is also the highest point in the growing region, can be calculated. The 'local height value' of a certain voxel is defined as the height difference between the certain voxel and the lowest voxel in the local block which contains the certain

voxel. And the 'global height value' of a certain voxel is defined as the height difference between the certain voxel and the lowest voxel in the entire 3D point cloud. Finally, voxel v_j ($j=1, 2, ..., N_v$) will be designated as ground voxel (a voxel which contains ground points) or non-ground voxel (a voxel which contains non-ground points) according to the following criterions. By filtering out voxels which are marked as ground voxels, ground points can be removed from the raw point cloud dataset. The criterion can be described as follows:

(1) Define a local ground undulation threshold h_g (which is determined by the maximum z value of tested datasets), which constrains the largest local ground undulation in each local block of a certain 3D point cloud scene.

(2) Define a global ground undulation threshold h_e (which is also determined by the maximum z value of tested datasets), which restrains the largest ground undulation in the entire point 3D point cloud scene.

(3) If the global height value of a voxel v_j ($j=1, 2, ..., N_v$) is smaller than h_e, and the local height value of the voxel v_h is smaller than h_g, the voxel v_j will be designated as a ground voxel and removed from the raw point cloud.

(4) If the global height value of a voxel v_j ($j=1, 2, ..., N_v$) is higher than h_e, or and the local height value of the voxel v_h is higher than h_g, then the voxel v_j will be designated as a non-ground voxel and all points in voxel v_j will remain.

The voxel-based upward-growing method has the following advantages: (1) it firstly segments the global point cloud into series of local point blocks and filters out the ground points in each block, which highly enhances the efficiency and reduces the running time to

process large-scene undulating 3D point cloud datasets. (2) the growing pattern of voxel-based upward-growing algorithm can effectively retain the integrity of the non-surface data. Figure 3-3(a) presents a raw 3D point cloud which contains abundant ground points and Figure 3-3(b) shows the non-ground points after ground points are filtered out by the voxel-based upward-growing algorithm.

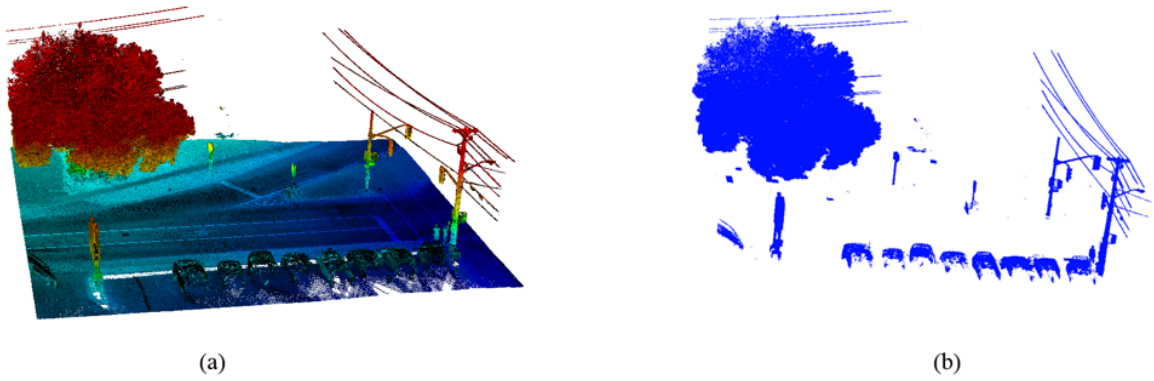


Figure 3-3 Demo of voxel-based upward-growing method

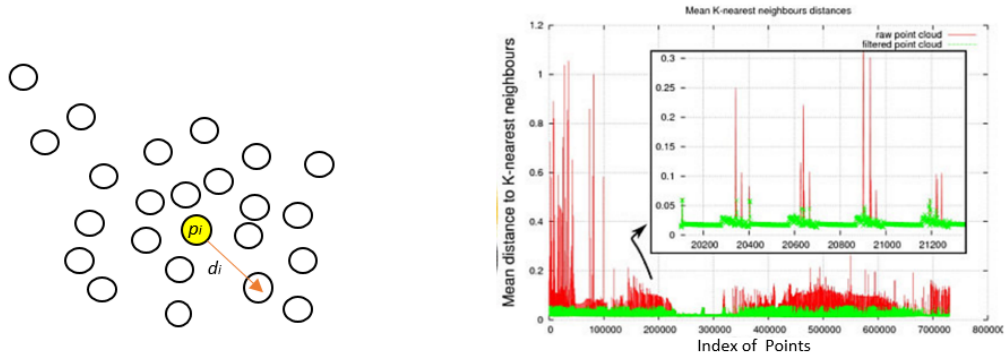
(a) A raw 3D point cloud which contains abundant ground points; (b) the filtered non-ground points

3.2.2 Noise Removal Using Statistical Analysis

Noise inevitably exists in the MLS point cloud even after ground points have been removed, which are primarily presented as isolated points, outliers and point mutations in some local areas. In order to remove noise out of non-ground features, a statistical analysis filter in PCL package (PCL, 2015) is applied here to differentiate noise from non-ground feature. The statistical analysis filter is generated by PCL in its point cloud processing library. PCL is a large-scale cross-platform open source C++ programming library, which is built up by

absorbing previous valuable algorithms in related point cloud studies. It implements ample relevant common algorithms and efficient data structure of point clouds, which refers to point cloud acquisition, filtering, segmentation, registration, retrieval, feature extraction, identification, tracking, surface reconstruction, and visualization. The window extraction algorithm in this thesis is proposed to be applied in practical commercial applications. The PCL package is an easy-to-access commercial library which can be directly taken advantage of here.

Noise removal algorithm based on statistical analysis has been proved in a series of research studies (Zhang, 1992; Rusu et al., 2008; Rusu & Cousins, 2011; PCL, 2015). It firstly implements by finding k (determined by average point density) nearest points from a certain point and computing distances from a point to its neighbours. For each point p_i ($i=1, 2, ..., N_p$) the mean distance of it from all its neighbours d_i ($i=1, 2, ..., N_p$) have then been calculated, where N_p is the total number of points in the tested non-ground points. Under the assumption that the distribution of mean distances of all the points should be Gaussian distribution with a certain mean value and a standard deviation, then points outside a thresholding interval will be considered as noise and removed from the non-ground points. The thresholding interval is determined by the computed mean value and standard deviation in Gaussian distribution. Figure 3-4 shows the principle of statistical analysis, Figure 3-4(a) indicates neighbour points of a given point p_i and distances from p_i to its neighbour points, and Figure 3- 4(b) presents the mean k-nearest neighbour distance.

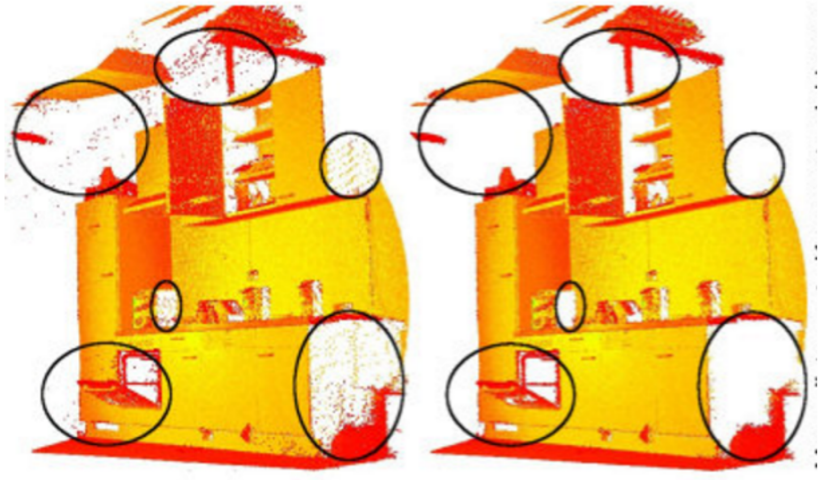


(a) (b)

Figure 3-4 Principle of noise removal using statistical analysis

(a) Neighbourhood of a certain point; (b) Mean K-nearest neighbours distances (PCL, 2015)

In this methodology, the procedure of noise removal is situated after ground point removal since high-density. Abundant ground points will enhance time and space complexity of the algorithm. Therefore, filtering out ground points at first will improve the efficiency of the entire processing algorithm. Figure 3-5 (a) gives an example of 3D point cloud with sparse noise. Figure 3-5 (b) shows the noise-removed point cloud by noise removal algorithm using statistical analysis, which indicates that noise can be successfully removed by noise removal algorithm using statistical analysis.



(a) (b)

Figure 3-5 Demo of noise removal using statistical analysis

(a) Raw point cloud with sparse noise; (b) Resulted point cloud after applying noise removal algorithm using statistical analysis (PCL, 2015)

3.3 3D Window Extraction

3.3.1 Conditional Euclidean Clustering

There are no topological relationships between point to point in a discrete and sparse 3D non-ground point cloud. Therefore, points belonging to the same semantic object in the scene are still isolated and unorganized even after filtering ground points and noises. In order to distinguish specific 3D objects from non-ground points, valid and sensitive clustering and segmentation method should be applied, to organize the sparse non-ground points into topological and semantic objects. This section utilizes conditional Euclidean clustering method to conduct fast segmentation of the 3D unorganized non-ground points. Conditional Euclidean clustering algorithm is a method to find the nearest neighbours which is inspired and proved by Yu et al. (2015). This method segments discrete points into series of clusters according to

a certain distance by computing Euclidean distances between a certain point and its adjacent ones. In this methodology, a Euclidean distance threshold (d_c) should be given first according to density and resolution of a tested dataset. If the Euclidean distance between two adjacent points is less than or equal to d_c, these two points will be designated into the same cluster. Otherwise, these two points will be assigned into different clusters. The detailed conditional Euclidean Clustering method will be presented below:

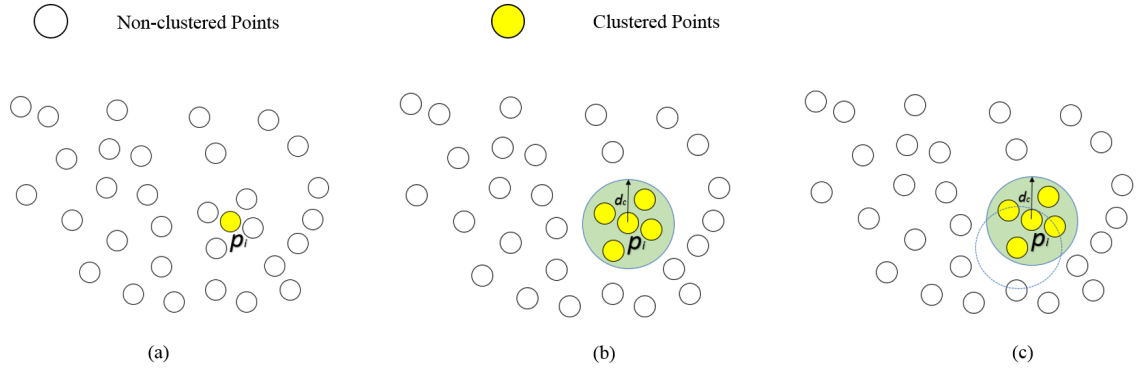
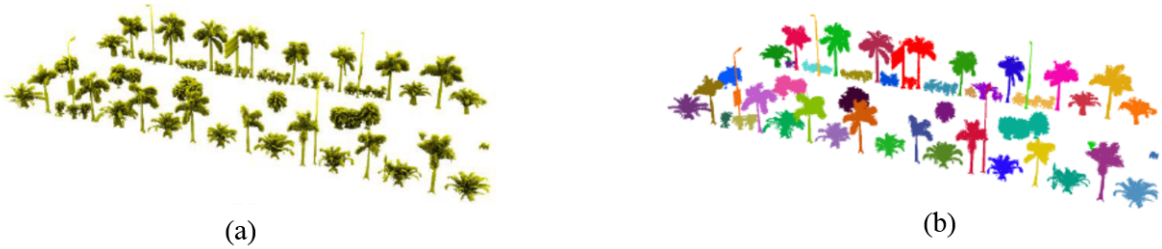


Figure 3-6 Principle of conditional Euclidean clustering method

Initially, all non-ground points are marked as non-clustered points. As shown in Figure 3-6 (a), the conditional Euclidean clustering algorithm starts at any one of the non-clustered points p_i, $i=1, 2, \ldots, N_{off}$, in which N_{off} is the total number of non-ground points. p_i is firstly marked as a clustered point, and assigned with a class identifier *cLabel*. Next, as shown in Figure 3-6 (b), p_i is designated as the centre of a sphere, and d_c is the radius of the sphere, all the non-clustered points which are within the spherical neighbourhood will be marked as clustered points, and the same class identifier *cLabel* will also be assigned to these new clustered points. Thirdly, as shown in Figure 3-6 (c), these new clustered points are respectively regarded as new starts of the conditional Euclidean clustering algorithm, and the same clustering pattern

will be continued outwards radially. By this analogy, the conditional Euclidean clustering method consistently marks non-clustered points which are within the spherical neighbourhood of a newly clustered point as clustered points, and assigns them the same class identifier *cLabel*. This recursive clustering process ends when there are no more non-clustered points within the spherical neighbourhood of these clustered points. Then, one of the rest of the non-clustered points will be selected as a new start, and the same conditional Euclidean clustering pattern will continue until all the non-ground points are clustered. When all non-ground points are clustered and assigned with different class identifiers, points with the same class identifier will represent for the same cluster. Figure 3-7 shows the clustering results by using conditional Euclidean clustering method. Figure 3-7 (a) presents noise-removed non-ground points, and Figure 3-7 (b) indicates that an independent semantic object can be effectively separated by using conditional Euclidean clustering algorithm.



(a) (b)

Figure 3-7 Demo of conditional Euclidean clustering algorithm (Yu et al., 2015)

As shown in Figure 3-7, a spatial independent semantic object can be effectively separated after conditional Euclidean clustering algorithm has been applied. In addition, the time complexity of conditional Euclidean algorithm is low so that segmentations of non-ground independent semantic objects can be completed at a high efficiency. Since the final targets in

this methodology are 3D windows on building facades, prior knowledge (size, height, etc.) of the targeted objects can also be utilized for in-depth filtering. It can further improve the processing efficiency of the 3D clustering algorithm, so that small sizes of useless clusters (cars, transmission lines, pole-like objects) can be removed. A height difference threshold h_c and a width threshold w_c are defined preliminarily, then a cluster whose height difference h_r is smaller than the given h_c or whose width w_r is smaller than the given w_c will be removed. "Height difference" represents for the absolute subtraction result between the maximum z value and minimum z value of a specific cluster. The width of a cluster is the absolute subtraction result between maximum x value and minimum y value, or the subtraction result between maximum y value and minimum y value of a specific cluster. The targeted clusters can be refined by the following equation:

$$\begin{cases} True\ cluster, & if\ \ h_r > h_c\ \&\&\ w_r > w_c \\ False\ cluster, & if\ \ h_r < h_c\ \ or\ w_r < w_c \end{cases} \quad \text{3-1}$$

3.3.2 Building Façade Extraction by Density/Width Analysis

MLS point clouds can directly record rigorous 3D coordinates from real scenes, which facilitates immediate 3D geometric feature extraction. Large field of view and powerful ability of acquiring high-resolution data make MLS point clouds can be more conducive not only in large-scale building structure detections, but also in detailed feature extraction. However, as mentioned in Chapter 2, there is still no sophisticated algorithms that can quickly and effectively classify 3D MLS point clouds measured from large-scale complex environments into individual objects (ground, building, wire, rod-shaped object, tree, etc.). Therefore, in this section, a density/width analysis method is put forward based on the density and geometric

properties of building facades in point clouds. The building façade extraction method by density/width analysis proposed in this section is inspired by the culling mechanism presented in Melzer & Briese (2004). In the culling mechanism, Melzer & Briese (2014) pointed out that features in 3D point cloud had different density properties. The algorithm put forward in this section is based on his assertion.

The X-Y plane is firstly subdivided into a 2D grid by g m× g m, in which g is pre-defined by average point density of input clusters generated by conditional Euclidean clustering algorithm. The input clusters are then projected into this gridded X-Y plane, and the z value of each point is remaining as a label. The total number of this cluster will be recorded as N_c. For each cluster, the length *Wid* and the average density *Den* of the projected cluster can be calculated by:

$$W_1 = x_{max} - x_{min} \tag{3-2}$$

$$W_2 = y_{max} - y_{min} \tag{3-3}$$

$$Wid = \begin{cases} W_1, & W_1 > W_2 \\ W_2, & W_1 < W_2 \end{cases} \tag{3-4}$$

$$Area = \begin{cases} W_1 \times W_2, & W_1 \neq 0 \; and \; W_2 \neq 0 \\ W_1, & W_2 = 0 \\ W_2, & W_1 = 0 \end{cases} \tag{3-5}$$

$$Den = N_c / Area \tag{3-6}$$

where x_{max} is the maximum x value in the input cluster, y_{max} is the maximum y value in the input cluster, x_{min} is the minimum x value in the input cluster and y_{min} is the minimum y value of the input cluster, *Area* is the pseudo rectangular acreage of the projected area. Especially, when the $W_1 = 0$ and $W_2 = 0$, the input cluster will be directly regarded as non-façade points.

Under the prior knowledge that clusters belonging to building facades have a relatively high average density and width at the same time, clusters whose average density and width are inside a given thresholding interval will be regarded as building facades and remained to be processed in the next step. The given thresholding interval is determined by the density of raw 3D point clouds.

3.3.3 Window Extraction Using Hole Detection Algorithm

Automated extraction of windows from point clouds relies on the extraction of the window edge on the building facades. Wang et al. (2011) proposed a hole-detection algorithm for window extraction. In their algorithm, prior knowledge of window sizes and distances between windows are needed. A pattern that classifies window frames into four categories is firstly conducted: horizontal borders on the top and bottom of windows, and vertical borders on the left and right side of windows always leave holes on building facades (Wang et al., 2011). In order to simulate neighbourhood relationships among points, a volumetric manipulator at a grid size vg is created to contain all the points of the extracted building façade, where vg is determined by the density of the input point cloud. Then an operator is conducted according to the window pattern to localize windows excluding window crossbars. For each voxel (*i, j, k*), I bespeak *f (i, j, k)* =0 if there is no laser point in this voxel, and *f (i, j, k)* =1 if there are laser points in this voxel, the equation of the operator lists below:

$$\begin{cases} horizontal\ window\ edge\ points, & if\ \{\sum_{k'=k}^{k'=k+1}\sum_{i}\sum_{j} f(i,j,k')\}=1\ \&\&\{\sum_{k'=k}^{k'=k-1}\sum_{i}\sum_{j} f(i,j,k')\}=0\ \&\&\{\sum_{i}\sum_{j} f(i,j,k)\}=1 \\ & OR\ if\ \{\sum_{k'=k}^{k'=k-1}\sum_{i}\sum_{j} f(i,j,k')\}=1\ \&\&\{\sum_{k'=k}^{k'=k+1}\sum_{i}\sum_{j} f(i,j,k')\}=0\ \&\&\{\sum_{i}\sum_{j} f(i,j,k)\}=1 \\ vertical\ window\ edge\ points, & if\ \{\sum_{i'=i}^{i'=i+1}\sum_{j}\sum_{k} f(i',j,k)\}=1\ \&\&\{\sum_{i'=i}^{i'=i-1}\sum_{j}\sum_{k} f(i',j,k)\}=0\ \&\&\{\sum_{i}\sum_{j} f(i,j,k)\}=1 \\ & OR\ if\ \{\sum_{i'=i}^{i'=i-1}\sum_{j}\sum_{k} f(i',j,k)\}=1\ \&\&\{\sum_{i'=i}^{i'=i+1}\sum_{j}\sum_{k} f(i',j,k)\}=0\ \&\&\{\sum_{i}\sum_{j} f(i,j,k)\}=1 \\ non-window\ edge\ points, & otherwise \end{cases} \quad 3\text{-}7$$

As shown in Eq. 3-7, the upper horizontal window border is recognized if the upper neighbour exists while the lower neighbour does not; the same rationale is manipulated to the lower border, left border and right border of windows. When window frames are successfully localized by this mechanism, points belonging to these four window borders will be designated into a classifier and marked as window points.

3.4 Validation

3.4.1 Window Regions in 2D

Digital images were taken as the references for validations. As mentioned in Chapter 2, digital images and 3D point clouds are collected together while the vehicle is moving so that the two kinds of sensors (optical camera and laser scanner) share the same environmental variables. Digital images have distortions, however, 3D points of window frames contain absolute coordinates and geometric information. Therefore, optical distortions in images should be eliminated first so that digital images can be a reference for the validation. In this

study, orthophotos of building façades should be generated to validate the extracted 3D windows before the accuracy assessment.

(1) Orthoimagery Generation

Figure 3-8 shows the geo-referenced frame of 3D point clouds. Detailed 2D information of the building façades is only included when the 3D point cloud is projected to a vertical plane (XoZ), where axes of X, Y, and Z are corresponding to axes in the UTM projection (UTM zone: 18N). Therefore, the extracted building windows are assessed their 2D projections on the vertical plane (XoZ), namely window outlines. The extracted window outlines are compared with the windows' polygons generated by digitizing building window outline from the vertical orthoimagery of building facades.

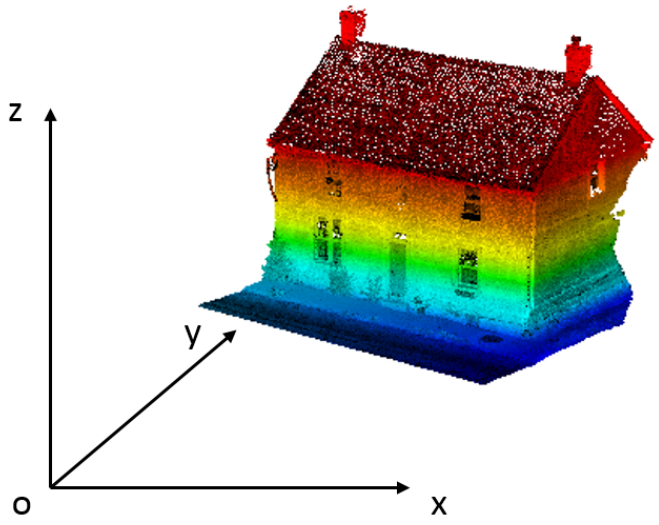


Figure 3-8 Geo-referenced frame in 3D point cloud

In order to generate orthophotos of the projected 3D point cloud in XoZ plane, several control points (C_1, C_2, … C_n, n=1, 2, 3…) in raw 3D points cloud are firstly selected to draw a

base map. Only 2D information (x, z) of the selected control points is used to generate the base map. As shown in Figure 3-9, the control points are gridded evenly according to their coordinates in the base map. Image space coordinate system is used to describe the location of image points in 2D plane in this section. The locations of image points are described by the numbers of rows and columns in pixel arrays in this paper. Specifically, coordinate *x* of a pixel is the row number in the pixel array, and coordinate *y* of a pixel is the column number in the pixel array. Generating orthophotos is indeed geo-referencing raster datasets. Orthophoto generation can be operated by the module Geo-referencing in ArcGIS 10.2.2. The referenced digital image for each dataset is input as a raster layer into ArcGIS first. The process identifies the spatial information of the raster dataset by linking locations on the raster layer with the corresponding locations (with known x,y coordinates) on the base map. Then the pixels in the raster layer will be relocated according to the control points on the base map.

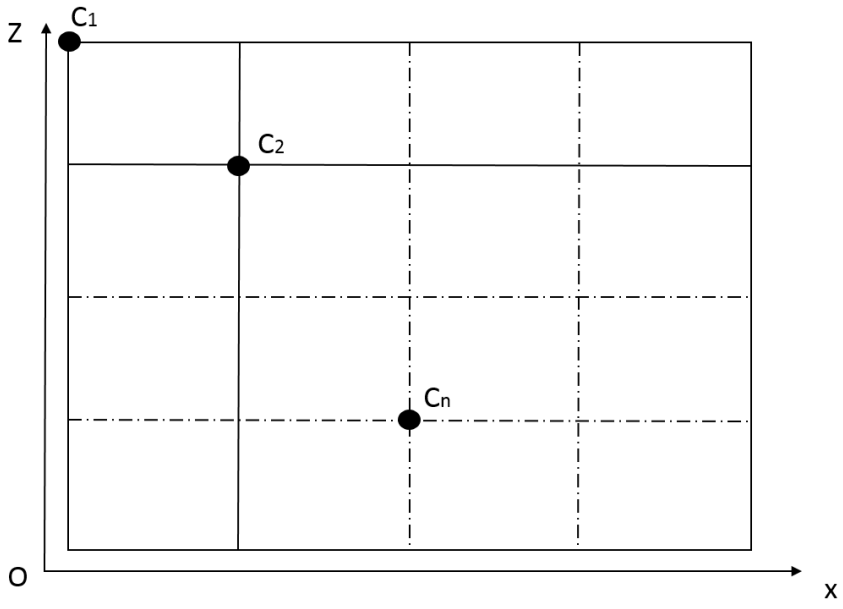


Figure 3-9 Base map of orthophoto generation

According to ArcGIS Resources (ArcGIS, 2016), a least-square fitting algorithm is applied here to optimize the global accuracy of the generated orthophotos. There are several modules in transforming a raster dataset in ArcGIS. One control point is needed for zero-order shift, 3 for affine transformation (1-order polynomial), 6 for second order polynomial transformation and 10 for third order polynomial transformation. The more control points, the higher the transformation order, and the more the distortion will be corrected. Nonetheless, this research needs to achieve optimal accuracy at least cost of control points. After repeated tests, it was found that second order polynomial transformation performed adequately on the digital images those needed to be transformed. Then by linking each control point in base map and its corresponding image point6 in raster layer, the distortion in digital images can be corrected, and final orthophotos of building facades on the vertical plane (XoZ) can be generated.

When orthophotos are successfully generated, the extracted windows will be overlapped on the orthophotos in Cloud Compare v2.6.2 to generate the overlapped photos. The resolutions of the referenced orthophoto and the overlapped photo for each dataset are adjusted to be the same to avoid the influences caused by different resolutions.

(1) Quantitative Assessment

The accuracy assessment mechanism of window regions in 2D is conducted based on completeness (*cpt*), correctness (*crt*), F-*measure* (*F*). As shown in Equation 3-8 to 3-10, the *cpt* represents the integrity of the extracted windows; the *crt* shows how many valid and correct windows are extracted by using the proposed methodology; and *F-measure* is a global score by integrating correctness and completeness. Where C_p represents f the number of valid pixels belonging to the exact windows in the extracted points, R_p is the number of the window pixels

interpreted manually from the generated orthophotos, and R_r is the number of pixels of the extracted points by the proposed method.

$$crt = C_p/R_r \quad 3\text{-}8$$

$$cpt = C_p/R_p \quad 3\text{-}9$$

$$\mathrm{F} = 2\times\frac{cpt\times crt}{cpt + crt} \quad 3\text{-}10$$

3.4.2 Window Regions in 3D

The accuracy assessment of window regions in 3D is based on correctness (*crt*). Window points are manually extracted from the raw point clouds in this section. For each 3D point extracted by the proposed method, a corresponding point extracted by the manual interpretation should be found. The *crt* shows the portion of correct window points extracted by the proposed method and it is defined as C_n/R_n, where C_n is the number of valid 3D window points extracted by the proposed method those can be found in manually interpreted points, and R_n is the total number of extracted 3D window points by the proposed method.

3.5 Chapter Summary

This chapter detailed the methodology. The rationales of the voxel-based upward-growing algorithm, statistical analysis, conditional Euclidean clustering algorithm, density/width analysis, and hole detection algorithm are presented in details. Quantitative assessment method has also been described. Some experimental datasets will be employed to test the feasibility, the efficiency and the accuracy of the proposed methodology in Chapter 4.

Chapter 4 Results and Validation

This chapter presents the results of window extraction and validation. Firstly, results of the test datasets after applying the stepwise algorithms are presented and discussed. Then the accuracy assessments in both 2D and 3D are demonstrated.

4.1 Design of the Environment

A study area is chosen to test the proposed approach for semi-automated extraction of building windows on different building types. The point clouds covering the building facades were acquired by a commercial MLS system. Accuracy of the extracted building windows is assessed using the reference data.

4.1.1 Study Area

The study area is located in the City of Kingston, Ontario, Canada (longitude 76°33'22.57"W, latitude 44°13'7.40"N). Kingston is a city located in Eastern Ontario where the St. Lawrence River flows out of Lake Ontario. It is midway between Toronto and Montreal (see Figure 4-1). Figure 4-2 shows the sections of King Street West and Front Road nearby the lakeshore of Lake Ontario, which were surveyed by a RIEGL VMX-450 system The point cloud datasets of a 3.7 km long road section of King Street West were used to evaluate the developed algorithms. On this road section, there exist different dwelling types (such as single-detached, semi-detached houses), commercial and high-rise buildings.

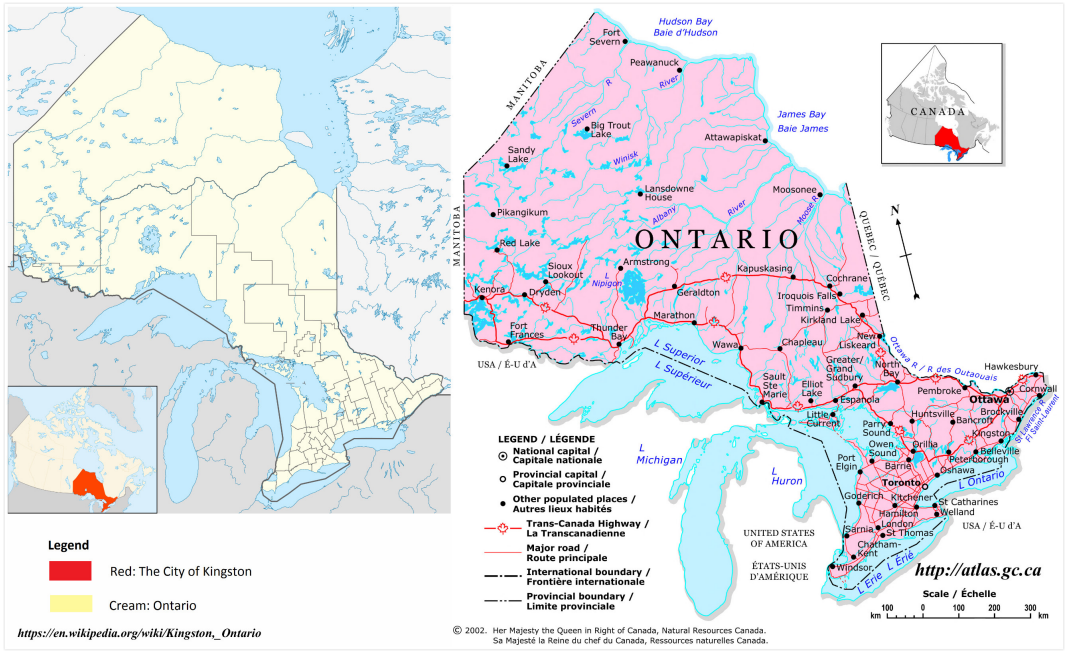


Figure 4-1 Location of the study area

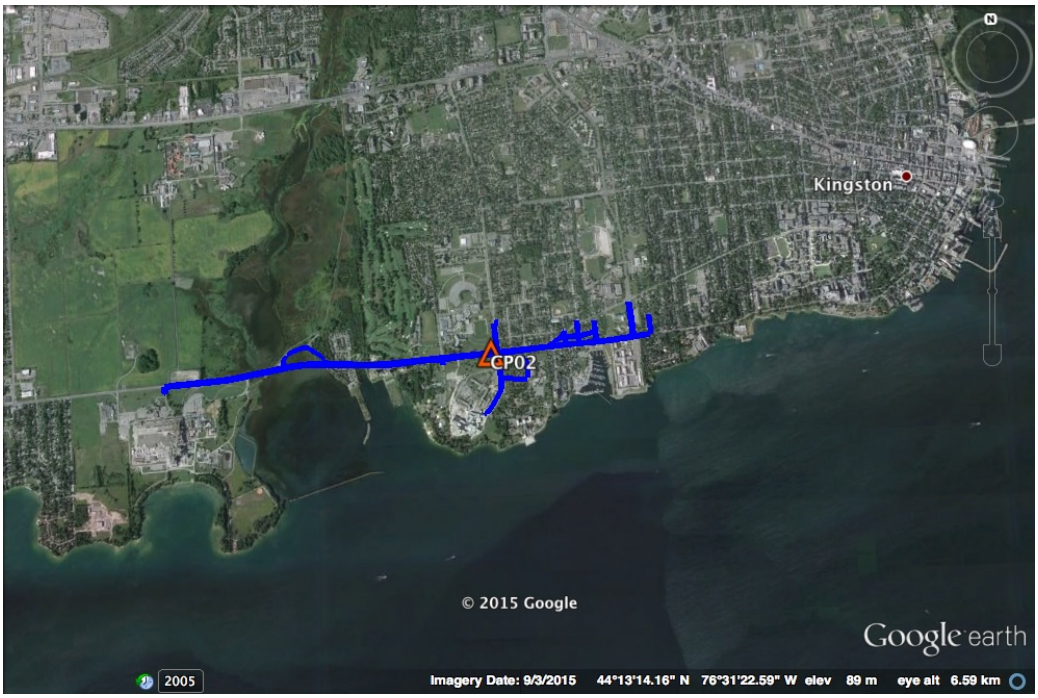


Figure 4-2 Trajectory of the study area (Google, 2015)

4.1.2 Point Cloud Data

The point cloud datasets were acquired on August 29th, 2013 by a RIEGL VMX-450 system from TULLOCH Engineering (as shown in Figure 4-3). The total of over 916 million points were stored in 45 LAS files, which use 23.8 GB generated by TerraScan Software. The 10 GB images taken by the CCD cameras were saved as JPEG format, including a CSV (Comma Separated Value) file containing the origin and direction coordinates, timestamp, roll, pitch, and yaw information.

Figure 4-3 The RIEGL VMX-450 system of TULLOCH Engineering

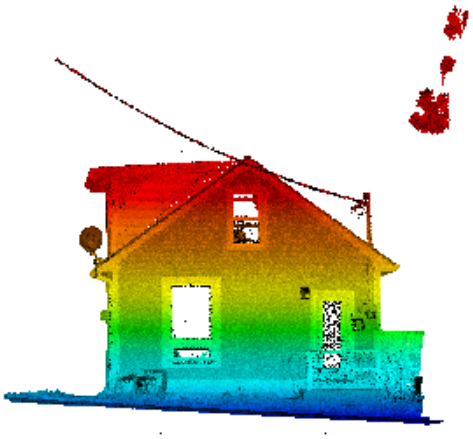
Figure 4-4 shows the five datasets that were selected from the data acquired by the RIEGL VMX-450 system. Figures 4-4(a), (c) and (e) show Dataset 1, a single-detached house with its digital image, front view and oblique view of an MLS point cloud with 787,235 points in about 86.79m^2. Figures 4-4(b), (d) and (f) show Dataset 2, a single-detached house covered by 1,442,607 points in approximately 185.31m^2. Figures 4-4(g), (i) and (k) show Dataset 3, a single-detached house covered by 1,281,313 points in about 212.94m^2. Figures 4-4(h), (i) and (l) is the fourth MLS point cloud with 725,579 points in approximately 324.86m^2, and Figures (m), (n) and (o) is the fifth MLS point cloud with 3,336,229 points in approximately

1,577.45m^2. Datasets 2 and 4 are selected since they include typical window types (rectangular, irregular and arc-rounded windows) in the study area. Dataset 3 was used to test whether the proposed method will be influenced by occlusions of trees. Dataset 5 was used to validate the flexibility of the proposed method in complex scenes. In addition, holes usually exist on building facades in 3D point cloud due to systematic errors in MLS systems. As such, Dataset 1 was used to test the influences that big holes will have on the hole detection algorithm.

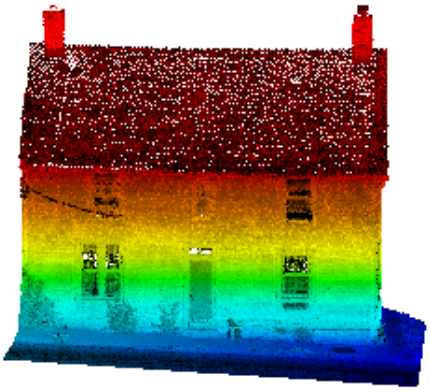
Figure 4-5 shows Gaussian density distributions of the five test datasets. Point density is influenced by elements such as distance from laser scanners to the scanned point, scanning angle, frequency of transmitted scanning beams, etc. The average densities of test datasets are different. The density analysis was completed using volume density analysis module in Cloud Compare v2.6.2. The entire point cloud was gridded into 1 m×1 m×1 m volumes, and points in these volumes are calculated. Different colours in the figures represent for areas within different densities. Colours demonstrated in Figures 4-5 (a), (c), (e), (g) and (i) and charts (Figures 4-5(b), (d), (f), (h) and (j)) are corresponding to each other. Generally speaking, features close to the road centre lines have higher densities than those of further away, and ground objects have higher density than non-ground objects. Additionally, as shown in Figures 4-5(b), (d), (f), (h) and (j), the average densities of Datasets 1 to 5 are 5,628.5, 6980.1, 4653.3, 1569.7, and 4771.2 pts/m^3, respectively.

(a) Digital image of Dataset 1

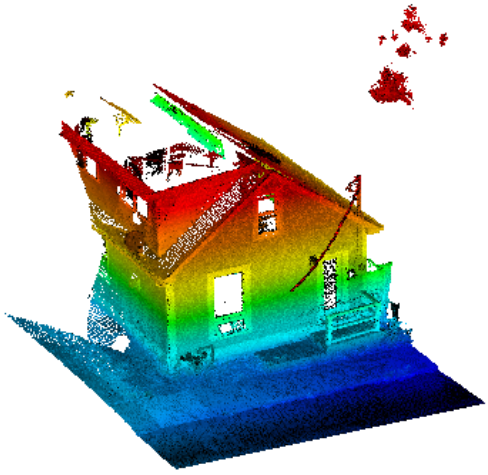
(b) Digital image of Dataset 2



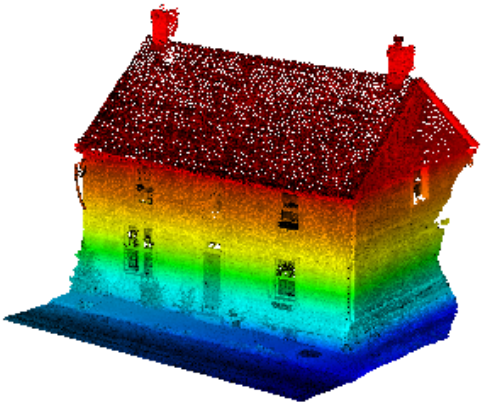
(c) Front view of Dataset 1



(d) Front view of Dataset 2

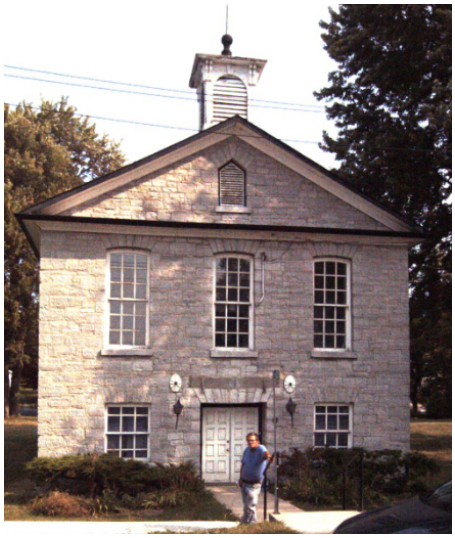


(e) Oblique view of Dataset 1

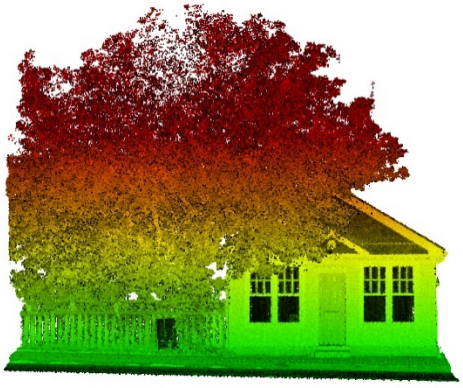


(f) Oblique view of Dataset 2

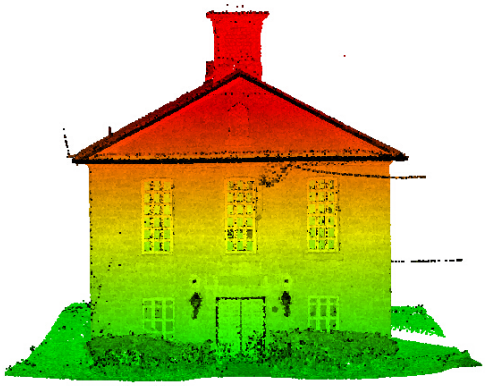
(g) Digital image of Dataset 3



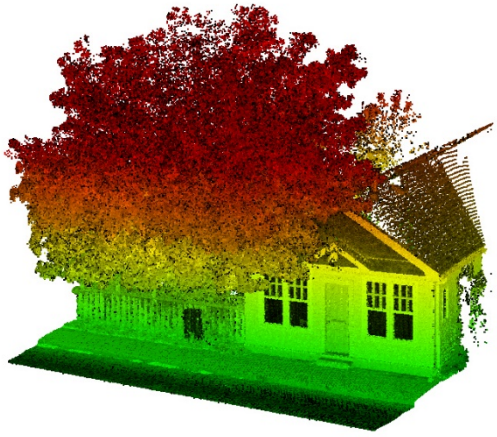
(h) Digital image of Dataset 4



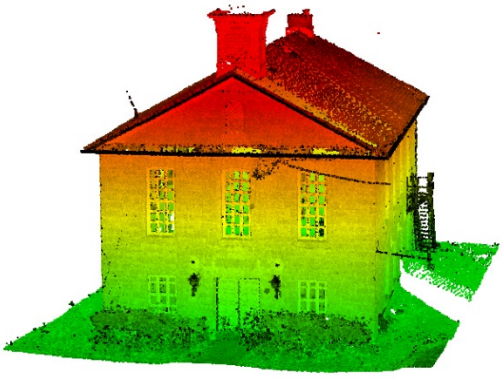
(i) Front view of Dataset 3



(j) Front view of Dataset 4

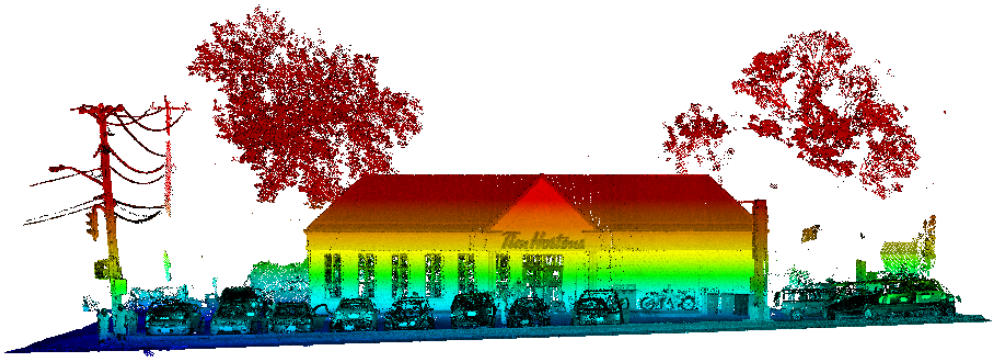


(k) Oblique view of Dataset 3

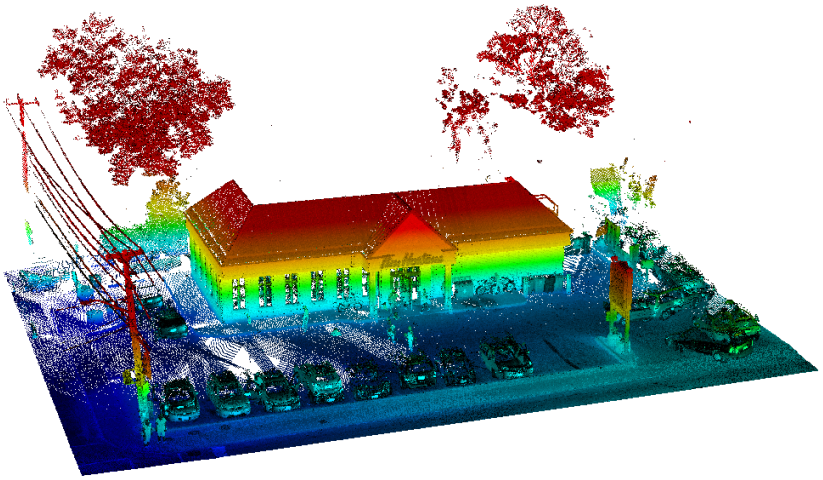


(l) Oblique view of Dataset 4

(m) Digital image of Dataset 5

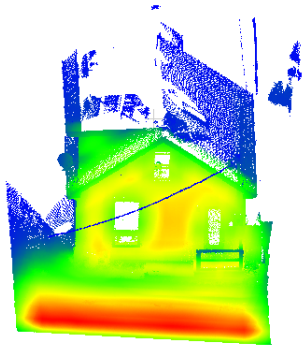


(n) Front view of Dataset 5

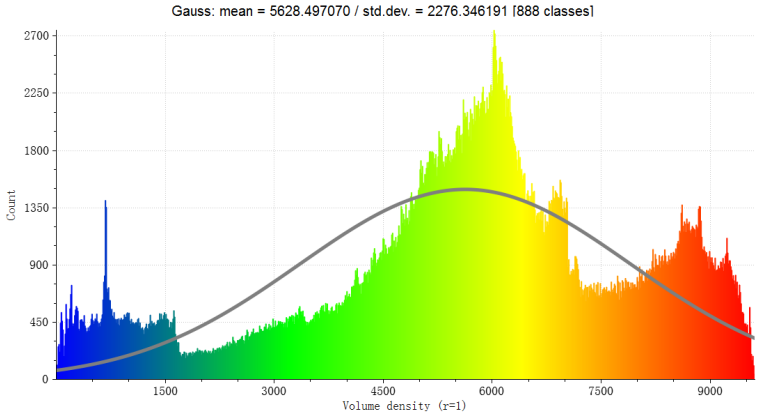


(o) Oblique view of Dataset 5

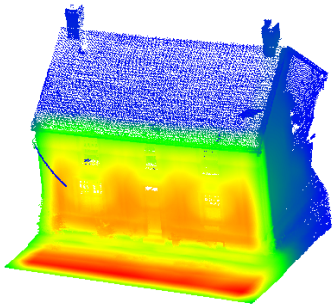
Figure 4-4 Five datasets used in this study



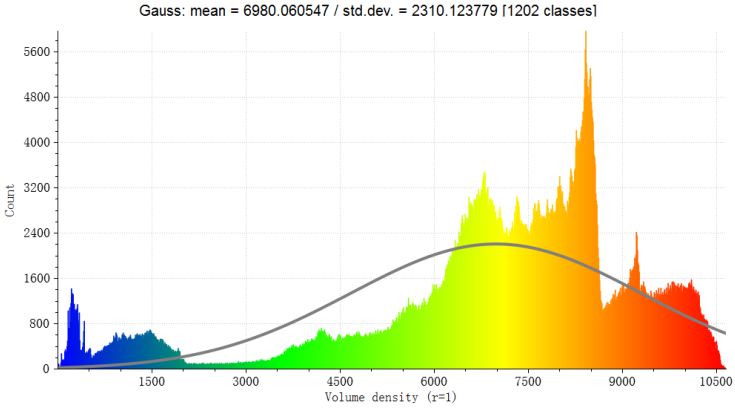
(a)


Gauss: mean = 5628.497070 / std.dev. = 2276.346191 [888 classes]
Count
Volume density (r=1)

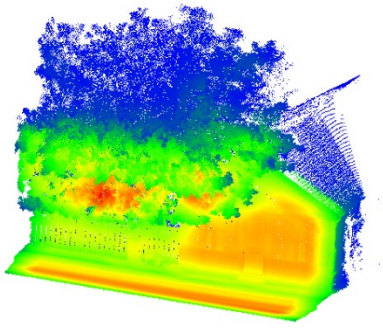
(b)



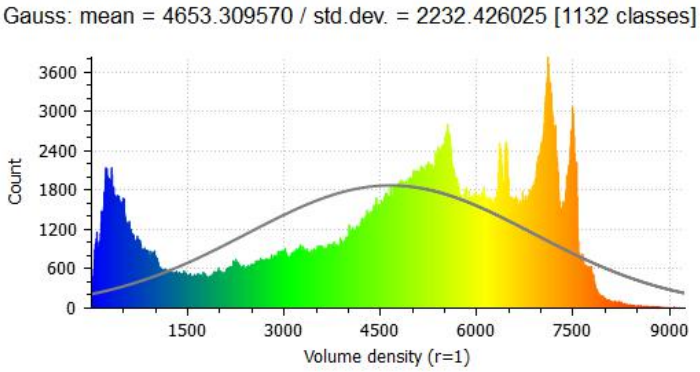
(c)


Gauss: mean = 6980.060547 / std.dev. = 2310.123779 [1202 classes]
Count
Volume density (r=1)

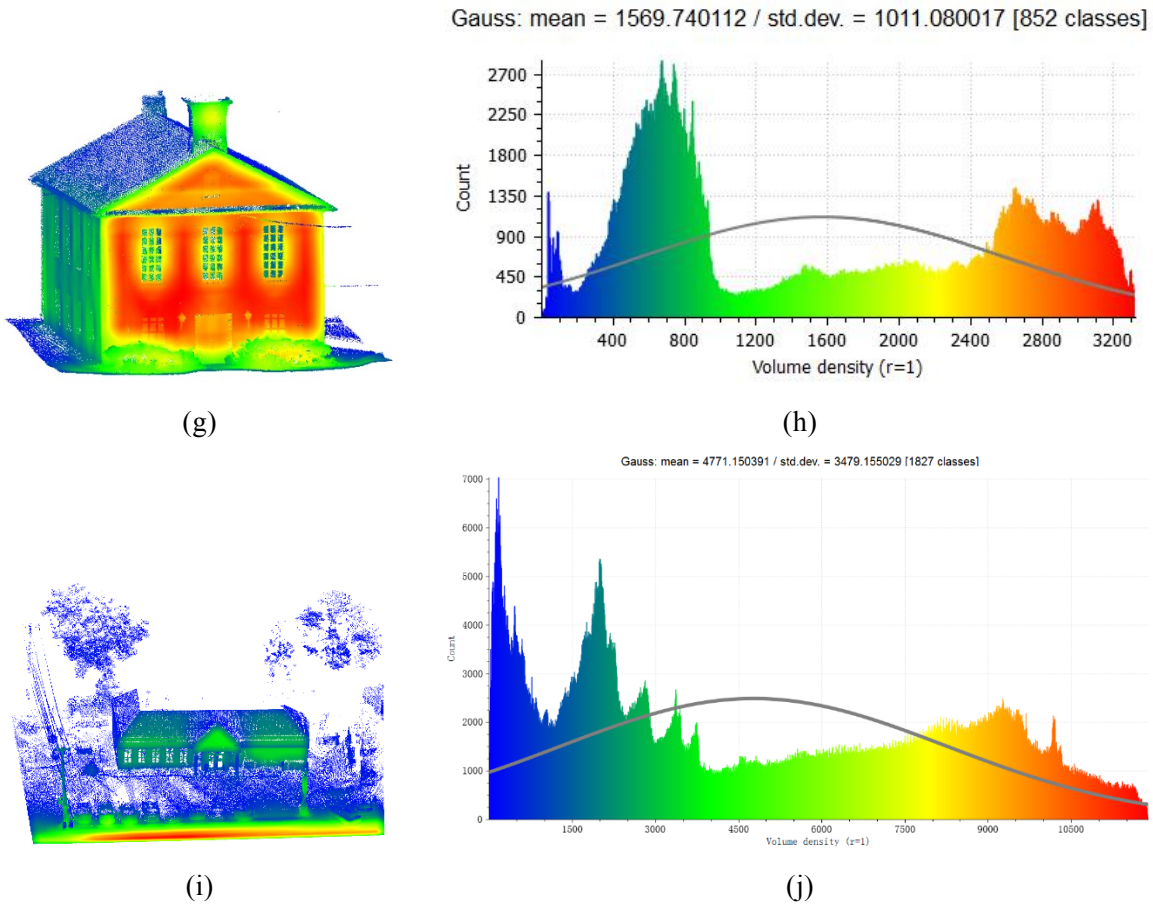
(d)



(e)


Gauss: mean = 4653.309570 / std.dev. = 2232.426025 [1132 classes]
Count
Volume density (r=1)

(f)



(g) (h)

(i) (j)

Figure 4-5 Gaussian density distribution of five test datasets

4.1.3 Reference Data

The four CS6 cameras of the RIEGL VMX-450 system captured digital colour images, which were used as the reference data for validation of the 2D window extraction in this study. The FOV of a RIEGL VMX-450-CS6 camera is 80˚×65˚(H×V), 5 mm lens. The trigger of the camera system has a constant time interval. The speed of the vehicle was about 20-30 km/h. So there are some overlaps between two adjacent images; the digital image which is selected to generate the orthophotos should the one who has the maximum coverage of the targeted

buildings. Accordingly, the digital images used in this study are shown in Figures 4-4(a), (b), (g), (h) and (m).

4.2 Results and Evaluation

4.2.1 Ground Removal

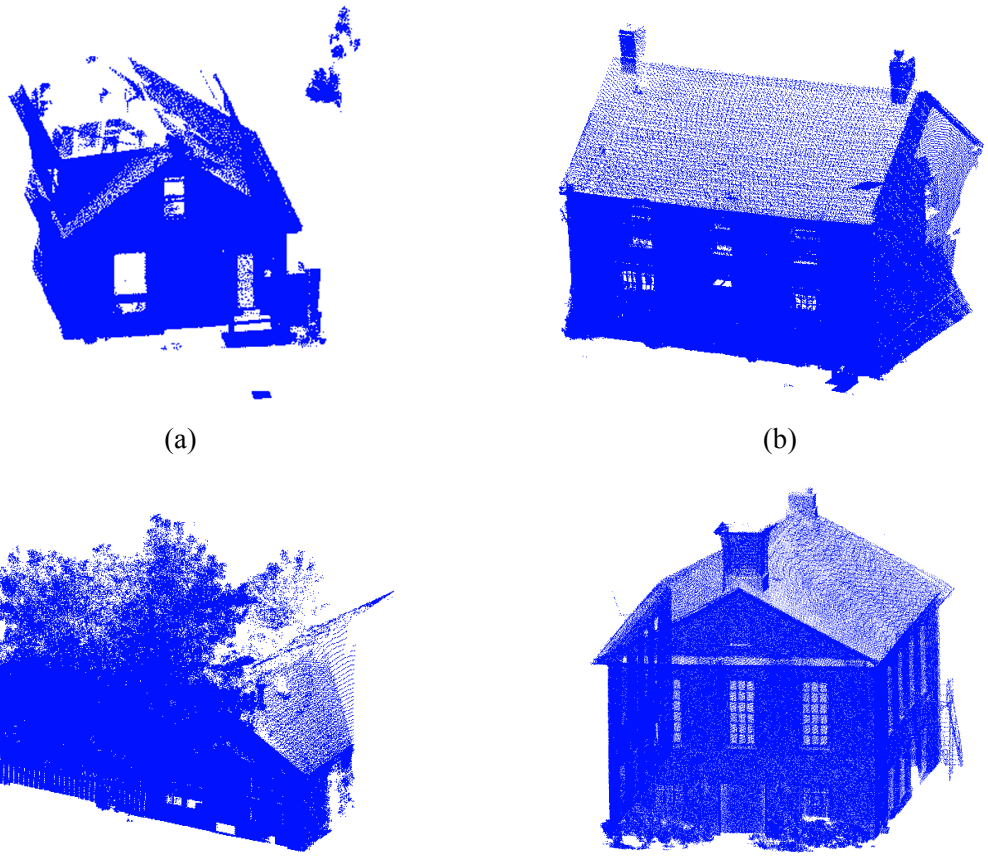
The parameters used in this section are listed below. As mentioned in Section 3.2, the results obtained by the voxel-based upward-growing algorithm is influenced by the following three parameters:

- w_v: size of a voxel,
- h_g: a pre-defined local ground undulation threshold, and
- h_e: a pre-defined global ground undulation threshold.

According to the point density of the test datasets, the voxel size was set as w_v = 0.5 m, h_g = 0.5 m, and h_e = 3.0 m, respectively, in this experiment. to ensure there are more than 500 points in one voxel to confirm the time complexity of this algorithm (since points on the ground have high density). Figure 4-6 shows the ground removal results obtained by the voxel-based upward-growing algorithm using the five test datasets. Table 2 lists the ground removal results, which show that removing ground points can considerably reduce the number of points and improve computing efficiency.

Table 2 Results of the ground removal

Dataset	Raw point cloud	Non-ground points	Percentage of ground points
1	787,235	339,105	57%
2	1,442,607	747,583	48%
3	1,281,313	865,063	32%
4	725,579	623,087	14%
5	3,336,229	1,061,303	68%



(a) (b)

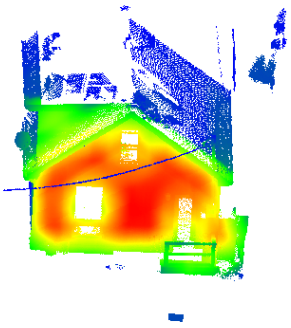
(c) (d)

(e)

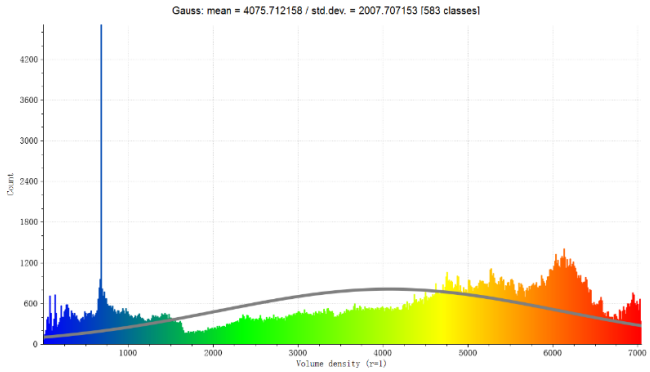
Figure 4-6 Ground removal results obtained using

(a) Dataset 1; (b) Dataset 2; (c) Dataset 3; (d) Dataset 4; (e) Dataset 5.

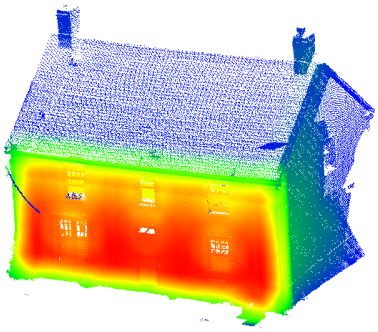
Figure 4-7 presents Gaussian point density analysis of non-ground points in the five test datasets. The average point densities of the non-ground points become 4,076, 5,920, 3,959, 1,643 and 1,628 pts/m^3 in Datasets 1 to 5.



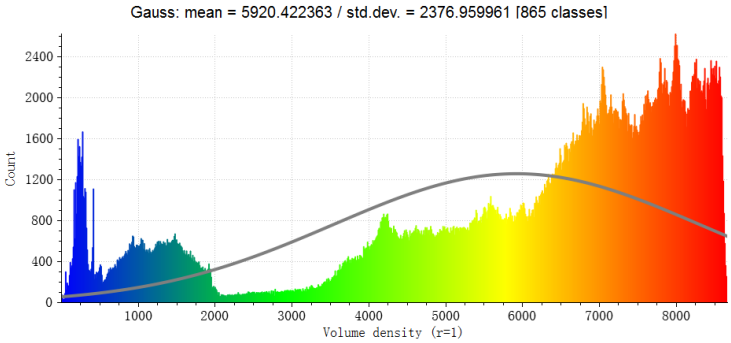
(a)



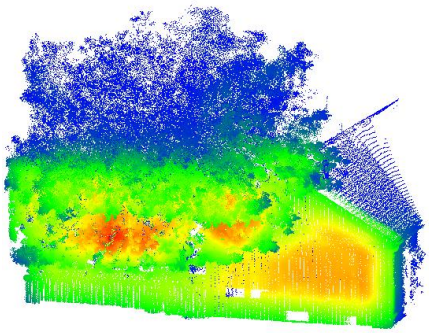
(b)



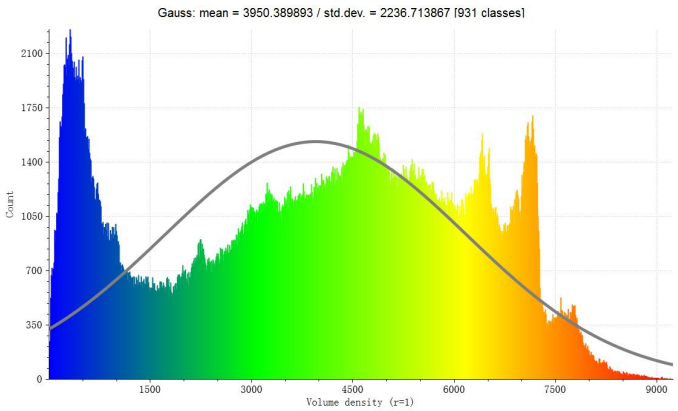
(c)


Gauss: mean = 5920.422363 / std.dev. = 2376.959961 [865 classes]
Count
Volume density (r=1)

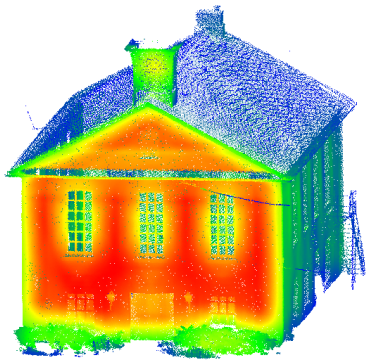
(d)



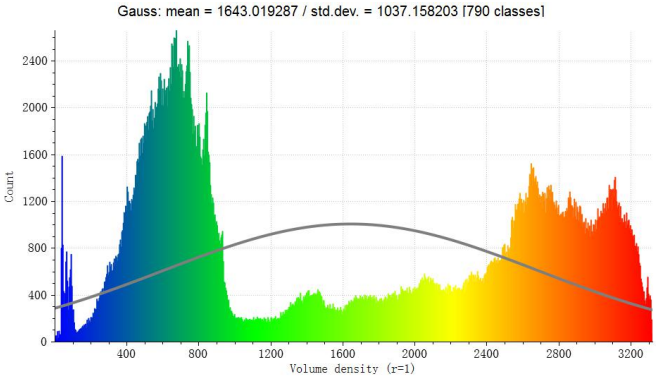
(e)


Gauss: mean = 3950.389893 / std.dev. = 2236.713867 [931 classes]
Count
Volume density (r=1)

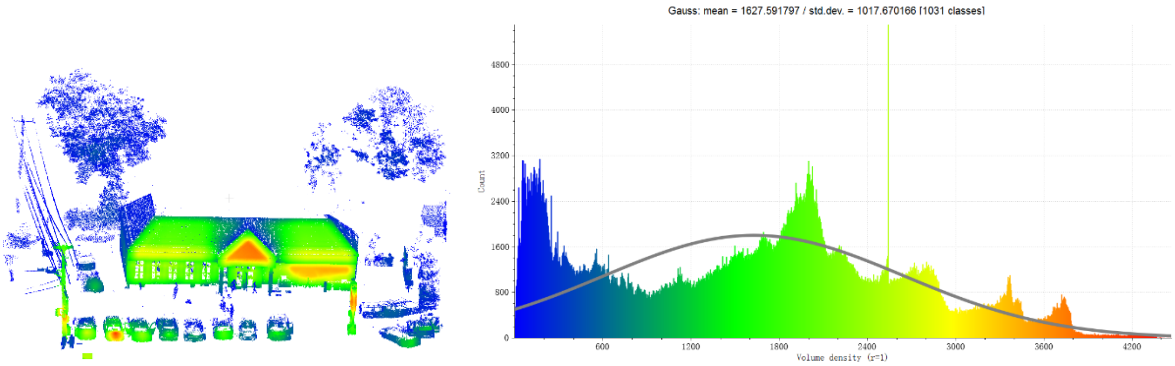
(f)



(g)


Gauss: mean = 1643.019287 / std.dev. = 1037.158203 [790 classes]
Count
Volume density (r=1)

(h)



(i) (j)

Figure 4-7 Gaussian density distribution of non-ground points

4.2.2 Noise Removal Using Statistical Analysis

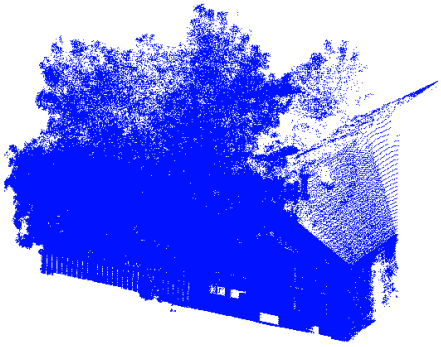
K-nearest neighbourhood should be searched before starting the algorithm, k should be set according to the raw point density. $k = 50$ is selected in this experiment according to the point density. Figures 4-8(a), (c), (e), (g) and (i) show the non-ground points of the five test datasets, while (b), (d), (f), (h) and (j) show the corresponding non-ground points after noise removal. As can be seen, discrete noises can be successfully removed from the test datasets. However, some points belonging to trees, utility lines, cars and rooftops are also removed. Those points belonging to rooftops in Datasets 2 and 4 are even removed because of their low point density. However, it will not make any difference to the final result, since the targeted objects of this methodology are windows on building facades. Points belonging to trees, utility lines, cars and rooftops can also be regarded as “noise points” and removed.

(a)

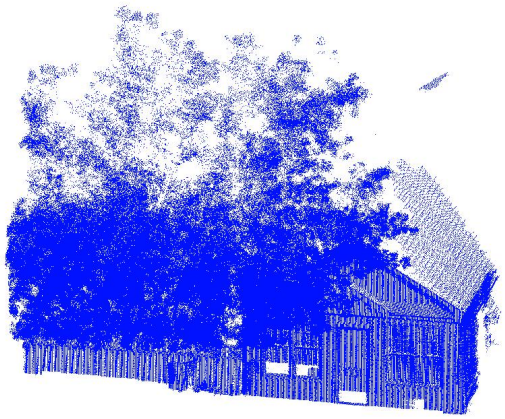
(b)

(c)

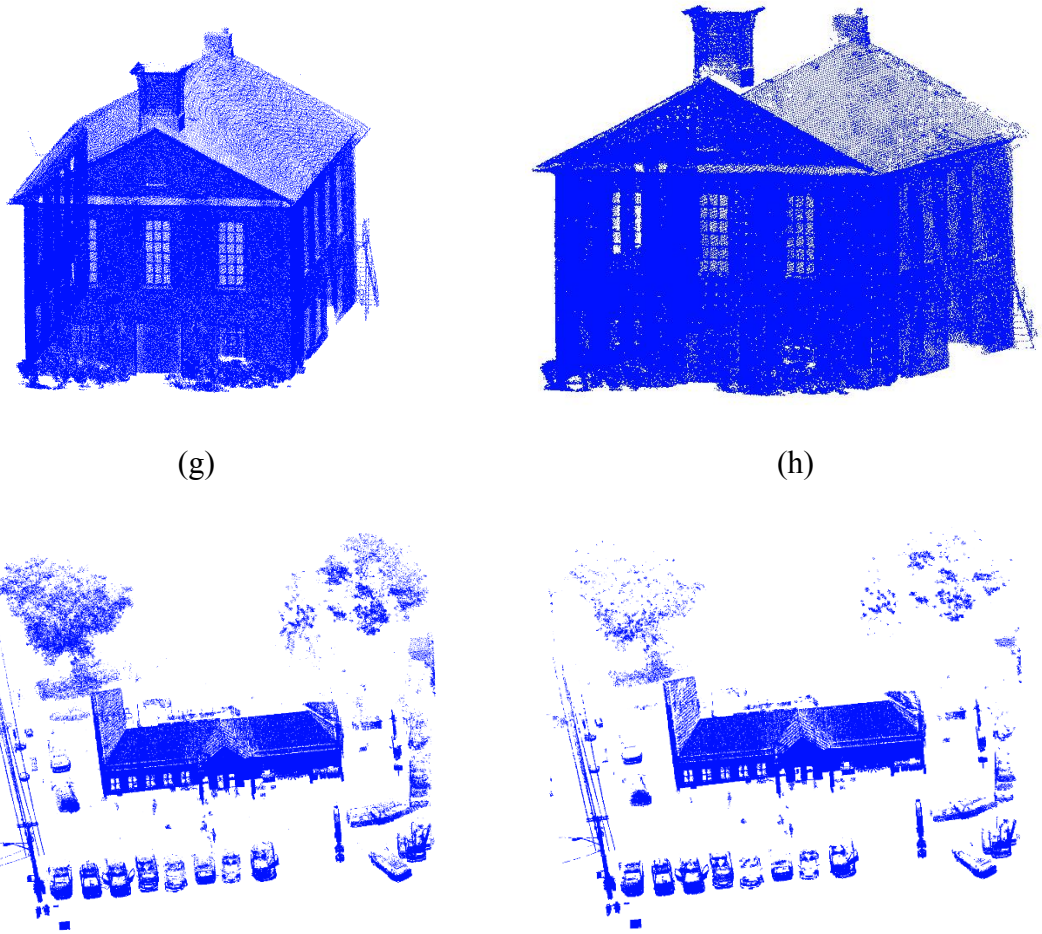
(d)



(e)



(f)

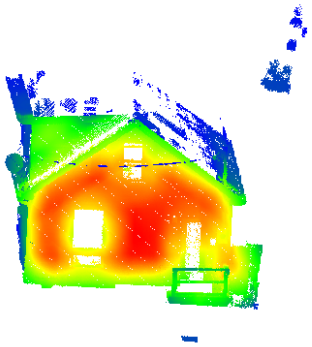


(g) (h)

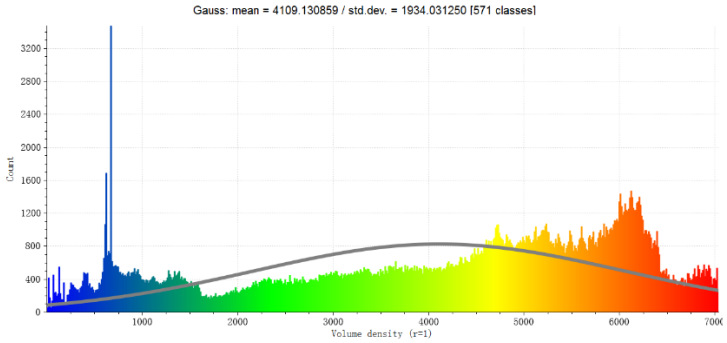
(i) (j)

Figure 4-8 Noise removal results

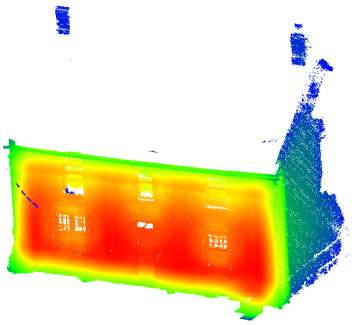
Discrete noise points can reduce the average point density of the entire point cloud after ground and noise removal. The Gaussian density distribution analysis for the five test datasets are displayed in Figure 4-9. As shown in Figures 4-9 (b), (d), (f), (h) and (j), the average density for the five test datasets become 4,109.1 pts/m3, 6,206.7pts/m^3, 4,129.4 pts/m^3, 1,647.4 pts/m^3, 1,741.1 pts/m^3, respectively. All of them are larger compared with the average density of non-ground points, which also indirectly suggests that sparse noise points are removed after processing.



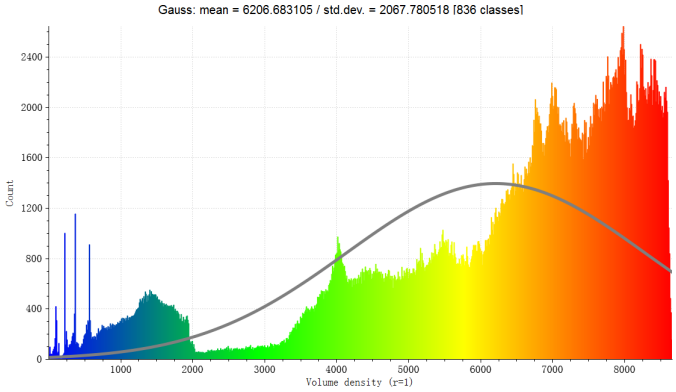
(a)


Gauss: mean = 4109.130859 / std.dev. = 1934.031250 [571 classes]
Count
Volume density (r=1)

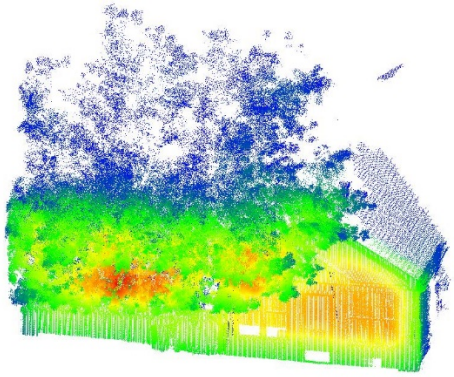
(b)



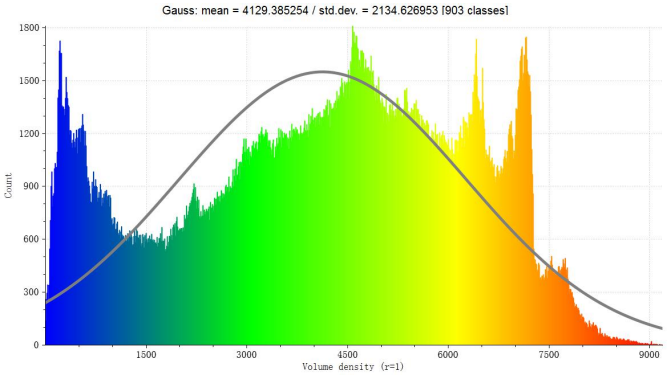
(c)


Gauss: mean = 6206.683105 / std.dev. = 2067.780518 [836 classes]
Count
Volume density (r=1)

(d)



(e)


Gauss: mean = 4129.385254 / std.dev. = 2134.626953 [903 classes]
Count
Volume density (r=1)

(f)

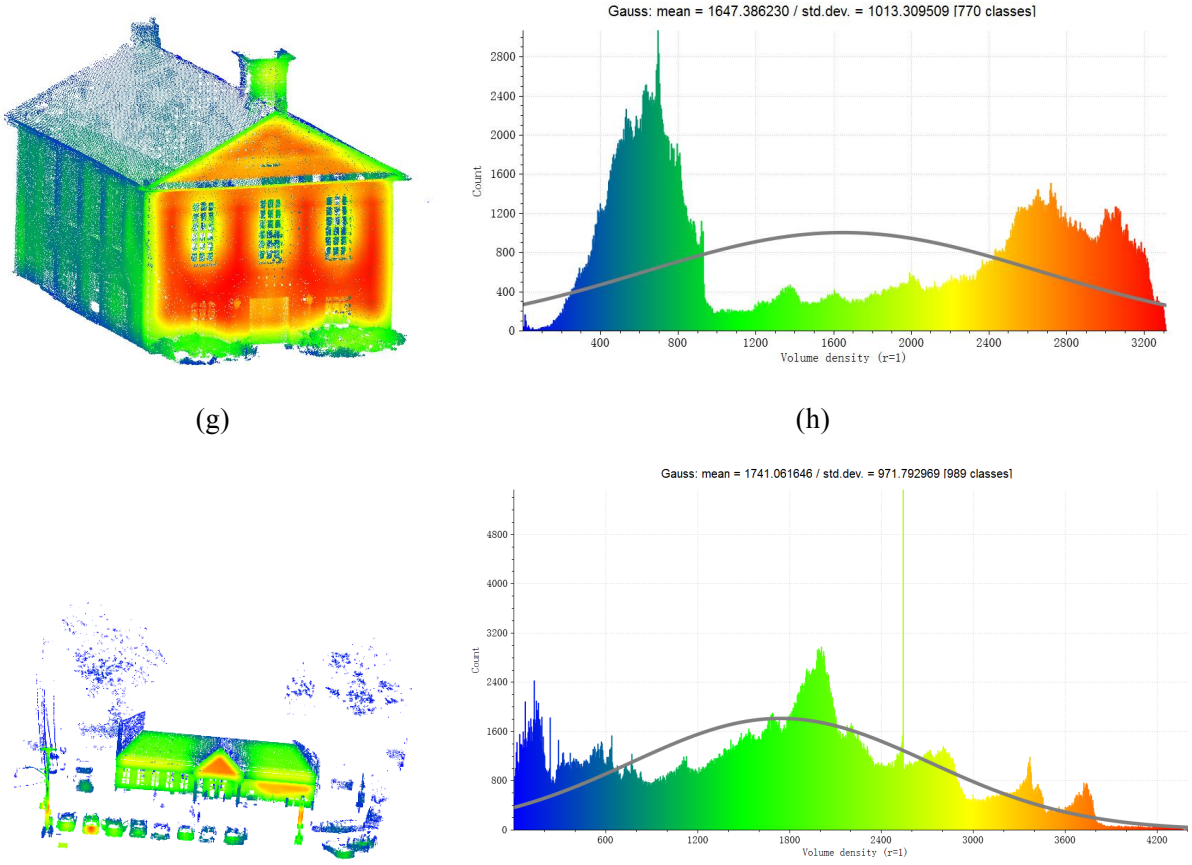


Figure 4-9 Gaussian density distribution of noise removal results

4.2.3 Classification by Conditional Euclidean Clustering

After ground and noise removal, the remained points still have no topological relationships. According to the Gaussian point density distributions of the five noise-removed results, d_c=0.1 m, h_c=1.0 m and w_c=2.0 m are used in this experiment, since the targeted building facades are always higher and wider than 2 m. As mentioned in Section 3.2, the results obtained by the conditional Euclidean clustering algorithm is influenced by the following three parameters:

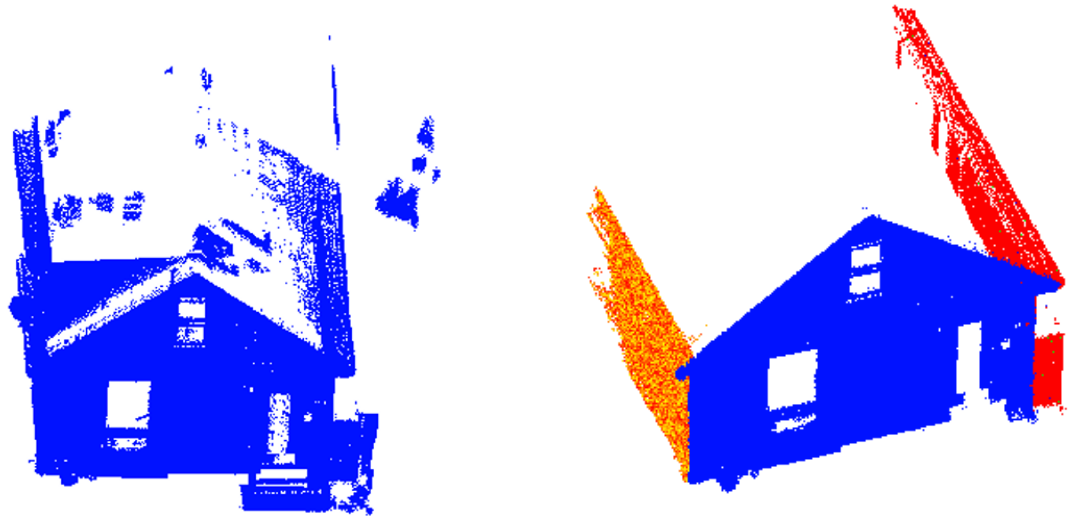
- d_c: a Euclidean distance threshold,

- h_c: a pre-defined height difference threshold
- w_c: a pre-defined width threshold.

Figures 4-10(a), (c), (e), (g) and (i) show the noise-removed datasets before applying the conditional Euclidean clustering algorithm. Figures 4-10(b), (d), (f), (h) and (j) show the clustering results of the five noise-removed datasets. As can be seen, sparse points (noise which still remains in the last procedure) and small clusters (such as stairs in front of the façade in Dataset 1, curtains and doors in Dataset 2, 3 and 4, and cars, trees, boats in Dataset 5) are all removed by setting the threshold h_c and w_c. Only a telegraph pole, an ad board, and house facades are remained in the five results. As shown in Figures 4-11(b), (d), (f), (h) and (j), the average density for the five test datasets become 4552.0 pts/m^3, 5666.4 pts/m^3, 3755.19 pts/m^3, 1564.33 pts/m^3, 2061.3 pts/m^3, respectively. Average point densities in Dataset 1 and 5 are larger than the average point densities of noise-removed datasets, which indicate that sparse points and small clusters are removed so that the remaining points have more compact geometry relationships after conditional Euclidean clustering. However, windows and doors in Dataset 2, 3, and 4 are concave on the front façade, they are not in the same plane as the front facade, so that the frames of windows and doors on the three datasets are removed together with curtains, resulting that the average point density become smaller than that in the noise-removed dataset.

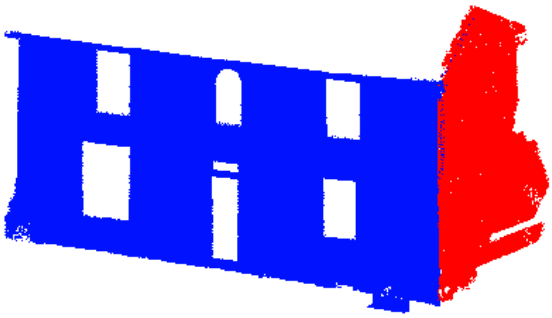
Disordered points have topological relationships now and clusters can be designated as semantic objects after conditional Euclidean clustering method. However, side facades are put into different clusters against front building facades, and some rooftops of houses are even

removed. After inspecting the raw noise-removed point cloud, it can be concluded because of scanning angles that there are gaps between side facades and building facades, rooftops and front facades, which are larger than the given threshold d_c. In addition, density of front facades, side facades and rooftops are also varied due to scanner angles and scanning distances from the scanning centre to targeted points. Therefore, points belonging to front facades, rooftops and side facades are assigned into various clusters. Especially, rooftops in Datasets 1, 3 and 5 are removed because their height differences (the substation result between the maximum z value and minimum z value) are smaller than the given threshold h_c.

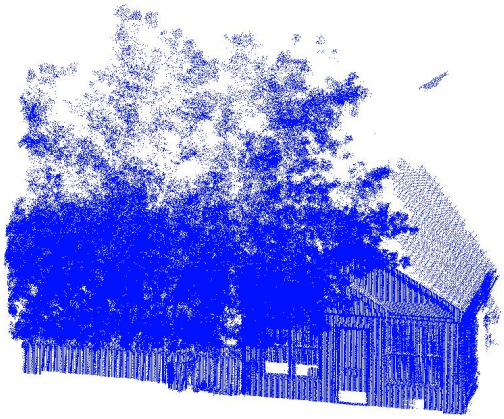


(a) (b)

(c)



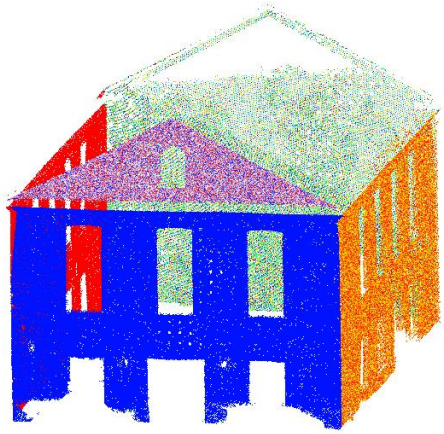
(d)



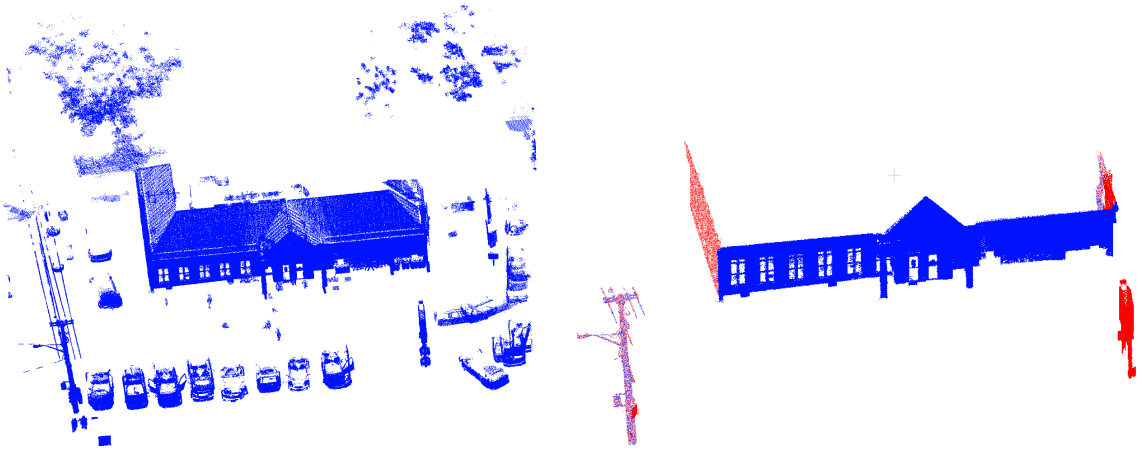
(e)

(f)

(g)

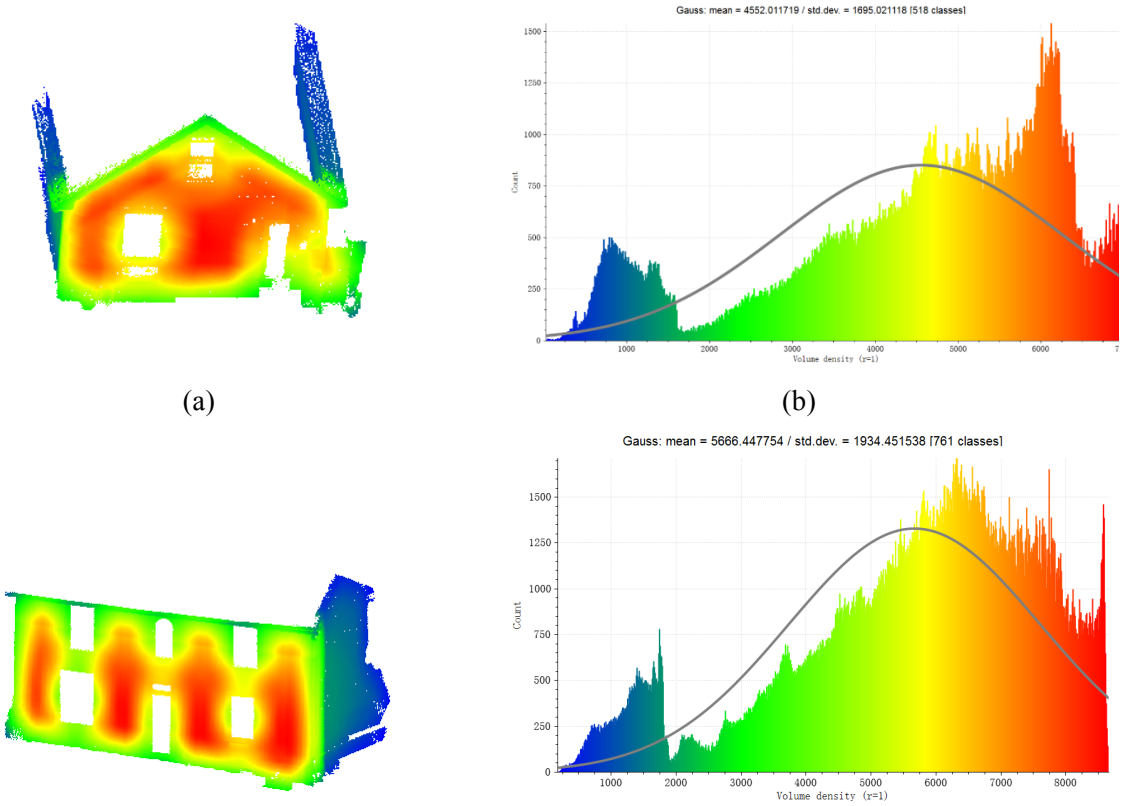


(h)



(i) (j)

Figure 4-10 Clustering results



(a) (b)

(c) (d)

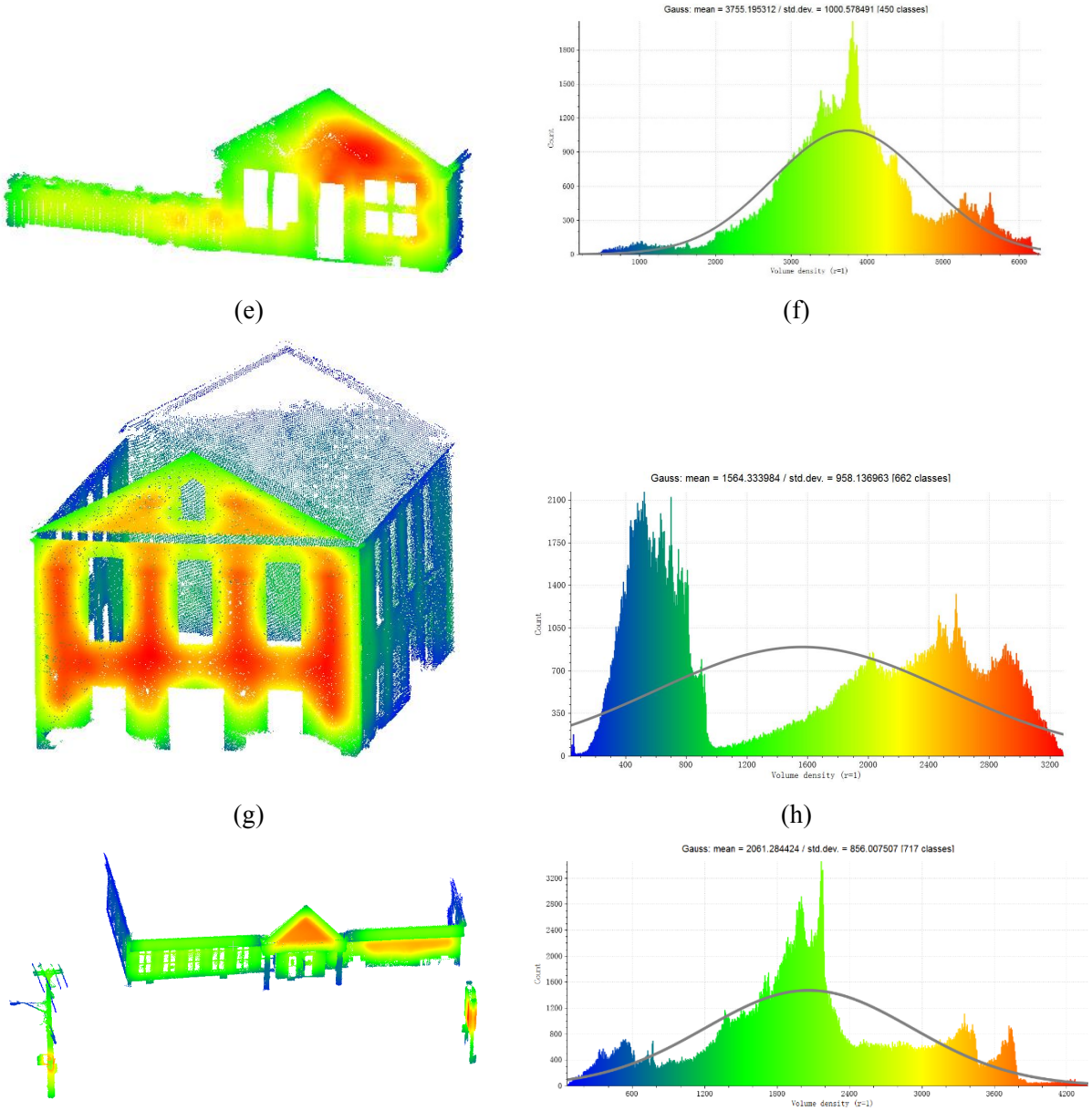


Figure 4-11 Gaussian density distribution of remaining clusters

4.2.4 Building Façade Extraction

When discrete points are segmented into different semantic clusters, distinguishing a specific semantic cluster from the others becomes much simpler. The results obtained by the density/width analysis is influenced by the following three parameters:

- Den_b: a pre-defined average density threshold of building facades,
- Wid_b: a pre-defined width threshold of building facades.

Figures 4-12 and 4-13 respectively indicate the ranges of widths and the densities of five typical features (trees, cars, building facades, pole-like objects and ad boards) in the study area. According to Figures 4-12 and 4-13, it can be concluded that width/ length of a house should not be smaller than 7 m, and the average density of front house facades in a X-Y plane should be larger than 4000 pts/m^2. Thereby, this experiment set Den_b=4000 pts/m^2, and Wid_b=8 m for all the test datasets.

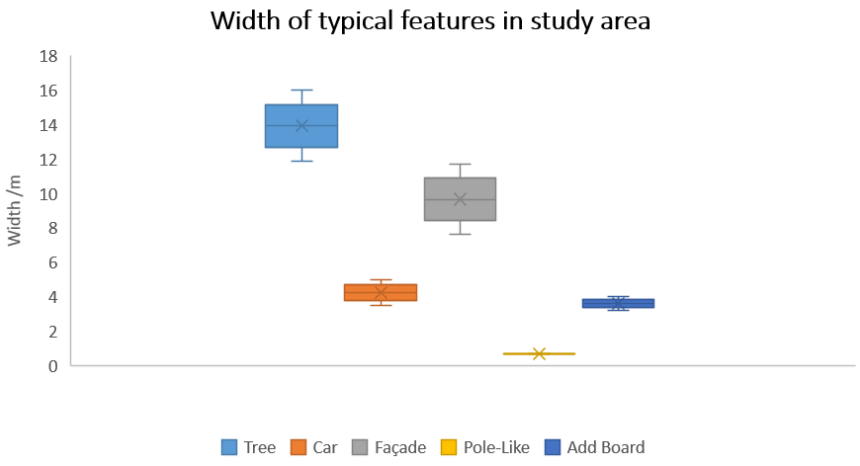


Figure 4-12 Width of five typical features in the study area

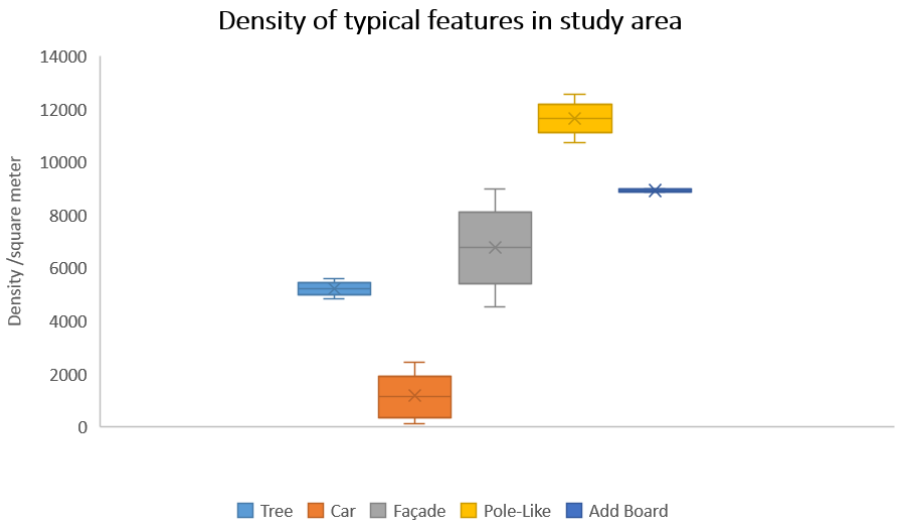


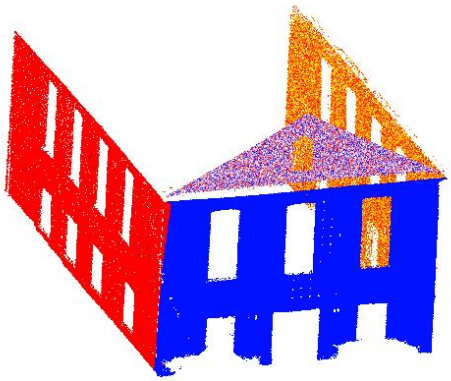
Figure 4-13 Density of five typical features in the study area

Figure 4-14 presents the extracted building facades of the test datasets. The front building facades in Datasets 1, 2, 3, and 5 can be elaborately differentiated from other clusters and extracted, while the side building facades are missing. The side building façade in Dataset 4 can be successfully extracted while its back façade is missing. After re-calculating the average planar density in the XoY plane of side facades, the result demonstrates that average planar densities of side facades in the test datasets except Dataset 4, and the back façade in Dataset 4 are lower than 1000 pts/m^2. Due to influences of scanner angles and scanning distances from a scanner centre to points on side facades, volumes of points on side facades cannot satisfy the given threshold, they are filtered out as other non-façade points in this Density/Width Analysis algorithm. Furthermore, holes and compact noises are still remaining on the extracted building facades according to insufficient points caused by system errors in MLS systems or curtains behind windows, which will inevitably have side-effects on the following procedures.

(a)

(b)

(c)

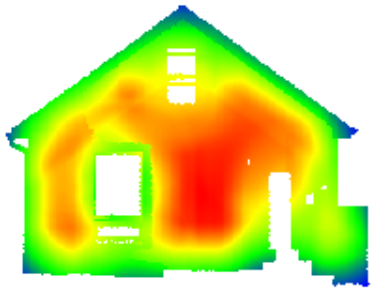


(d)

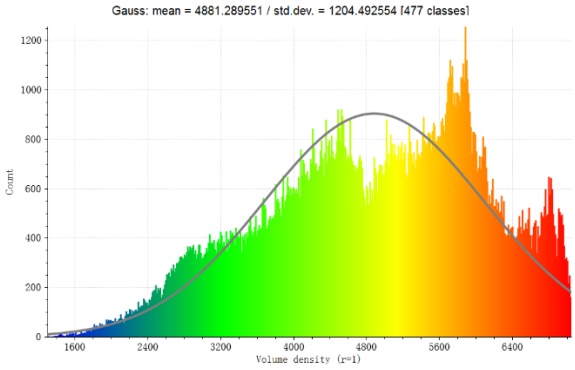
(e)

Figure 4-14 Building facades extracted from the five test datasets

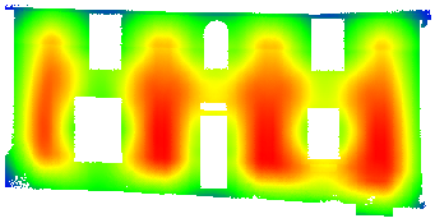
Figure 4-15 show the Gaussian point density of the extracted building facades. The average point densities of the front façade in the five test datasets become about 4,881, 6,013, 3,830, 1,702 and 2,163 pts/ m^3, respectively, which all become higher due to the more compact points.



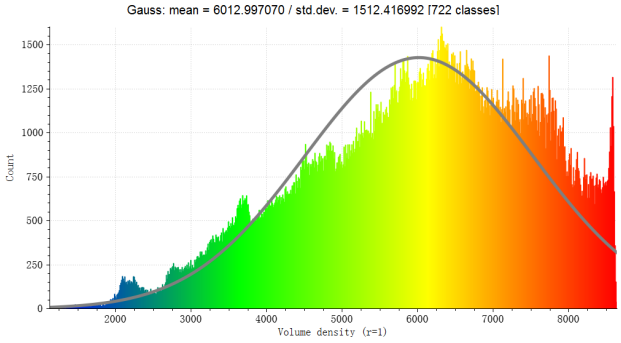
(a)



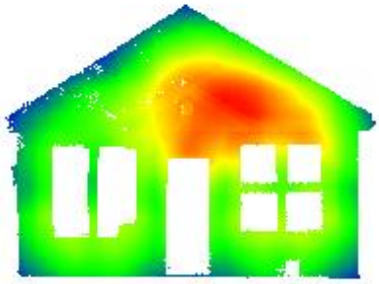
(b)



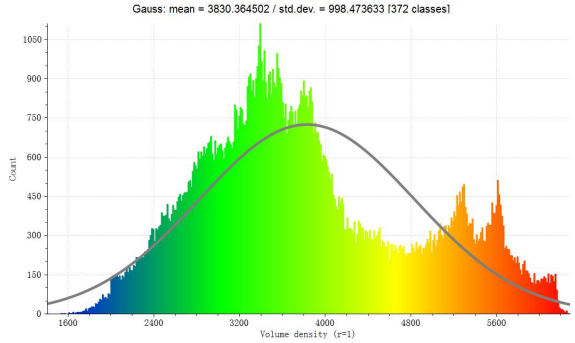
(c)



(d)



(e)



(f)

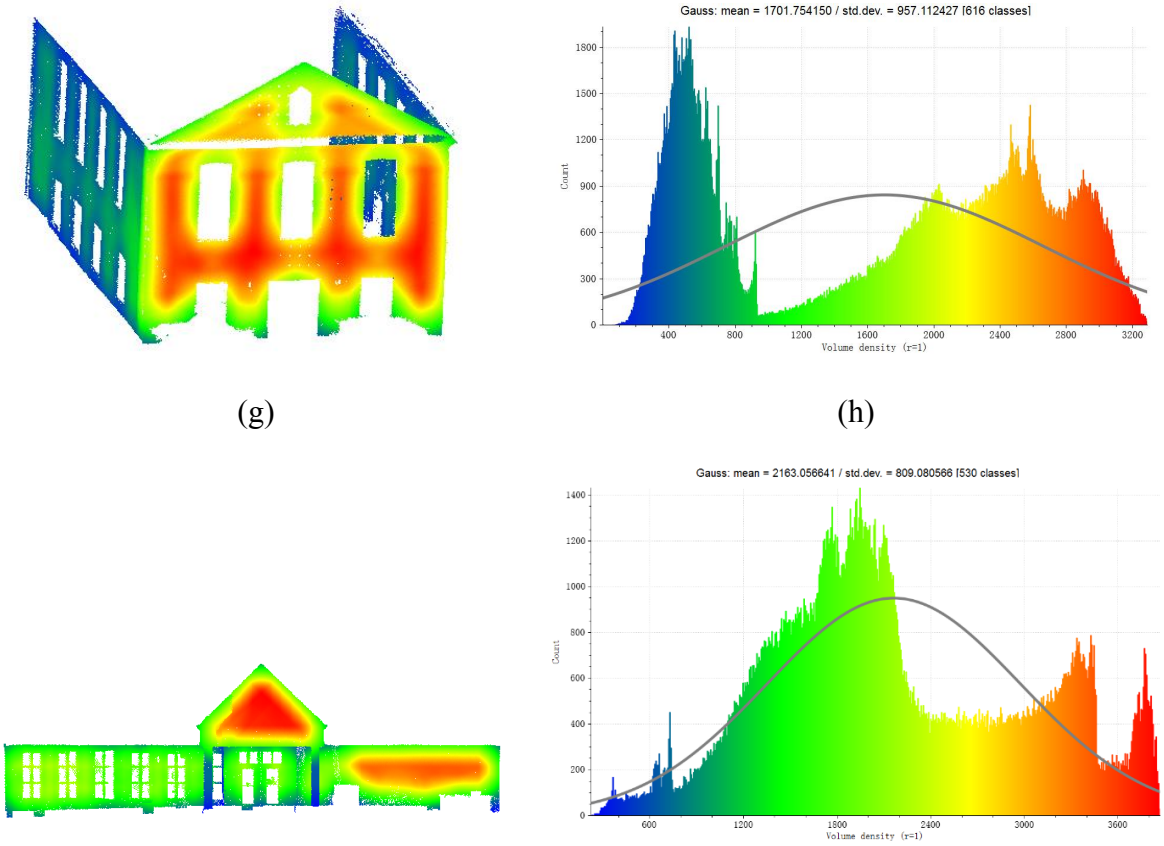


Figure 4-15 Gaussian density distribution of front facades

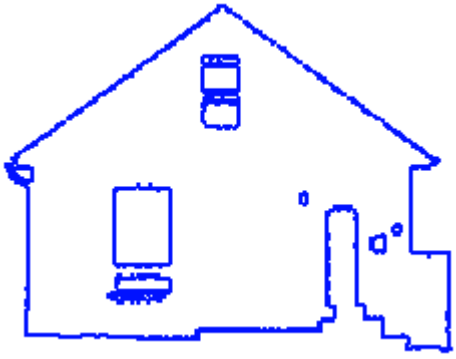
4.2.5 Window Extraction Using Hole Detection Algorithm

In this section, the size of a voxel still needs to be different from that used in ground removal, since the hole detection algorithm used in this section needs to elaborately examine each voxel and its neighbours. To guarantee the time complexity as well as the precision of this algorithm, there should be no more than 10 points in the same voxel. As shown in Figure 4-15, the highest density of the test front facades is about 9000 pts/m^3, so the size of a voxel

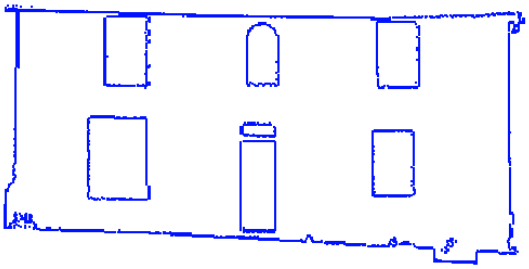
was set as v_g = 0.05 m in this section to make sure there cannot be more than 2 points in a voxel. The extracted windows are shown in Figure 4-16. The promising results indicate that the proposed window extraction algorithm can successfully extract all windows on the test datasets, including rectangular, irregular and arc-rounded windows. In addition, as shown in the results of Dataset 3, the proposed method is not influenced by occlusions of trees. However, as shown in Figures 4-16 (c), (d), (k) and (l), some big holes caused by systematic errors of MLS systems are also extracted from the raw point cloud. Furthermore, as shown in Figure 4-16 (l), only window frames on front façade in Dataset 4 can be extracted.

(a) Dataset 1 in digital images

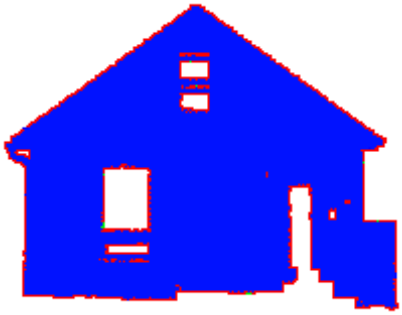
(b) Dataset 2 in digital images



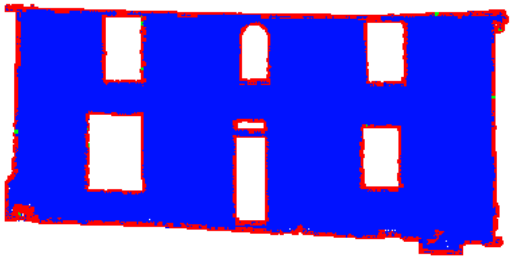
(c) Extracted windows and building outliers of Dataset 1



(d) Extracted windows and building outliers of Dataset 2



(e) Extracted windows overlapped on the front façades of Dataset 1

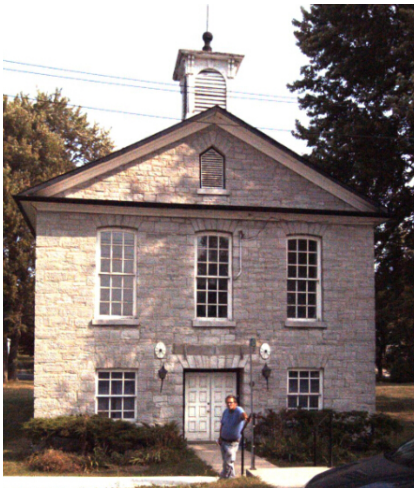


(f) Extracted windows overlapped on the front façades of Dataset 2

(g) Extracted windows overlapped on raw point clouds of Dataset 1

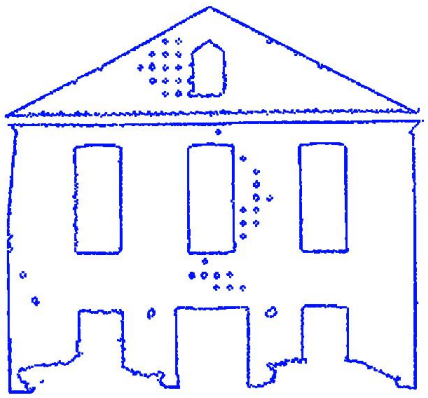
(h) Extracted windows overlapped on raw point clouds of Dataset 2

(i) Dataset 3 in digital images



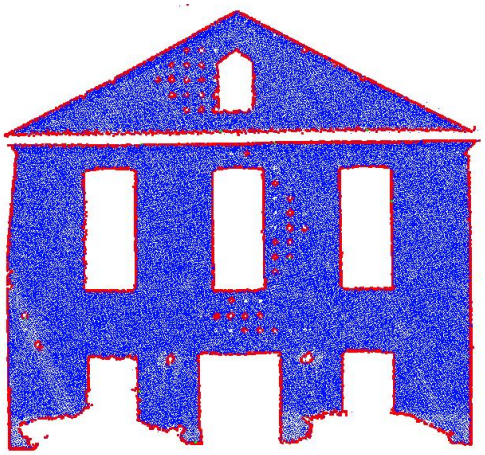
(j) Dataset 4 in digital images

(k) Extracted windows and building outliers of Dataset 3

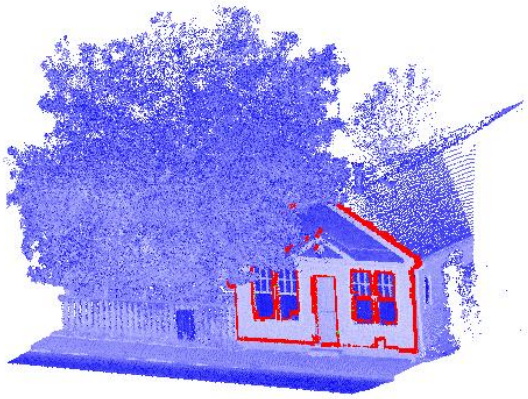


(l) Extracted windows and building outliers of Dataset 4

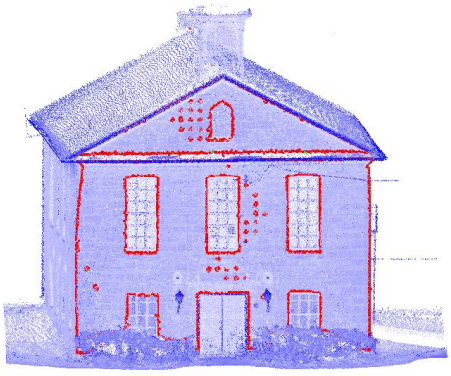
(m) Extracted windows overlapped on the front façades of Dataset 3



(n) Extracted windows overlapped on the front façades of Dataset 4

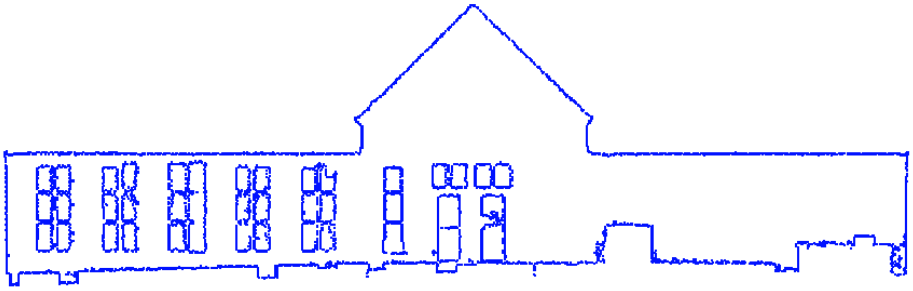


(m) Extracted windows overlapped on raw point clouds of Dataset 3

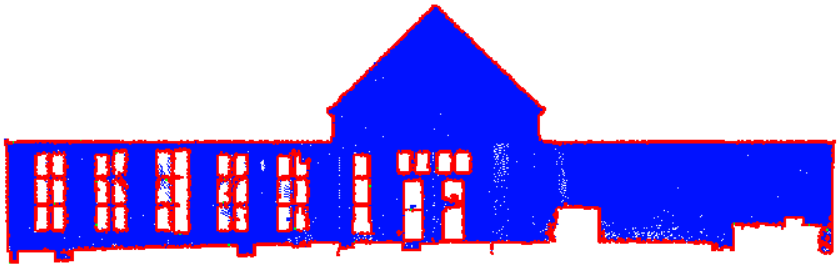


(n) Extracted windows overlapped on raw point clouds of Dataset 4

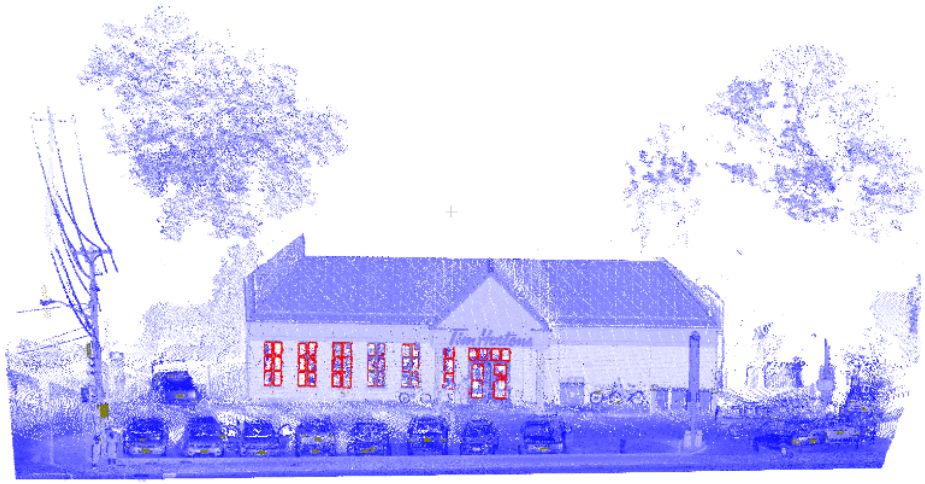
(o) Dataset 5 in digital images



(p) Extracted windows and building outliers of Dataset 5



(q) Extracted windows overlapped on the front façades of Dataset 5



(r) Extracted windows overlapped on raw point clouds of Dataset 5

Figure 4-16 Extracted windows

In addition, Table the efficiency of the proposed method is shown in Table 5. For each dataset, the consuming time of each procedure has been calculated. The computer used in this study is a Mac Pro with an 8 GB RAM, and an Intel i5-4728U CPU. The consuming time of each procedure suggests that the efficiencies of the proposed building façade extraction algorithm and window extraction algorithm are both high. As mentioned in Chapter 1, the objectives of this study are mainly focused on developing semi-automatic algorithms of building façade extraction and window extraction, so the algorithms of noise removal and clustering directly took advantage of the commercial programming package and Yu's (Yu et al., 2015) work. Total efficiency of the proposed method can be further improved by modifying the algorithms of noise removal and clustering.

Table 3 Efficiency of the proposed method

Dataset	1	2	3	4	5
Size	26 MB	50MB	41.5 MB	23.5 MB	164MB
Point Number	787,235	1,442,607	1,281,313	725,579	3,336,229
Ground Removal	1.5s	2.1s	1.8s	1.3s	6.4s
Noise Removal	19.3s	45.3s	40.4s	17.6s	199.9s
Clustering	40.2s	74.1s	65.9s	27.2s	217.7s
Façade Extraction	9.0s	13.1s	11.4s	7.2s	20.7s
Window Extraction	5.5s	6.5s	5.7s	3.2s	12.5s
Total Time	75.5s	140.2s	125.2s	56.5s	457.2s

4.3 Results in Accuracy Assessment

4.3.1 Window Regions in 2D

Figures 4-17 (a), (c), (e), (g) and (i) show the manual interpretation results in generated orthophotos in ArcGIS 10.2.2. Figures 4-17 (b), (d), (f), (h) and (j) present the orthophotos overlapped by extracted window points. As mentioned in Chapter 3, the manually interpreted orthophoto and the overlapped orthorphoto are set in the same resolution. After pixel analysis of Figure 4-17, the correctness (*crt*), completeness (*cpt*) and *F-measure* of these five test datasets are listed in Table 4. As shown in the Table 4, the correctness, completeness, and *F-measure* of Dataset 1 are relatively low since big holes in the raw point cloud are also extracted by the hole-detection algorithm. In Datasets 2, 3 and 4, the correctness is adequately high but the completeness is relatively low. As shown in Figure 4-17(c), (e) and (g), curtains are

dropped down when the data was collecting; in addition, the windows are recessed on the building façade and they are not on the same vertical plane as the building façade. Therefore, only points of outer frames of the windows are put into the same cluster as the building façade (see Figure 4-10(d) in Section 4.2.3). As a result, the outer frames of the windows in the three datasets can be extracted, while the inner crossbars of the windows are removed. The completeness, correctness, and *F-measure* of Dataset 5 are all very high, since there are no impacts of curtains, holes, or occlusions on the raw point cloud.

Table 4 2D performance evaluation

Dataset	Correctness	Completeness	*F-measure*
1	79.67%	95.4%	0.868
2	97.60%	85.17%	0.910
3	95.42%	72.74%	0.826
4	88.54%	63.51%	0.740
5	97.79%	97.63%	0.977

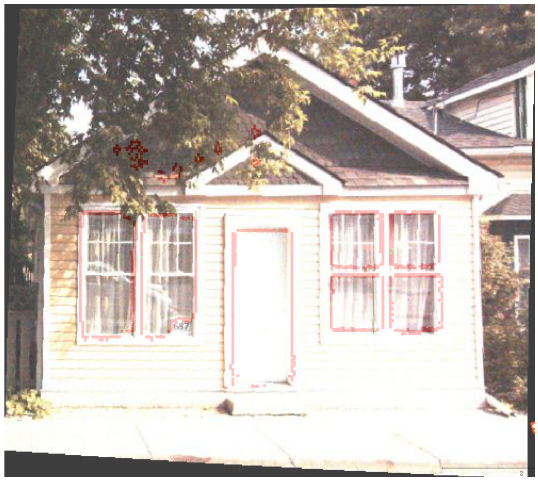
(a) Manual interpreted polygon in the orthophoto of Dataset 1

(b) Orthophoto overlapped by extracted window points of Dataset 1

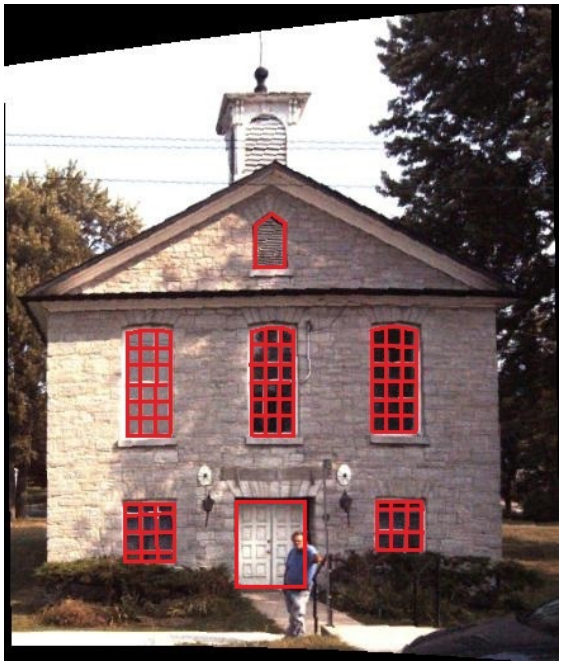
(c) Manual interpreted polygon in the orthophoto of Dataset 2

(d) Orthophoto overlapped by extracted window points of Dataset 2

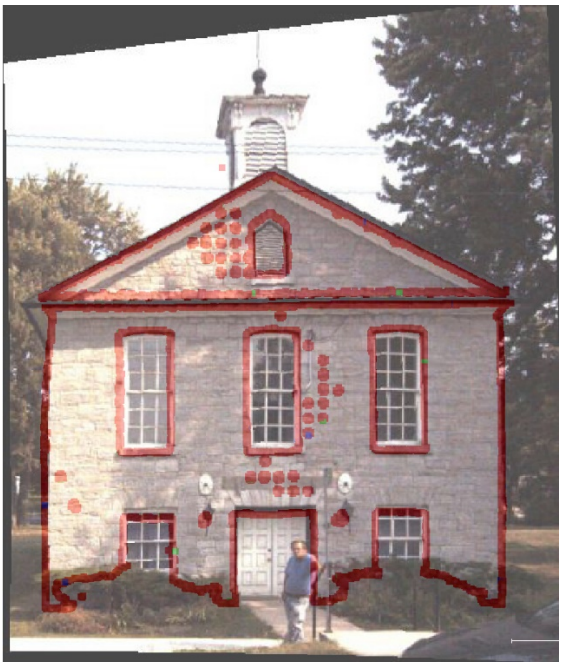
(e) Manual interpreted polygon in the orthophoto of Dataset 3



(f) Orthophoto overlapped by extracted window points of Dataset 3



(g) Manual interpreted polygon in the orthophoto of Dataset 4



(h) Orthophoto overlapped by extracted window points of Dataset 4

(i) Manual interpreted polygon in the orthophoto of Dataset 5

(j) Orthophoto overlapped by extracted window points of Dataset 5

Figure 4-17 2D validation results

4.3.2 Window Regions in 3D

The results of 3D performance evaluation are listed in Table 5. Compared with the performance evaluation results in Table 4, it can be concluded that the values of the correctness in the test datasets in 2D and 3D are basically aligned with each other. The correctness values of Datasets 2, 3 and 5 are 95.34%, 96.23%, and 97.96%, respectively. Such results prove that the proposed method can extract accurate 3D window frames when there are no defects in the raw point clouds. However, the correctness values of Datasets 1 and 4 are 79.58%, and 86.12%, respectively, which are lower among the test datasets. It again suggests that the big holes caused by system errors have considerably negative effects on the proposed method.

Table 5 3D performance evaluation

Dataset	C_n	R_n	Correctness
1	3035	3814	79.58%
2	8638	9061	95.34%
3	7000	7274	96.23%
4	4495	5226	86.12%
5	7445	7600	97.96%

4.4 Chapter Summary

This chapter presents the experimental results of the proposed method. According to the performance evaluations in 2D and 3D, it can be concluded that the proposed method can be successfully applied in extracting rectangular and curved windows with high accuracy in the test datasets. However, for those windows with curtains drawn or concave on walls, the proposed method can only extract the outer frames of the windows. In addition, point defects in the raw point cloud will also have considerably negative impact on the final result. Holes caused by system errors of MLS systems will also be extracted by the proposed hole detection algorithm.

This chapter also provides the efficiency analysis of the proposed method. It indicates that the efficiencies of the proposed algorithms of building façade extraction and window extraction are high, however, the total efficiency of the proposed method should be improved in the future by modifying algorithms of noise removal and clustering.

Chapter 5 Conclusions and Recommendations

5.1 Conclusions

According to its unique data structure and convenient data collection methods, MLS systems have enormous commercial potentials in applications of the future LoD3 building modeling. MLS systems utilize active laser imaging technique to avoid influences of lighting conditions so that data can be collected both day and night. And by employing integrated position and orientation technology, MLS systems can directly acquire geo-referenced, high-density and highly accurate datasets. However, discrete points collected by MLS systems are in large volumes, evenly distributed and have no topological or semantic relationships with each other. Therefore, there is still no sophisticated commercial software nowadays, which can successfully handle and interpret information contained in MLS point clouds for practical applications.

To provide a credible approach for the LoD3 building modeling, this study proposes a method that can semi-automatically extract 3D window points from MLS point clouds. Non-ground points are filtered out from noise and ground points by using voxel-based upward-growing algorithm and statistical analysis. Then semantic clusters are segmented by conditional Euclidean clustering algorithm. Next, building facades are extracted using density/width algorithm. Eventually, a manipulator is applied according to the structural characteristics of window frames to extract the potential window points.

Five datasets are tested in this study to prove the feasibility of the proposed method. The *F-measure* of Dataset 1, 2, 3, 4 and 5 in the 2D validation are 0.737, 0.910, 0.826, 0.740 and 0.977. The correctness of the five test datasets in 3D validation are 79.58%, 95.34%, 96.23%,

86.12% and 97.96%. After detailed analysis of the experimental results, it demonstrates that the proposed method in this thesis can successfully extract all types of 3D windows and glass doors (including rectangular, irregular and arc-rounded ones) from the test datasets with promising accuracies. However, for those windows with curtains drawn, concave in walls or there being holes in raw point clouds, the accuracy of the proposed method will be influenced. In addition, the experimental results also suggest that the proposed method is not influenced by tree occlusions.

Generally, this thesis provides a promising method to extract 3D building windows for LoD3 building models. It offers a potential idea to meet the demands of commercial companies, such as Google and Microsoft, who are endeavoring to establish realistic 3D city models nowadays.

5.2 Contributions

The method put forward in this study makes several contributions to LoD3 modeling; generally, the objectives stated in Chapter 1 have been fulfilled. The detailed contributions can be described in the following aspects:

1. A semi-automated 3D building façade segmentation algorithm in MLS point clouds has been presented. As discussed in Chapter 3, the density and width properties of building facades in 3D point clouds facilitate to segment facades against other non-ground features. This algorithm reduces accuracy loss by segmenting facades directly from 3D point clouds. It also has higher computing efficiency over those previous methods which applied planar growing algorithms.

2. A revised hole detection algorithm to extract windows from MLS point cloud has been proposed. Windows are regarded as holes in this algorithm. By introducing the characteristics of window frames, the algorithm functions effectively in extracting windows directly from 3D point clouds. Furthermore, the algorithm can be applied in extracting all types of windows, including rectangular, arc-rounded and irregular ones.

5.3 Limitations and Recommended Further Works

According to the experimental results in Chapter 4, some limitations still exist in the proposed method. Therefore, some recommendations are made in order to the conquer the limitations of the proposed method. The details can be described as follows:

(1) Side facades are filtered out by conditional Euclidean clustering algorithm and density/width analysis. Because points clouds covering side facades usually have relatively low density than those of front facades. Therefore, the building façade segmentation method proposed in this study can only be applied in extracting front facades. However, side facades are also important in LoD3 building models. Related works on extracting side facades are still needed to be continued.

(2) Big holes caused by system errors of laser scanners are also extracted by the proposed hole detection algorithm. Thus more works are still needed to remove the extracted holes. Some algorithms in computer vision, such as contour tracing, have already been proposed to distinguish circular or semi-circular contour lines in point clouds.

(3) Doors recessed in building facades or those with glass are also extracted by the proposed method. There is no effective method to remove doors from the extracted window frames so far. More studies should be involved in the future.

(4) There are no high-rise buildings in the test study area. Therefore, the proposed method is only proved in houses and townhouses. Point cloud density is influenced by the distance between the centre of the laser scanner and the targeted point. Point clouds on higher floors may have lower densities. It may influence the performance of the conditional Euclidean clustering algorithm. Therefore, in the future, tests on high-rise buildings are needed to test the application range of the proposed method.

(5) The efficiency of the proposed method is still need to be improved. As mentioned in Chapter 4, the noise removal algorithm and the clustering algorithm completely depend on others' work. However, they are proved to be time-consuming in the test datasets. Therefore, the algorithms of removing noise and clustering still warrant discussion.

(6) The extracted window points can provide real 3D geographic information. For some applications which need aesthetic 3D building models, these extracted points still need to be regularized with digital images afterwards.

References

Aijazi, A. K., Checchin, P. & Trassoudaine, L., 2014. Automatic detection and feature estimation of windows in 3D urban point clouds exploiting façade symmetry and temporal correspondences. International Journal of Remote Sensing, 35(22), 7726-7748.

Arachchige, N. H., Perera, S. N, & Maas, H. G., 2012. Automatic processing of mobile laser scanner point clouds for building facade detection. ISPRS Archives, 39(B5), 187-192.

ArcGIS, 2016. ArcGIS Help 10.2, 10.2.1, and 10.2.2. Retrieved from ArcGIS Resources: http://resources.arcgis.com/en/help/main/10.2/index.html#//009t000000mn000000

Atlassian, 2016. World Geodetic System 1984. Retrieved from Hydrographic and Marine Software Solutions: https://confluence.qps.nl/pages/viewpage.action?pageId=29855173

Barrier Free Environments, 2010. The Accessible Housing Design File. John Wiley & Sons.

Brenner, C., 2009. Extraction of features from mobile laser scanning data for future driver assistance systems. Advances in GIScience, Springer, 25-42.

Canada Mortage and Housing Corporation, 2015. Accessible Housing by Design—House Designs. Retrieved from Canada Mortage and Housing Corporation: https://www.cmhc-schl.gc.ca/odpub/pdf/66093.pdf

Cheng, L., Gong, J., Li, M. & Liu, Y., 2011. 3D building model reconstruction from multi-view aerial imagery and LiDAR data. Photogrammetric Engineering & Remote Sensing, 77(2), 125-139.

Chrysler, C., 2014. Chrysler Active Park Assist Demo. Retrieved from Youtube : https://www.youtube.com/watch?v=5IiffHMGxEc

Defense Mapping Agency, 2015. Supplement to DoD WGS 84 Technical Report. Retrieved from Official diagram of the WGS 84 Reference Frame: http://earth-info.nga.mil/GandG/publications/tr8350.2/TR8350.2-b/Sections%201-5.pdf

Demir, N. & Baltsavias, E., 2012. Automated modeling of 3D building roofs using image and LiDAR data. ISPRS Annuals, Melbourne, Australia, 1(4).

Frankl, A., Zwertvaegher, A., Poesen, J. & Nyssen, J., 2013. Transferring Google Earth observations to GIS-software: example from gully erosion study. International Journal of Digital Earth, 6(2), 196-201.

Gannes, L., 2014. Here's What It's Like to Go for a Ride in Google's Robot Car. Retrieved from Recode: http://recode.net/2014/05/13/googles-self-driving-car-a-smooth-test-ride-but-a-long-road-ahead/

GIM, 2013. Mobile LiDAR Mapping Focus for International LiDAR Mapping Forum. Retrieved from http://www.gim-international.com/content/article/mobile-lidar-mapping-focus-for-international-lidar-mapping-forum?output=pdf

Google, 2016. See Notes on Google Earth Releases. Retrieved from Earth Help: https://support.google.com/earth/answer/40901?hl=en HYPERLINK "https://support.google.com/earth/answer/40901?hl=en&ref_topic=2376010"& HYPERLINK "https://support.google.com/earth/answer/40901?hl=en&ref_topic=2376010"ref_topic=2376010

Google, 2016. Get to Know Google Earth. Retrieved from Earth Help: https://support.google.com/earth/answer/148176?hl=en

Gruen, A., Huang, X., Qin, R., Du, T., Fang, W., Boavida, J., & Oliveira, A., 2013. Joint processing of UAV imagery and terrestrial mobile mapping system data for very high resolution city modeling. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 40(1/W2), 175-182.

Guan, H., 2013. Automated Extraction of Road Information from Mobile Laser Scanning Data. Retrieved from Doctoral dissertation, University of Waterloo: https://uwspace.uwaterloo.ca/handle/10012/8273

Guan, H., Yu, Y., Ji, Z., Li, J., & Zhang, Q., 2015. Deep learning-based tree classification using mobile LiDAR data. Remote Sensing Letters, 6(11), 864-873.

Guercke, R., Götzelmann, T., Brenner, C., & Sester, M., 2011. Aggregation of LoD 1 building models as an optimization problem. ISPRS Journal of Photogrammetry and Remote Sensing, 66(2), 209-222.

Guizzo, E., 2011. How Google's Self-Driving Car Works. Retrieved from IEEE Spectrum Online: http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works

Haala, N., & Kada, M., 2010. An update on automatic 3D building reconstruction. ISPRS Journal of Photogrammetry and Remote Sensing, 65, 570-580.

Haala, N., Peter, M., Kremer, J., & Hunter, G., 2008. Mobile LiDAR mapping for 3D point cloud collection in urban areas—a performance test. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 37, 1119-1127.

Huang, X., Gruen, A., Du, R. Q., & Fang, W., 2013. Integration of mobile laser scanning data with UAV imagery for very high resolution 3D city modeling. International Symposium on Mobile Mapping Technology, Tainan, Taiwan, 1-3.

Hyyppä, J., Jaakkola, A., Chen, Y., & Kukko, A., 2013. Unconventional LiDAR mapping from air, terrestrial and mobile. Proceedings of the Photogrammetric Week, Stuttgart, Germany, 205-214.

Jochem, A., Höfle, B., & Rutzinger, M., 2011. Extraction of vertical walls from mobile laser scanning data for solar potential assessment. Remote Sensing, 3(4), 650-667.

Kane, S., 2012. 2012 Family Cars with Self-Parking Technology. Retrieved from The Car Connection: http://www.thecarconnection.com/news/1067819_2012-family-cars-with-self-parking-technology

Lavrinc, D., 2012. Exclusive: Google Expands Its Autonomous Fleet With Hybrid Lexus RX450h. Retrieved from Wired: http://www.wired.com/2012/04/google-autonomous-lexus-rx450h/

Leick, A., Rapoport, L., & Tatarnikov, D., 2015. GPS Satellite Surveying. John Wiley & Sons.

Lemmens, M., 2011. Geo-information: Technologies, Applications and The Environment. Springer Science & Business Media.

Lichti, D. D., 2010. Terrestrial laser scanner self-calibration: correlation sources and their mitigation. ISPRS Journal of Photogrammetry and Remote Sensing, 65(1), 93-102.

Marshall, G. F., 2011. Hand Book of Optical and Laser Scanning. CRC Press.

Melzer, T., & Briese, C., 2004. Extraction and Modeling of Power Lines from ALS Point Clouds. na.

Mesolongitis, A., & Stamos, I., 2012. Detection of windows in point clouds of urban scenes. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, RI, USA, 17-24.

Nam, T., & Pardo, T. A., 2011. Conceptualizing smart city with dimensions of technology, people, and institutions. Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times, MD, USA, 282-291.

Nguatem, W., Drauschke, M., & Mayer, H., 2014. Localization of windows and doors in 3D point clouds of facades. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Zurich, Switzerland, 2(3), 87-89.

Nixon, M. S., & Aguado, A. S., 2012. Feature Extraction & Image Processing for Computer Vision. Academic Press.

Olsen, M. J., 2013. Guidelines for the Use of Mobile LiDAR in Transportation Applications. Transportation Research Board.

Osborne, C., 2015. Google's autonomous car injuries: blame the human. Retrieved from ZDNet: http://www.zdnet.com/article/googles-autonomous-car-injuries-blame-the-human/

Over, M., Schilling, A., Neubauer, S., & Zipf, A., 2010. Generating web-based 3D city models from Open Street Map: the current situation in Germany. Computers, Environment and Urban Systems, 34(6), 496-507.

Paromtchik, I. E., & Laugier, C., 1996. Motion generation and control for parking an autonomous vehicle. IEEE International Conference in Robotics and Automation, MN, USA, 4, 3117-3122.

PCL, 2015. Conditional Euclidean Clustering. Retrieved from PCL: http://pointclouds.org/documentation/tutorials/conditional_euclidean_clustering.php#conditional-euclidean-clustering

PCL, 2015. Removing Outliers Using a StatisticalOutlierRemoval Filter. Retrieved from PCL: http://pointclouds.org/documentation/tutorials/statistical_outlier.php#statistical-outlier-removal

Poli, D. & Caravaggi, I., 2013. 3D modeling of large urban areas with stereo VHR satellite imagery: lessons learned. Natural Hazards, 68(1), 53-78.

Pu, S. & Vosselman, G., 2007. Extracting windows from terrestrial laser scanning. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36, 12-14.

Pu, S. & Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 64(6), 575-584.

Pu, S., Rutzinger, M., Vosselman, G. & Elberink, S. O., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. ISPRS Journal of Photogrammetry and Remote Sensing, 66(6), 28-39.

Puente, I., González-Jorge, H., Martínez-Sánchez, J. & Arias, P., 2013. Review of mobile mapping and surveying technologies. Measurement, 46(7), 2127-2145.

RIEGL, 2014. Data Processing Software, RiPROCESS for RIEGL Scan Data. Retrieved from RIEGL: http://www.riegl.com/uploads/tx_pxpriegldownloads/11_Datasheet_RiProcess_2014-09-18.pdf

RIEGL, 2015. Compact Mobile Laser Scanning System, RIEGL VMX-450. Retrieved from RIEGL Laser Measurement Systemts: http://www.riegl.com/uploads/tx_pxpriegldownloads/DataSheet_VMX-450_2015-03-19.pdf

Rusu, R. B. & Cousins, S., 2011. 3D is here: Point Cloud Library (PCL). IEEE International Conference in Robotics and Automation, Shanghai, China, 1-4.

Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M. & Beetz, M., 2008. Towards 3D point cloud based object maps for household environments. Robotics and Autonomous Systems, 56(11), 927-941.

Rutzinger, M., Pratihast, A. K., Oude Elberink, S. J. & Vosselman, G., 2011. Tree modeling from mobile laser scanning datasets. The Photogrammetric Record, 26(135), 361-372.

Rybka, R., 2011. Autodesk and Bentley Systems talk about mobile LiDAR. LiDAR Magazine, 2. Retrieved from Spatial Media: http://www.lidarmag.com/PDF/LiDAR_Magazine_Vol1No2_Rybka.pdf

Sahin, C., Alkis, A., Ergun, B., Kulur, S., Batuk, F. & Kilic, A., 2012. Producing 3D city model with the combined photogrammetric and laser scanner data in the example of Taksim Cumhuriyet Square. Optics and Lasers in Engineering, 50(12), 1844-1853.

Schrock, G., 2013. Accessible Mobile Mapping. Professional Surveyor Magazine.

SimStadt., 2016. Workflow. Retrieved from SimStadt: http://www.simstadt.eu/en/3d_model.html

Steder, B., Rusu, R. B., Konolige, K. & Burgard, W., 2011. Point feature extraction on 3D range scans taking into account object boundaries. IEEE International Conference in Robotics and Automation, Shanghai, China, 2601-2608.

Valente, J. & Soatto, S., 2015. Perspective distortion modeling, learning and compensation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, MA, USA, 9-16.

Wang, R., 2013. 3D building modeling using images and lidar: a review. International Journal of Image and Data Fusion, 4(4), 273-292.

Wang, R., Bach, J. & Ferrie, F. P., 2011. Window detection from mobile LiDAR data. IEEE Workshop in Applications of Computer Vision, Hawaii, USA, 58-65.

Williams, K., Olsen, M. J., Roe, G. V. & Glennie, C., 2013. Synthesis of transportation applications of mobile LiDAR. Remote Sensing, 5(9), 4652-4692.

Wolcott, R. W. & Eustice, R. M., 2014. Visual localization within LiDAR maps for automated urban driving. IEEE International Conference in Intelligent Robots and Systems, Chicago, USA, 176-183.

Yang, B., Fang, L., Li, Q. & Li, J., 2012. Automated extraction of road markings from mobile LiDAR point clouds. Photogrammetric Engineering & Remote Sensing, 78(4), 331-338.

Yang, B., Wei, Z., Li, Q. & Li, J., 2012. Automated extraction of street-scene objects from mobile LiDAR point clouds. International Journal of Remote Sensing, 33(18), 5839-5861.

Yang, B., Wei, Z., Li, Q. & Li, J., 2013. Semiautomated building facade footprint extraction from mobile LiDAR point clouds. IEEE Geoscience and Remote Sensing Letters, 10(4), 766-770.

Yen, K. S., Ravani, B. & Lasky, T. A., 2011. LiDAR for Data Efficiency. No. WA-RD.

Yokoyama, H., Date, H., Kanai, S. & Takeda, H., 2011. Pole-like objects recognition from mobile laser scanning data using smoothing and principal component analysis. ISPRS Laser Scanning Workshop, Calgary, Canada, 38, 115-120.

Yu, Y., Li, J., Guan, H., Wang, C. & Yu, J., 2015. Semi-automated extraction of street light poles from mobile LiDAR point-clouds. IEEE Transactions on Geoscience and Remote Sensing, 53(3), 1374 -1386.

Zhang, H., 2016. Rapid inspection of pavement markings using mobile laser scanning point clouds, MSc Thesis, Department of Geography and Environmental Management, University of Waterloo.

Zhang, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision, 13(2), 119-152.

Zhou, L. & Vosselman, G., 2012. Mapping curbstones in airborne and mobile laser scanning data. International Journal of Applied Earth Observation and Geoinformation, 18, 293-304.

Zhu, L. & Hyyppa, J., 2014. The use of airborne and mobile laser scanning for modeling railway environments in 3D. Remote Sensing, 6(4), 3075-3100.

Zhu, L., Hyyppä, J., Kukko, A., Kaartinen, H. & Chen, R., 2011. Photorealistic building reconstruction from mobile laser scanning data. Remote Sensing, 3(7), 1406-1426.