Semi-automated Generation of Road Transition Lines Using Mobile Laser Scanning Data

by

Han Jiang

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, 2017

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

Recent advances in autonomous vehicles (AVs) are exponential. Prominent car manufacturers, academic institutions, and corresponding governmental departments around the world are taking active roles in the AV industry. Although the attempts to integrate AV technology into smart roads and smart cities have been in the works for more than half a century, the High Definition Road Maps (HDRMs) that assists full self-driving autonomous vehicles did not yet exist. Mobile Laser Scanning (MLS) has enormous potential in the construction of HDRMs due to its flexibility in collecting wide coverage of street scenes and 3D information on scanned targets. However, without proper and efficient execution, it is difficult to generate HDRMs from MLS point clouds.

This study recognizes the research gaps and difficulties in generating transition lines (the paths that pass through a road intersection) in road intersections from MLS point clouds. The proposed method contains three modules: road surface detection, lane marking extraction, and transition line generation. Firstly, the points covering road surface are extracted using the voxel-based upward-growing and the improved region growing. Then, lane markings are extracted and identified according to the multi-thresholding and the geometric filtering. Finally, transition lines are generated through a combination of the lane node structure generation algorithm and the cubic Catmull-Rom spline algorithm.

The experimental results demonstrate that transition lines can be successfully generated for both T- and cross-intersections with promising accuracy. In the validation of lane marking extraction using the manually interpreted lane marking points, the method can achieve 90.80% precision, 92.07% recall, and 91.43% F_1-score, respectively. The success rate of transition line generation is 96.5%. Furthermore, the Buffer-overlay-statistics (BOS) method validates that the proposed method can generate lane centerlines and transition lines within 20 cm-level localization accuracy from MLS point clouds. In addition, a comparative study is conducted to indicate the better performance of the proposed road marking extraction method than that of three other existing methods. In conclusion, this study makes a considerable contribution to the research on generating transition lines for HDRMs, which further contributes to the research of AVs.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Dr. Jonathan Li, for providing me professional research backgrounds and continuous supports throughout my master program.

I would like to thank my Master's Thesis Examining Committee, which includes Prof. Dr. Michael Chapman at the Department of Civil Engineering, Ryerson University, Prof. Dr. Richard Kelly, at the Department of Geography and Environmental Management, University of Waterloo, and Prof. Dr. Alexander Wong at the Department of Systems Design Engineering, University of Waterloo for reviewing my thesis, attending my defense, and providing me the valuable comments.

Special thanks go to Fujian Key Laboratory of Sensing and Computing for Smart Cities, Xiamen University for providing the RIEGL VMX-450 point cloud datasets used in this thesis.

Sincere thanks to my colleagues working in Mobile Sensing and Geodata Analytics Lab, University of Waterloo, especially to Menglan Zhou, Weikai Tan, Lanying Wang, Haocheng Zhang, Lingfei Ma, and He Zhao for their academic assistances.

I am very thankful to my parents for their love and support. They not only financially supported my studies, but also encouraged me to get through all this work and stress.

Table of Contents

List of Abbreviations

2D	Two-dimensional
3D	Three-dimensional
ALS	Airborne Laser Scanning
AV	Autonomous Vehicle
BFS	Breadth-first Searching
BOS	Buffer-overlay-statistics
DEM	Digital Elevation Model
DMI	Distance Measurement Indicator
GIS	Geographic Information System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GRF	Geo-referenced Feature
HDRM	High-definition Road Map
HL-3DRNM	Hierarchical Lane-oriented 3D Road-network Model
IDW	Inverse Distance Weighted
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging
MLS	Mobile Laser Scanning
PCA	Principal Component Analysis
PCL	Point Cloud Library
POS	Position and Orientation System
RADAR	Radio Detection and Ranging

TLS	Terrestrial Laser Scanning
ToF	Time-of-Flight
WGS84	World Geodetic System 1984
UAV	Unmanned Aerial Vehicle
UTM	Universal Transverse Mercator Projection

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

1.1.1 Requirements for High-definition Road Maps

The development of Autonomous Vehicles (AVs) starts in the 1980s, when Carnegie Mellon University introduced the Navlab vehicles that operated autonomously in a variety of road environments (Thorpe et al., 1991). AVs promise numerous improvements to vehicular transportation, including an increase in traffic capacity, less fuel consumption, and fewer accidents (Luettel et al., 2012). A range of technology companies, automotive manufacturers and suppliers, academic organizations, and government bodies are leading various technological efforts to make this technology more responsive, more accessible and ultimately safe (Abraham et al., 2017). For example, Ontario Good Roads Association published a white paper, which states that Ontario will develop an ecosystem across the province to help Ontario win the presently AV race (Shuja, 2015).

The range of vehicular automation can be described by the Levels of Vehicle Automation, which is put forward by the U.S. Department of Transportation's National Highway Traffic Safety Administration. The agency separates vehicle automation into five levels: no automation (Level 0), function-specific automation (Level 1), combined function automation (Level 2), limited self-driving automation (Level 3), and full self-driving automation (Level 4) (National Highway Traffic Safety Administration, 2013). Vehicles at level 4 are designed to monitor roadway conditions and operate all driving functions for an entire trip. To help full self-driving automation, extra platforms including motion planning systems, perception systems, mission planners, and behavioral systems, should be set on board (Fagnant & Kockelman, 2015).

The platforms control the behaviors of an autonomous vehicle based on two kinds of data: real-time perception results from onboard sensors and pre-loaded navigation maps (Schreiber et al., 2013). The onboard sensors, such as optical cameras, Radio Detection and Ranging (RADAR) and Light Detection and Ranging (LiDAR) systems, are essential parts to support the autonomous vehicles driving in road environments. Features around the vehicles are extracted out of the measured raw data with a goal of finding obstacles and determining motions. However, urban scenes could be too complex to be completely detected by the sensors. In such cases, road maps, which are used to generate routing trajectories for autonomous driving in well-structured environments, can support real-time motion planning if detailed road information is precomputed (Buehler et al., 2009).

Finding the best path from starting position to a destination is an important function of road maps. Common road maps, such as commercial navigation maps, usually represent roads as connected lines with additional attributes. This information might be helpful enough for a driving assistant system for manual driving, as it solves the question of which road should be taken. However, to serve as an assistant system for autonomous driving, there is a further question that should be answered by the maps: which lane should be taken? Note this question, much detailed road information should be computed and added to the conventional road maps (Gwon et al., 2017). Structured features on roads, including lanes, road markings, and traffic signs, are basic components of an HDRM and important signals for the behavioral system of autonomous vehicles. Simulated features such as lane centerlines and transition lines, which autonomous vehicles move on, play an essential role in trajectory planning (de Lima & Victorino, 2016). HDRMs will provide great supporting information for the onboard sensors especially when lanes or obstacles are not perceptible.

1.1.2 MLS Compared with Other Techniques

A classical road map for navigation is created from a Geographic Information System (GIS) through information filtering, organizing, and visualizing. The road information stored in a GIS database is mostly gathered by surveying and remote sensing techniques. Aerial photogrammetry, one remote sensing data source, is commonly used for road and road feature extraction. The method starts with extracting features from edges or regions on the images; road areas can then be extracted based on their features (Yin et al., 2015). With the extracted road areas, road centerlines can be generated through pattern recognition (Shi et al., 2014). These methods work very well along the main road network in study areas. However, simultaneously, they can be seriously impacted by blocking features along the road, such as high-rise buildings and trees. Moreover, when taking a further step toward road surface information extraction or branch road extraction, the small disconnects in picturing road details caused by blocking objects and missing elevation information makes the reconstruction of roads difficult (Wegner et al., 2013). In summary, due to the high complexity and diversity of road environments, the generation of HDRMs from aerial images is problematic.

Compared with aerial images, laser scanning, as a continuously promoted remote sensing technique, can provide quick acquisition of high-quality 3D information of road scenes when integrated with positioning technology (Yu et al., 2015b). Currently, laser scanning data, or 3D point clouds, can be acquired through Airborne Laser Scanning (ALS) platforms, Mobile Laser Scanning (MLS) platforms, and Terrestrial Laser Scanning (TLS) platforms (Hyyppä et al., 2013). Point clouds recorded by the three types of laser scanning platforms contain not only factual geographic coordinate information but also the reflective intensity of feature surfaces. Laser scanning technology uses non-contact measurement to directly access scanned surfaces, depending little on illumination. Although ALS, MLS, and TLS systems record similar properties, features of targets are represented distinctly in the point clouds from different platforms. A proper type of point cloud should be chosen for particular tasks, based on the scale of targets and desired point density.

Choosing an appropriate type of point clouds for information extraction relies on the features of interest in the road environments. On the one hand, the point clouds of a road network should be easily acquired with the help of a few technicians. On the other hand, the resolution of the point clouds should be high enough to distinguish road surface features, including road curbs, road markings, and road signs. ALS platforms can undoubtedly cover a great scale of a city within a short time, whereas completing such a task with TLS platforms would be hard (Zhu & Hyyppa, 2014). MLS platforms are useful for providing point clouds for roads within a small city, but remaining a time-consuming approach for the oversized cities. In terms of point density, TLS and MLS point clouds commonly have more than 10,000 pts/m^2, which provides greater than 1 cm in resolution, while it is hard for ALS point clouds to reach that precision (Wang, 2016). As a result, road features such as lane markings are only distinguishable on TLS and MLS point clouds. Due to its higher flexibility and acquisition rate in large-scaled road scenes, MLS point clouds is more applicable than ALS and TLS point clouds for transition line generation in this study.

1.1.3 Challenges of Road Transition Line Generation Using MLS Point Clouds

Transition lines are driving paths that pass through road intersections (Guo et al., 2016). Note that transition lines are not real lines that are marked on the road surface, the generation of them should take the lane markings, which represents the edges of lanes, and traffic rules as essential references (Yang et al., 2013). Therefore, to generate transition lines, we first need to extract road markings, especially the lane markings, from MLS point clouds. However, the complex patterns of road markings in road intersections make the acquisition of road markings challenging. Outliers, such as road side building facades, road pavements, and other points on the road surface, should also be distinguished and removed (Yang et al., 2012). Another challenge comes from conducting human knowledge (traffic rules) in lane centerline and transition line generation. Two directions of a road and forks of a road intersection should be classified, and then all the possible paths that vehicles may take to pass a road intersection should be determined following traffic rules (Liu et al., 2013a). need to be determined following traffic rules. In addition, any generated transition lines must be carefully confirmed to ensure their quality and to provide reference information for HDRMs.

In addition to the complex patterns of road environments, challenges also arise from MLS point clouds. Such point clouds contain a mass of dense points with 3D geographic information, reflection information, return information, and so on. Organising these features would be a key pre-condition in developing efficient algorithms. First, the spatial discontinuity of points makes searching a target point or its nearby points time-consuming if points are not well organized in an index data structure (Yang et al., 2013). Second, intensity values become degraded when laser shooting points get far away from the sensor even though the features of the target surface keep the same, which makes distance an unavoidable variable when using intensity values (Cheng et al., 2017). Third, it is inefficient to use the whole MLS point cloud for information extraction. Key features and key points should be detected and extracted by a method so that lane centerlines and transition lines can be precisely generated. In summary, challenges for simulating turning lines mainly come from the environmental complexity of roads and the features of MLS point clouds. This study thus focuses on proposing a solution for these challenges.

1.2 Objectives of the Thesis

The goal of this study is to develop a semi-automated method for the generation of transition lines in typical urban road intersections. The specific objectives of this thesis are as follows:

- To develop an improved computerized algorithm for road surface extraction from MLS point clouds of urban road intersections with road curbs.
- To develop a novel semi-automated algorithm for node structure generation from clustered lane marking, which represents the edge of a lane, point clouds in road intersections.
- To implement algorithms for ground point detection, road marking detection and transition line generation in T-intersections and cross-intersections.
- To conduct a method for the validation of detected lane markings and transition lines and a comparative study for the evaluation of the road marking extraction method.

1.3 Structure of the Thesis

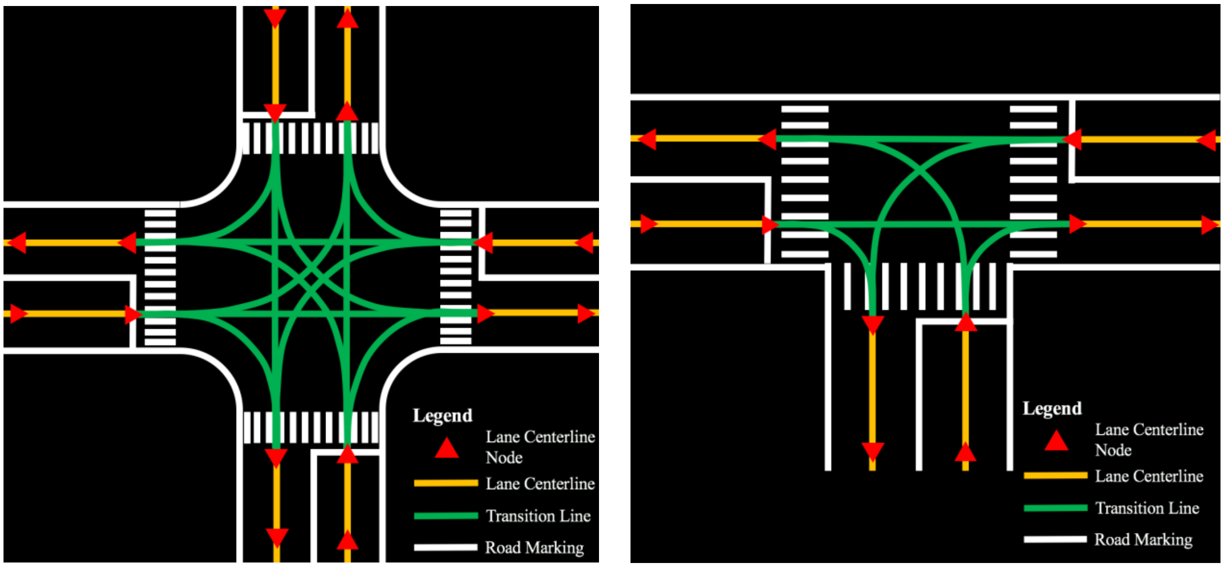
This thesis is extended from the successive four chapters. Chapter 2 introduces MLS, HDRMs, and the related work in road surface detection, road marking extraction and transition line generation. Chapter 3 describes the characteristics of the test datasets and elaborates the stepwise data processing methodology, including road surface extraction, lane marking extraction, and transition line generation. Moreover, the validation method is introduced. Chapter 4 presents and discusses the experimental results and the validation results in this study. Chapter 5 draws the conclusion with findings, summarises its contributions and limitations, and makes suggestions for future studies.

Chapter 2 Background and Related Work

This chapter reviews the literature related to the generation of transition lines at road intersections from MLS point cloud data. Section 2.1 presents an introduction to HDRMs. Section 2.2 introduces MLS technology. Section 2.3 documents recent studies of road surface detection, road marking extraction, and transition line generation. Section 2.4 summarises this chapter by figuring out the gaps in the previous studies. This chapter provides a theoretical background for the proposed method in this study.

2.1 Introduction to HDRMs

HDRMs describe the road geometry as linked lane segments. They characterize roads, on the one hand, with more recall and, on the other hand, with higher accuracy than standard digital maps to fulfill the requirements of driving assistance systems (Bétaille & Toledo-Moreo, 2010). The road network information stored in an HDRM database includes both road-level and lane-level information. The lane-level information contains lane geometry, lane attributes, and road intersection geometry. Each set of the information in HDRMs has multiple potentialities for vehicular applications (Hillel et al., 2014). Lane geometry can be utilized for curve speed control (Jiang et al., 2010) and lane change aiding (Gao & Aghajan, 2009). Lane attributes, including lane width, elevation, grade, edge coordinates, and bank angle, can be used for lane keep assistance (Wu et al., 2008) as well as lane departure warning (Danescu & Nedevschi, 2009). Road intersection geometry can add additional safety benefit to navigation systems (Du, 2004).



(a) Cross-intersection (b) T-intersection

Figure 2-1 Definition of transition lines

Figure 2-1 clarifies the definition of transition lines in a cross-intersection (Figure 2-1 (a)) and a T-intersection (Figure 2-1 (b)). Three constraints are involved in the generation of transition lines for an HDRM database. First, each road is decomposed into a sequence of analytic lane centerline curves, which represent the centerlines of road lanes. Second, lane centerlines are directed line segments that have start and end nodes. Third, lane centerlines are connected by transition lines that are represented as curved or straight line segments. The acceptable vehicle positioning errors for lane-level navigation are based on a total error that combines the expected error of the road map database with vehicle positioning error. For three-meter-wide lanes, the lane-level accuracy is specified to be 30 cm in the road map database, with less than 20 cm vehicle positioning error (Du, 2004). To help the construction of HDRMs, the main discussion of this thesis will focus on the generation of transition lines in road intersections.

Transition lines in the HDRM database approximate real vehicle turning paths and describe the geometric details of a road intersection. They present all the possible paths that autonomous vehicles may take to pass the road intersection. Moreover, the connectivity description provides a coordinate description to the inner attributes of the road intersection. Therefore, the vehicle positioning results from onboard sensors can be matched and constrained more continuously and precisely (Liu et al., 2013a). Transition lines can support smooth switch among a road intersection and provide more benefits to map-based vehicle positioning solutions (Zhang et al., 2011).

2.2 Introduction to MLS

MLS is a surveying technique that uses laser scanners mounted on a mobile platform to obtain surface information of objects (Talaya et al., 2004; El-Sheimy, 2005). By constantly emitting and receiving near-infrared laser beams, MLS systems can collect 3D coordinates of scanned surfaces (Haala et al., 2008; Marshall, 2011). Additional information including intensity properties, return number and number of returns are simultaneously recorded when laser energy returns (Lemmens, 2011). Compared with other surveying methods, such as surveying with GPS, total stations, and photogrammetry, MLS promises many advantages: rapid data acquisition, high point density, high accuracy, and flexible working time (Yen et al., 2011, Zhu et al., 2011).

2.2.1 Components of MLS Systems

As a sub-category of the mobile mapping system, MLS systems utilize motion tracking and time synchronized data acquisition. Generally, an MLS system has five essential components: laser scanners, optical cameras, an Inertial Measurement Unit (IMU), a Distance Measurement Indicator (DMI), a Global Navigation Satellite System (GNSS), and a control system for sensor synchronization (Talaya et al., 2004). Among these components, the laser scanners and the optical cameras are responsible for data collecting. The GNSS, the DMI, and the IMU compose a Position and Orientation System (POS).

(1) Laser scanner

Laser scanners constantly emit laser pulses at fixed angular increments and collect the backscattered energy for range measurement (Glennie, 2007). In different MLS systems, the type and the number of laser scanners might vary. Fixed scanning head laser scanners set the laser scan heads to a fixed angle, and only internal mirror movement takes place in measurements. To collect a 360-degree range of points, multiple laser scanners are required. Rotating scanning head laser scanners have rotating scan heads that can collect points in a full 360-degree planar sweep. In both cases, the scanning plane of the sensors coupled with the movement of the vehicle enables the system to collect point clouds in a wide window. The scan data collected through laser scanners consist of angles, intensities, and ranges with time stamps.

The range to objects can be calculated mainly based on two techniques: time-of-flight and phase shift (Lichti, 2010). A laser scanner using time-of-flight technique launches a short laser pulse to a target, and then the range to the target (R) can be determined by the time difference between the emitted and received laser energy through the following equation (Lichti, 2010):

$$R = \frac{c\Delta t}{2}, \tag{2-1}$$

where Δt is the time interval between emitting and receiving of the laser pulse and c is the speed of light. In contrast, for phase based laser scanners, the range to the target is computed using the phase difference between the emitted and received laser energy of an amplitude modulated continuous wave. The equation used for calculating the range from phase shift is (Lichti, 2010):

$$R = \frac{\Delta\varphi}{2\pi}\frac{\lambda}{2} + \frac{\lambda}{2}n, \tag{2-2}$$

where $\Delta\varphi$ is the phase shift, λ is the modulation wavelength and n is the number of transmitted waves. Compared with time-of-flight scanners, phase-based scanners can achieve a higher accuracy and point density, while their scanning range is shorter (Petrie and Toth, 2008).

(2) Digital camera

Digital cameras are always incorporated into MLS systems as a supporting system for visualization and referencing purposes. To aid in visualization, they individually take optical images of targets in a fixed time interval as a video stream. These optical images can be superimposed upon the point clouds to add detailed color and texture information to them (Zhou & Vosselman, 2012). This additional information might be helpful to advanced feature extraction such as traffic sign detection (Wen et al., 2016; Soilán et al., 2016). To aid in referencing, geo-referenced images can be mapped to point clouds, which enables users to interact with images rather than oversized point cloud data. Further, ortho-images can be generated for validation purposes. Different arrangements of digital cameras could be applied to the MLS systems based on the scanning angle of laser scanners so that they can assist interpretation of point clouds.

However, when we use the digital images as ancillary datasets, two features of them should be well considered. Firstly, note that photogrammetry is a passive remote sensing technique, the quality of images acquired by digital cameras varies depending on light conditions for the study areas and focus of the cameras. Secondly, perspective distortion, which refers to a complex deformation of the image because of the combined physical effects of distance and focal length, is indispensable in MLS systems due to the movement of vehicles (Valente & Soatto, 2015). When distorted images are mapped to point clouds, they might provide false information for the subsequent processing.

(3) Position and Orientation System (POS)

To make a continuous scanning, vehicles should keep moving, which makes the positions of laser scanners various in data acquisition. Hence, position, velocity, and time information are essential for the geo-referencing of point clouds (Glennie, 2007). This information is provided by the POS that is composed of a GNSS, a DMI, and an IMU. The integration of these sensors allows exploiting the complementary nature of them.

GNSS mainly provides three measurements: position, velocity, and time. Although GNSS can produce accurate positioning information in road environments, the signal quality might be affected by the blocking of high-rise buildings, roadside trees, or other obstacles. Furthermore, IMU can afford attitude information (heading, roll, and pitch) of the vehicle, and it can sense rotation angles and acceleration from three axes when satellite signals are not available. However, the accuracy of orientation and position information decrease in IMU with the accumulation of measuring time (Puente et al., 2013). Therefore, in MLS systems, the GNSS and IMU are integrated to ensure the orienting and positioning accuracy. IMU can enhance the positioning accuracy of GNSS when satellite signals are weak, while GNSS can update positioning information in IMU.

DMI can provide additional positioning information for IMU and GNSS. It is fixed on one wheel of the vehicle to directly estimates traveling distance by measuring wheel rotation. When GNSS is under intermittent satellite signal coverage, rotation information can provide constraints to time drift in IMU.

(4) Control system

A control system can synchronize laser scanners, digital cameras, and the position and orientation system. It stores the data acquired by laser scanners and digital cameras and integrates signals collected by the position and orientation system.

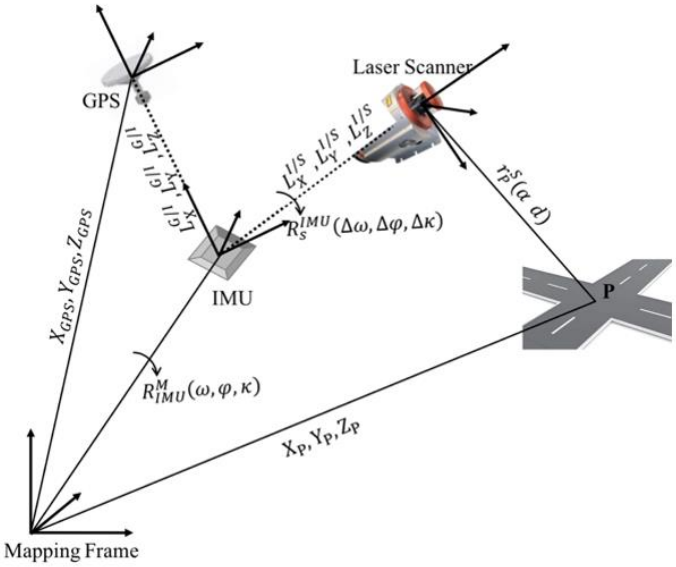
2.2.2 Direct Geo-referencing and Scanning Parameters

In an MLS system, when the surveying vehicle is moving, GNSS constantly measures the real-time geodetic coordinates of GNSS antenna phase center, and IMU records instant attitude angles including roll, pitch, and yaw. The position and orientation data are precisely time stamped to integrate sensors and determine the coordinates of scanned objects in the mapping frame. Laser scanners measure the distances and scanning angles from the scanned target to the scanning center. Digital cameras acquire the texture information of surrounding objects. As shown in Figure 2-2, laser scanners are referenced using their orientations and positions relative to the mapping frame and the positional relationship among IMU, GNSS and laser scanners. Then, the coordinates of scanned targets in the mapping frame can be computed through the distances and the scanning angles.

The coordinates of scanned point P are calculated by the following equations (Glennie, 2007; Barber et al., 2008):

$$\begin{bmatrix} X_{\mathrm{P}} \\ Y_{\mathrm{P}} \\ Z_{\mathrm{P}} \end{bmatrix}^{\mathrm{M}} = R_{IMU}^{M}(\omega,\varphi,\kappa)\cdot\left(R_{S}^{IMU}(\Delta\omega,\Delta\varphi,\Delta\kappa)\cdot r_{P}^{S}(\alpha\ \ d) + \begin{bmatrix} L_X^{I/S} \\ L_Y^{I/S} \\ L_Z^{I/S} \end{bmatrix}_{s}^{IMU} - \begin{bmatrix} L_X^{G/I} \\ L_Y^{G/I} \\ L_Z^{G/I} \end{bmatrix}_{IMU}^{GPS} \right) + \begin{bmatrix} X_{GPS} \\ Y_{GPS} \\ Z_{GPS} \end{bmatrix}^{M} \tag{2-3}$$

where the descriptions of the parameters in Equation 2-3 are shown in Table 2-1.



GPS
Laser Scanner
$L_X^{G/I}, L_Y^{G/I}, L_Z^{G/I}$
$L_X^{I/S}, L_Y^{I/S}, L_Z^{I/S}$
$R_S^{IMU}(\Delta\omega, \Delta\varphi, \Delta\kappa)$
$r_P^S(\alpha\ d)$
IMU
P
$X_{GPS}, Y_{GPS}, Z_{GPS}$
$R_{IMU}^M(\omega, \varphi, \kappa)$
X_P, Y_P, Z_P
Mapping Frame

Figure 2-2 Illustration of direct geo-referencing

Table 2-1 Descriptions of parameters in the direct geo-referencing equation

Parameters	**Description**
X_P, Y_P, Z_P	Location of scanned point P in the mapping frame.
$R_{IMU}^M(\omega, \varphi, \kappa)$	Rotation matrix between the mapping frame and the IMU coordinate system. $\omega, \varphi,$ and κ are the roll, pitch, and yaw of IMU with respect to the mapping frame. These parameters are defined by the system calibration.
$R_S^{IMU}(\Delta\omega, \Delta\varphi, \Delta\kappa)$	Rotation matrix between IMU and the laser scanner. $\Delta\omega, \Delta\varphi,$ and $\Delta\kappa$ are boresight angles that align the frame of the laser scanner with IMU body frame. These parameters are provided by the system calibration.
$r_P^S(\alpha\ \ d)$	The relative position of P in the laser scanner coordinate system. $r_P^S(\alpha\ \ d) = d \cdot \begin{bmatrix} \cos\alpha \\ 0 \\ \sin\alpha \end{bmatrix}$, where d and α are range and scanning angle measured by the laser scanner.
$L_X^{I/S}, L_Y^{I/S}, L_Z^{I/S}$	Lever-arm offsets from the origin of IMU to the origin of the laser scanner coordinate system.
$L_X^{G/I}, L_Y^{G/I}, L_Z^{G/I}$	Lever-arm offsets from the origin of IMU to the origin of GNSS antenna.
$X_{GPS}, Y_{GPS}, Z_{GPS}$	Location of GNSS phase center in the mapping frame.

2.2.3 RIEGL VMX-450 System

This thesis utilizes the data collected by a RIEGL VMX-450 MLS system from Xiamen University. The RIEGL VMX-450 MLS system consists of two RIEGL VQ-450 laser scanners, four RIEGL VMX-450-CS6 optical cameras, and an Applanix POS LV 520 processing system. The components are integrated by a POS computer system and mounted onto a vehicle (Figure 2-3). The position and navigation solution is provided by two GNSS antennas, a DMI, and an IMU in Applanix POS LV 520 system. RIEGL VQ-450 laser scanners are rotating head laser scanners which have a 360° field of view and produce 'X' shaped scan lines. The coordinate system and the plane projection system of the point clouds can be transformed to WGS84 and UTM in the data processing software RiPROCESS. Additionally, vehicle trajectory data are obtained and provide as ancillary data. A vehicle trajectory is a path that the vehicle moves along a road as a function of time (Guan et al., 2014). While the flight-of-time (ToF) MLS measurements were collected, the vehicle trajectory was recorded by the Position and Orientation System (POS). The raw GNSS data and IMU data were processed in Applanix POSPac^{TM} MMS software, and the best estimation of the vehicle trajectory (hereafter referred to as trajectory points) is produced. The vehicle trajectory is essential for the direct geo-referencing of point clouds.

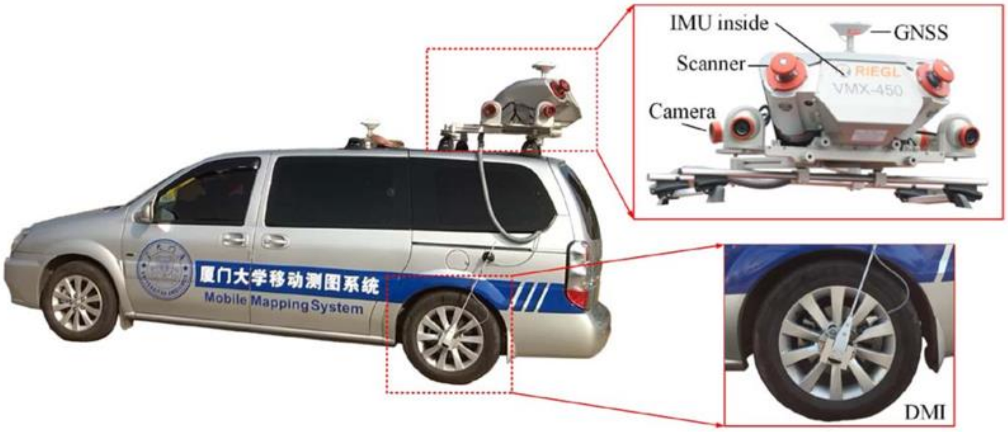


Figure 2-3 A RIEGL VMX-450 system (Yu, 2015c)

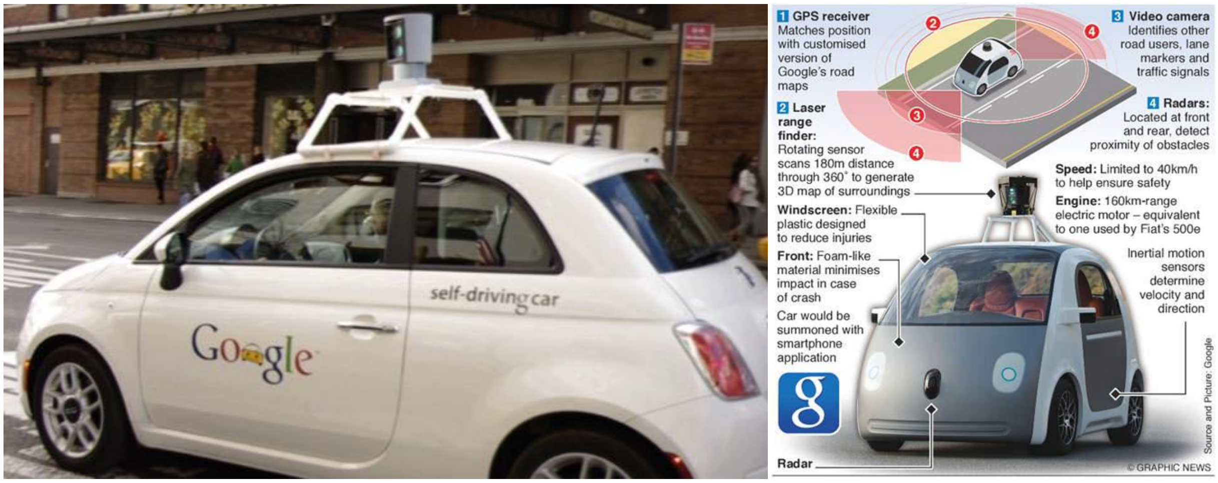

self-driving car
Google
1 GPS receiver Matches position with customised version of Google's road maps
2 Laser range finder: Rotating sensor scans 180m distance through 360° to generate 3D map of surroundings
3 Video camera Identifies other road users, lane markers and traffic signals
4 Radars: Located at front and rear, detect proximity of obstacles
2
4
3
4
Speed: Limited to 40km/h to help ensure safety
Engine: 160km-range electric motor – equivalent to one used by Fiat's 500e
Windscreen: Flexible plastic designed to reduce injuries
Front: Foam-like material minimises impact in case of crash
Car would be summoned with smartphone application
Inertial motion sensors determine velocity and direction
Radar
Source and Picture: Google
© GRAPHIC NEWS

Figure 2-4 A Google autonomous vehicle (Google, 2017)

The type and the function of laser scanners for autonomous vehicle navigation and HDRM construction are different. For example, on a Google's autonomous vehicle, a Velodyne HDL-64E laser scanner is configured on the top of the vehicle (Figure 2-4). Point clouds collected by the laser scanner are used for the real-time navigation of the vehicle. However, in a RIEGL VMX-450 system, two RIEGL VQ-450 laser scanners are symmetrically mounted on the right and left sides, pointing to the rear of the vehicle. Point clouds collected by the two laser scanners are commonly used for post-processing.

Table 2-2 shows the configuration and the measurement performance of a Velodyne HDL-64E laser scanner on a Google autonomous vehicle and a RIEGL VQ-450 laser scanner in a RIEGL VMX-450 system. Both the two laser scanners use TOF measuring technology and provide 360° coverage. A RIEGL VQ-450 laser scanner can generate up to 550 thousand pts/s, while a Velodyne HDL-64E laser scanner can generate 2.2 million pts/s, which is better for real-time computation. However, the accuracy of a RIEGL VQ-450 laser scanner can reach 8 mm in measurement accuracy and 0.001° in angle measurement resolution, which is much higher a Velodyne HDL-64E laser scanner's 2 cm in measurement accuracy and 0.08° in measurement resolution. Therefore, a RIEGL VQ-450 laser scanner can generate much more accurate point clouds than a Velodyne HDL-64E laser scanner. Since the construction of an HDRM is a one-time mission, point clouds collected by the RIEGL WMX-450 system are more suitable for that mission. When the base HDRM is created, it could be updated by AVs.

Table 2-2 Comparison of RIEGL VQ-450 and Velodyne HDL-64E

Laser scanner	RIEGL VQ-450	Velodyne HDL-64E
Configuration		
Measuring principle	TOF measurement	TOF measurement
Maximum range	Up to 800 m	120 m
Minimum range	1.5 m	1.5 m
Accuracy/precision	8 mm/5 mm (1 σ)	2 cm
Field of view	360°	360°
Angle measurement resolution	0.001°	0.09°
Effective measurement rate	Up to 550 kHz @ 200m	Up to 2.2 million points per sec
Laser wavelength	Near infrared	Near infrared
Laser beam divergence	0.3 mrad	-
Laser beam footprint	7 mm @ exit aperture 17 mm @ 50 m 32 mm @ 100 m	-
Reference	(RIEGL, 2017)	(Glennie & Lichti, 2010)

2.3 Techniques for Generating Road Transition Lines

2.3.1 Curb-based Road Surface Detection Using MLS Point Clouds

Several methods have been developed to detect road surfaces from MLS point clouds, and these algorithms can be categorized into three types based on data structure: scan line, road patch, and voxel.

Scan lines are commonly generated using the scanning angle field or the Global Positioning System (GPS) time stamp field in MLS point clouds if points are time-wise sorted. With a few points in a single scan line, computation can be conducted efficiently. McElhinney et al. (2010) detected road curbs (the boundaries of roads) to extract road surface points from scan lines. Elevation jump and slope of two adjacent points were computed and labeled as road curb points if both were greater than the thresholds. Yang et al. (2013) detected road curbs from a scan line through constructing a moving window operator. The moving window operator was used to detect curb points based on pre-defined curb patterns. Instead of searching for road boundaries, some methods directly extract road surface points from a scan line. Yan et al. (2016) utilized elevation jump between trajectory points and road surfaces to extract seed points, then road surface points were detected by moving least squares line fitting. Similarly, Riveiro et al. (2015) performed Principal Component Analysis (PCA) to the local neighborhood of seed points for road surface detection.

When conducting the scan line based methods in ideal street scenes, a succession of parallel U-shaped polylines should be formed after the segmentation. However, obstacles such as parked cars, passengers, and pavement grass could break the ideal arrangement. Cabo et al. (2016) split and simplified scan lines using an improved version of the Douglas-Peucker algorithm. Lines that belonged to the same planar were regrouped based on the proximity and the similarity to reduce the influence of obstacles. Although dividing a point cloud into scan lines makes complex computation and algorithm applicable, information contained in a scan line is still not enough to detect road curbs in complex road environments (Cabo et al., 2016). Furthermore, when MLS point clouds are not sorted by the time stamp field, or they are mosaicked from several datasets, scan line based algorithms need extra time for sorting of point clouds.

Road patches can be generated by segmenting MLS point clouds into blocks with the assistance of trajectory data at a specified time or distance interval. Compared with scan lines, road patches not only have high computational efficiency but also contain more geometric information on road surfaces. Many road surface detection methods are based on road patches. Guan et al. (2014) and Yu et al. (2015b) transversely sectioned road patches into corresponding profiles and then gridded the profiles to form a pseudo scan-line with a given width. Next, principal points within a grid were extracted based on elevation jump. Points in a grid were labeled as outliers if the elevation jump was greater than a threshold. In these studies, road curbs in each profile were detected based on slope and elevation jump between two consecutive points. Wang et al. (2015) and Soilán et al. (2017) measured the saliency of points within each patch and then clustered salient points using region growing algorithm to compute the characteristics of road scenes and detect road curbs. Although road patch based methods provide promising results in road surface detection, the accuracy is influenced by the trajectory data. In T-intersections, the curved trajectories might cause false segmentation of MLS point clouds.

Voxels are cuboid-shaped space with a given length, width, and height. Point clouds can be gridded using voxels. Some experts have tried to detect road surfaces and road curbs in voxels. Liu et al. (2013b) constructed a dynamic Digital Elevation Model (DEM) with the average height of points in voxels to diminish the impact of moving vehicles around the scanners and to capture the complete geometric features of the road surfaces. Candidate road curbs were detected based on the height jump within a single voxel, and slope and height variance of two adjacent voxels. Dynamic DEMs can present complete road curbs and road surfaces; however, they are built from 2D sequential laser range finder data and vehicle state data, which are not available in most MLS point cloud datasets. Luo et al. (2016) conducted multi-category labeling based on the voxelization of MLS point clouds. The semantic labeling model was trained and a library was constructed using the local object information and graph analysis. Voxels were labeled as road surface when their features had correspondences to the graphs in the library. Prior semantic knowledge was essentially needed in this method. Furthermore, Xu et al. (2017) calculated density gradients of adjacent voxels. If gradients of a voxel were large in more than one direction, the voxel was labeled as candidate road curb voxel. Energy function was constructed based on the area of a surface, with a rule that large surface had more energy. Least cost path model was then used for extracting labeled road curb voxels and removing noise. The method achieves promising results; however, it mainly focused on detecting road curbs. Road surface points should be further extracted with the detected curbs. Moreover, the computational time of applying this method could be lengthy when processing high-density and high-volume MLS point clouds in road intersections.

2.3.2 Road Marking Extraction Using MLS Point Clouds

Extracting road marking points from MLS point clouds contains two typical steps: candidate road marking point extraction and post-refinement. The input datasets of most road marking extraction methods are road surface point clouds due to the computing time consideration. In road surface point clouds, road marking points have a higher intensity than nearby road surface points. Thus, intensity thresholding has been commonly used in road marking extraction. The extraction methods can be classified into two categories: geo-referenced feature (GRF) image-based methods and point-based methods. Methods in both categories can extract candidate road marking points from road surface points. However, outliers, such as false positive and negative noise and cracks, are unavoidably selected as candidate road marking points. Post-refinement of candidate road marking points is necessary. According to the features of outliers, they are filtered out through different algorithms.

The generation of GRF images was mainly based on the Inverse Distance Weighting (IDW) interpolation (Guan et al., 2014). To extract road markings, intensity values were set as the gray scale of GRF images relied on two rules: a point with higher intensity had a higher weight and a point close to the central point was assigned a higher weight. Yang et al. (2012) filtered the generated GRF images by computing parameters including area, width, and length of the minimum bounding box of each 4-connected region on GRF images. The Progressive Probabilistic Hough Transform operator was then conducted to extract road marking pixels on the filtered images (Yang et al., 2012). Although road markings are highly reflective objects painted on road surfaces, the reflected laser pulse intensities also depend on the incident angle of the emitted laser beams and the range between measured road markings and laser scanners. Generally, the laser pulse intensity value decreases with the increase of range and incident angle (Yu et al., 2015b). Therefore, the method proposed by Yang et al. (2012) failed to extract road markings that were near road boundaries.

To solve the uneven distribution and fluctuation of intensities, Guan et al. (2014) conducted the multi-threshold segmentation method for road marking extraction using GRF images. The distribution of point density was fitted to a Gaussian distribution function. According to the range r that was calculated by the standard deviation and the estimated mean of the distribution, road surface points were segmented into point bins. Consequently, Otsu’s thresholding algorithm was applied to extract road marking points from each bin. Kumar et al. (2014) and Guan et al. (2015b) used estimated transverse range to divide GRF images into blocks and then determined an intensity threshold for each block based on its range from laser scanners. The accuracy of GRF image based methods can be over 90%; however, road markings were projected to a 2D plane, which inevitably led to some accuracy loss.

The refinement of candidate road marking points in GRF images based methods is directly applied to the extracted pixels. Generally, eliminating false positive and false negative errors and filling the incomplete road markings are two main goals of refinement. False positive and false negative errors can be efficiently eliminated by the median filtering (Cheng et al., 2017) or the multi-scale tensor voting (Guan et al., 2015a). To fill the incomplete road markings, Kumar et al. (2014) converted labeled GRF images to binary images. A linear shaped structuring operator was then used to enlarge the road marking areas and refine the extraction results.

Unlike GRF image based methods, point based road marking extraction methods directly analyze 3D road surface point clouds and detect candidate road marking points. Yan et al. (2016) segmented road surface points into scan lines according to the scanning angle. Then, in each scan line, road marking edge points were detected and preserved through a dynamic window median filter. Finally, an Edge Detection and Edge Constraint method was conducted to remove the gradient edges of extracted road markings. This method required highly accurate MLS point clouds to detect edges between road surface points and road marking points. Yu et al. (2015a) adopted a multi-thresholding strategy to road surface point clouds to solve the uneven distribution of intensities. Road surface point clouds were firstly partitioned into a set of blocks using vehicle trajectory data. Then, each block was compared to a threshold separately to extract candidate road markings. The thresholds are determined via the Otsu's thresholding algorithm (Otsu, 1979). This method offers a promising approach in road marking extraction and it can achieve high accuracy.

The refinement of candidate road marking points extracted by point based methods is also critical since small elements on the road surface might have a high reflective level and cause false extraction noise. False positive and false negative errors can be removed by dividing the candidate road marking points into scan lines (Yan et al. 2016). The method essentially examined the length of extracted road markings and then removed the one shorter than the threshold. However, it had low computational efficiency due to the dividing. Yu et al. (2015a) developed a spatial density filter to eliminate false positive and false negative errors. For each point, the filter analyzed the distribution of its nearby points to determine whether a point should be removed. In addition, the overall geometric features of candidate road markings can be controlled by distance filtering (Holgado-Barco et al., 2017). According to the road painting standards, the method was performed to introduce the semantic constraints, including width and length of road markings, for candidate road marking refinement.

2.3.3 Lane Centerline and Transition Line Generation

In recent studies, lane centerline generation was widely conducted by analyzing GPS trajectories. Lane centerlines were extracted using an analytical equation that best fits the trajectories, with the constant curve as circles and null curve as straight lines (Toledo-Moreo et al., 2010). The method was promising if road pattern was simple. Du & Barth (2008) generated a node structure to represent planar roadways for macro-scale navigation. The node structure consisted of a node table, a link table, a node attribute table, and a link attribute table. A pair of curved lines which represented the boundaries of a lane was described by a set of node points, and then the centerline of the lane was generated by computing the midpoint of every corresponding pair of node points. However, GPS trajectory based lane centerline generation methods required the survey to each lane and driving as close as possible to its center, which is not applicable for large urban areas.

Road intersections are defined as the places where two or more road sections across (Liu et al., 2013a). In road environments, there is no marked transition path from a lane to another in road intersections; however, it does not indicate that vehicles can drive freely in these areas. Therefore, generating a “virtual” line that connects one lane to another while maintaining the continuity between these two lanes is a great solution for HDRM construction. Cao & Krumm (2009) proposed a basic pattern for square areas in intersections. Lanes that entered the square areas and exited the square areas were classified as ‘entry’ and ‘exit’ nodes of an intersection, respectively. ‘Entry’ nodes can be connected to “exit” nodes based on the prior knowledge of traffic rules. A virtual line that entered from an ‘entry’ node and exited from an ‘exit’ node would represent the transition path for the two lanes.

To generate transition lines, Zhang et al. (2011) and Liu et al. (2013a) assumed that vehicles traveled at the road intersection with a constant steering angle. A lane centerline was firstly simplified as a line segment with a pair of ‘entry’ and ‘exit’ node, and then direction symbols were calculated by the vectors of the start lane and the end lane. Finally, an arc section was calculated to represent the transition line. This method provided a solution to link ‘exit’ nodes and ‘entry’ nodes; however, it cannot preserve the continuity at joint points. Guo et al. (2016) presented a promising method for transition line generation. They employed the cubic Catmull-Rom spline with five control points to generate line segments from an ‘exit’ node to an ‘entry’ node and to keep the continuity at joint points. The Central point of a road intersection, the ‘exit’ lane, and the ‘entry’ lane were utilized to determine the five control points. The success rate reached 90% in the experiment.

2.4 Chapter Summary

This chapter firstly reviews the curb-based road surface detection methods in recent studies. It can be summarised that scan line, road patch and voxel are three types of computing unit for road surface detection using MLS point clouds. Although scan line and road patch based algorithms promise impressive results in curb detection, most of them are only applicable to straight roads. Curved trajectories in road intersections might cause false segmentation of MLS point clouds. Moreover, most of the voxel based algorithms have low computational efficiency. In allusion to the problems in the existing studies, an algorithm for road surface detection in road intersections should be proposed.

Furthermore, methods for road marking extraction and transition line generation are elaborately reviewed. It can be concluded that Yu’s method (Yu et al., 2015a) can achieve a promising result in road marking detection using MLS point clouds. Moreover, Guo’s method (Guo et al., 2016) can successfully generate transition lines and preserve continuity at joint nodes using GPS trajectory data. However, generating transition lines using road markings extracted from MLS point clouds has not yet been realized. To solve this problem in existing studies, a road marking node structure generation algorithm should be developed.

This thesis is attempting to tackle the problems mentioned in this chapter.

Chapter 3 Methodology for Transition Line Generation

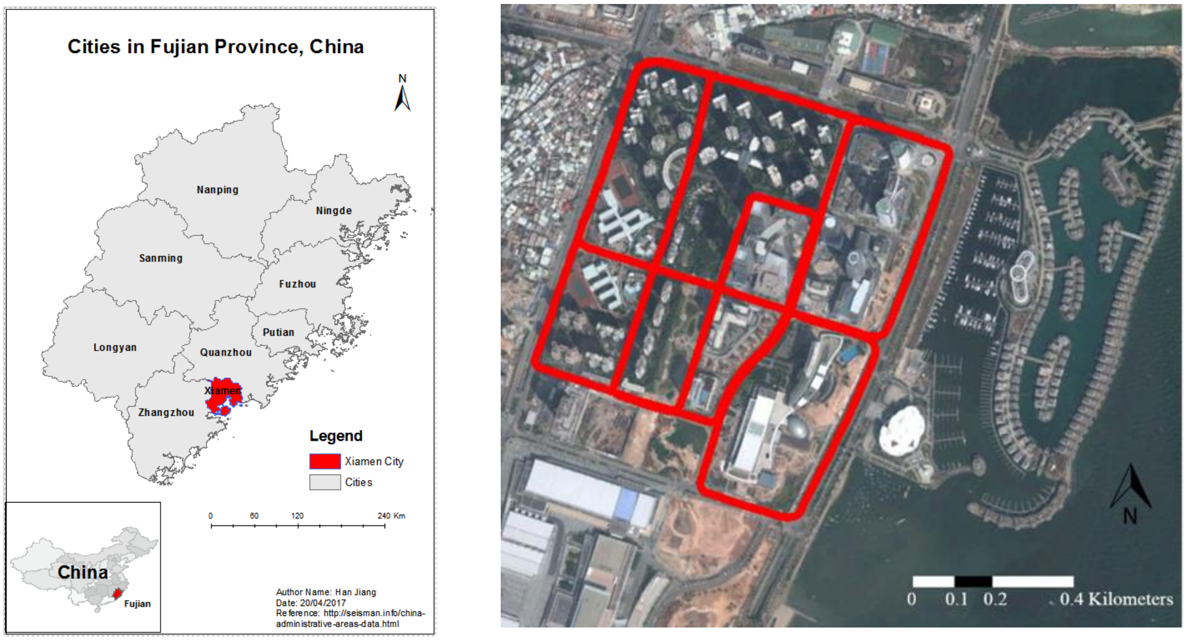
This chapter elaborately introduces the methodology. Section 3.1 describes the study area, the MLS datasets, and the reference datasets. Section 3.2 presents an overview of the workflow. Section 3.3 explains the road surface detection method. Section 3.4 explains the lane marking extraction method. Section 3.5 explains the lane centerline and transition line generation method. The accuracy assessment mechanism is presented in Section 3.6. Section 3.7 summarises this chapter.

3.1 Study Area and Datasets

3.1.1 Study Area

The study area is in Xiamen City, southeastern Fujian, China (longitude 24°28’47.41″N, 118°05’21.91″E) (Figure 3-1 (a)). It is a tropical city that has a monsoonal humid subtropical climate. This kind of climate is characterized by mild and dry winter and hot and humid summer. The vegetation in Xiamen City is year-long green. Since there is less rainfall during the winter season and it is more suitable for data collection, the point clouds were collected on December 20th, 2013. Figure 3-1 (b) shows the block embraced by Roundabout Main Road, Huandao East Road, Huizhan North Road, and Lvling Road. Roads in this area were scanned by a RIEGL VMX-450 MLS system.

The surveyed area, called International Conference and Exhibition Center Block, is an urban area where traffic is free-flowing. As a result, most road markings and road curbs in this area are complete, and cracks are rare on road surfaces. Moreover, roads with two lanes (one lane in each direction) or four lanes (two lanes in each direction) are intersected in this area. T-intersections and cross-intersections are formed. In addition, the intersections are characterized by numerous roadside trees, vehicles, light poles, and traffic poles. Diverse types of road intersections with the various numbers of lanes makes this area ideal for testing the proposed method.



(a) Location of the study area (b) Trajectory (red) overlaid on Google imagery

Figure 3-1 Study area in Xiamen, Fujian, China

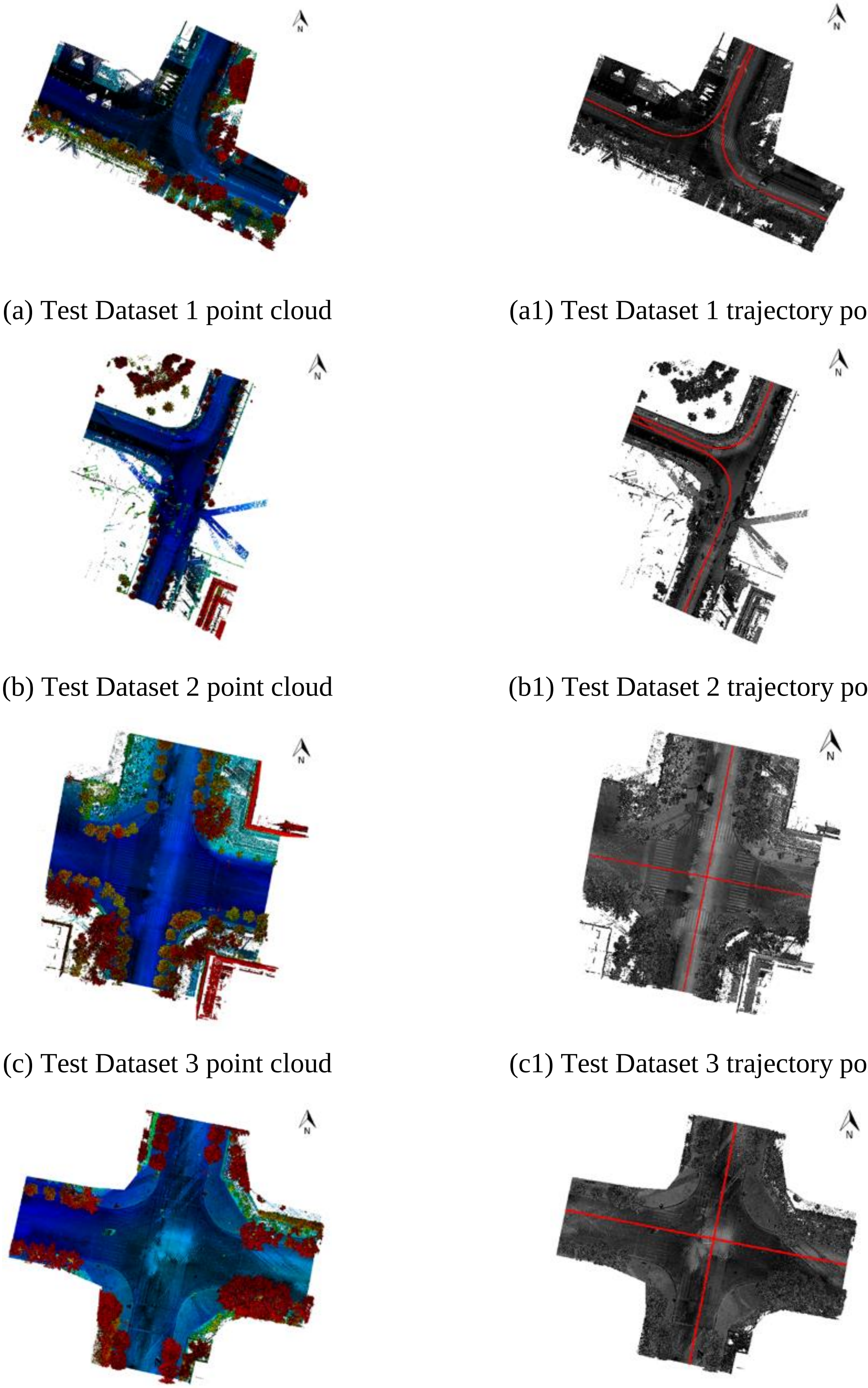
3.1.2 Characteristics of MLS Point Cloud Datasets

A total of 2.961 billion points with the size of 19.7 GB were collected and stored in 17 LAS files. The gray-scale values of the MLS intensity were transformed to 0 to 255. About 0.78 billion trajectory points were generated and provided as ancillary data. Test datasets in this study are selected according to road intersection type, road intersection pattern, and road marking type. Firstly, there are two typical types of road intersections in the study area: T-intersection and cross-intersection. Furthermore, there are two-lane roads and four-lane roads in the study area. Road intersection patterns and constraints in transition line generation are different when roads with different number of lanes are intersected. Moreover, road markings might be various in different road intersections. In addition, vehicle trajectory should cover all branch roads of an intersection, since trajectory points are used as essential prior knowledge and input data in the proposed method.

Therefore, four test datasets, including two T-intersections (Test Datasets 1 and 2) and two cross-intersections (Test Datasets 3 and 4) are selected from the original point clouds. Test Dataset 1 is a T-intersection where two four-lane roads and one two-lane road intersected, and Test Dataset 2 is a T-intersection where three four-lane roads intersected. Test Datasets 3 and 4 are cross-intersections where four four-lane roads intersected. Since they cover all road intersection types and road marking types in the study area, these four datasets are selected. All the four road intersections are characterized by detectable road curbs, standard road markings, and complex roadside objects. In Test Datasets 1, 3, and 4, road markings are in ideal condition and outliers such as vehicles, passengers are rare. In Test Dataset 2, some road markings are blocked by obstacles. Statistical information of the four test datasets is shown in Table 3-1, where ∆X, ∆Y, and ∆Z is an extension of the test datasets in X, Y, and Z direction, respectively. Furthermore, Figure 3-2 presents the four test datasets and their corresponding trajectory points.

Table 3-1 Statistical information about four test datasets

Test Dataset	No. of Points	Average Point Density (pts/m^2)	∆X (m)	∆Y (m)	∆Z (m)	No. of Trajectory Points	Intersection Type
1	19,243,445	2644	117.7	94.6	61.7	13748	T-intersection
2	25,537,193	2782	173.8	119.2	40.6	25982	T-intersection
3	17,444,202	2216	95.3	86.2	72.3	12946	Cross-intersection
4	16,094,181	3210	85.8	73.2	31.1	16973	Cross-intersection



(a) Test Dataset 1 point cloud

(a1) Test Dataset 1 trajectory points

(b) Test Dataset 2 point cloud

(b1) Test Dataset 2 trajectory points

(c) Test Dataset 3 point cloud

(c1) Test Dataset 3 trajectory points

(d) Test Dataset 4 point cloud

(d1) Test Dataset 4 trajectory points

Figure 3-2 Four test datasets (a) to (d) and their trajectory data (a1) to (d1).

3.2 Overview of Workflow

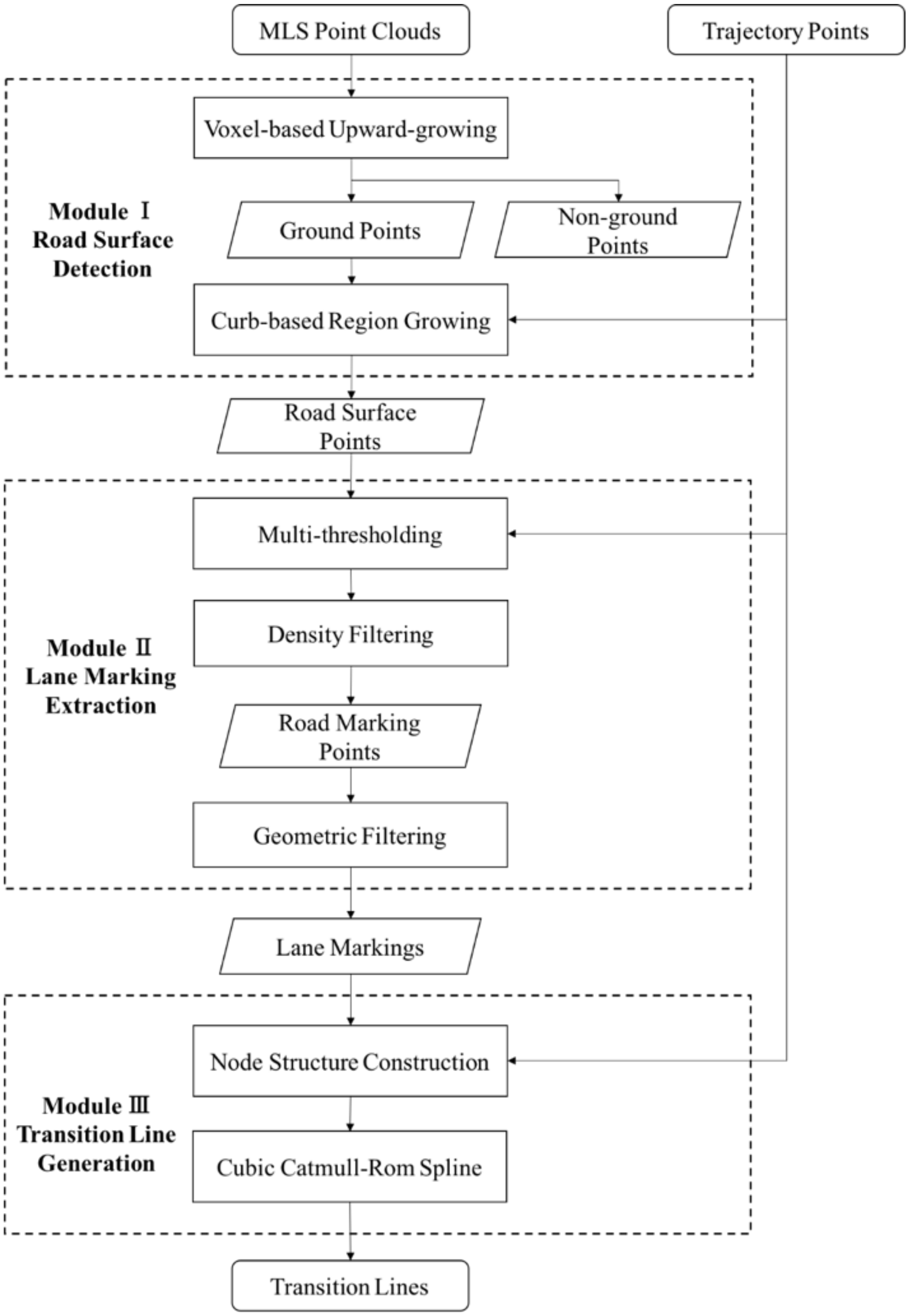


Figure 3-3 Workflow of the proposed method

The proposed methodology contains three modules: road surface detection, lane marking extraction, and transition line generation. Figure 3-3 presents the workflow of the proposed methodology. In Module I, the voxel-based upward-growing algorithm (Yu et al., 2015b) is employed for ground point detection from input point clouds. The region growing algorithm is improved to enhance curb-based road surface detection. In Module II, the multi-thresholding algorithm (Yu et al., 2015a) is used for road marking extraction. Geometric feature filtering (Holgado-Barco et al., 2017) is implemented to extract lane markings. In Module III, a node structure generation algorithm is proposed to generate lane geometries and lane centerlines from lane markings. The cubic Catmull-Rom spline (Catmull & Rom, 1974; Guo et al., 2016) is employed for transition line generation.

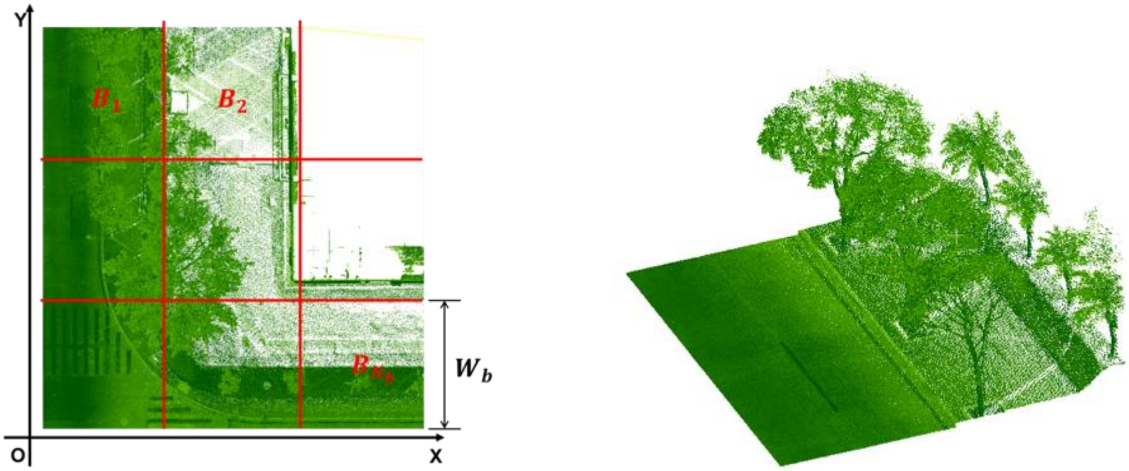
The programming platform used in this thesis is Microsoft Visual Studio 2012. Algorithms, including the voxel-based upward-growing, the region growing, the multi-thresholding, the geometric filtering and the lane node structure generation, are implemented in C++ programming language. A third-party library, Point Cloud Library, is employed to realize K-dimensional tree construction, least-squares fitting, and density filtering. MATLAB R2016 is used to generate the cubic Catmull-Rom splines. In addition, Cloud Compare 1.04 is employed to display the test datasets and experimental results. Accuracy assessment is conducted in ArcGIS v10.2.2.

3.3 Module I: Road Surface Detection

The essential information for transition line generation is merely from road surface points. In other words, the non-road surface points should be removed to improve the efficiency. Thus, an algorithm that identifies road surfaces in road intersections is required. In this thesis, the region growing algorithm is improved to enhance road surface detection. The algorithm is characterized by using trajectory points as a prior knowledge and using voxels as the region growing data structure.

3.3.1 Ground Point Detection

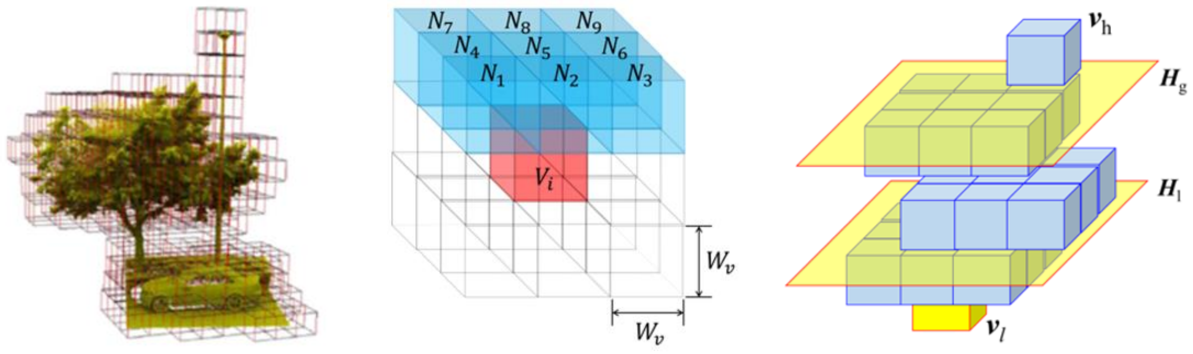
The voxel-based upward-growing algorithm (Yu et al., 2015b) is employed to detect ground points. Since MLS point clouds have large data volumes and high point density, the removal of non-ground (such as traffic signs, traffic lights, light poles, and tree crowns) can improve the computational efficiency and accuracy of the following algorithms. The algorithm contains three steps: point cloud partitioning, upward growing, and ground point labeling.



(a) Point cloud segmented into blocks (b) A sample of point block

Figure 3-4 Segmentation of point clouds

Point clouds are firstly partitioned into point blocks to alleviate the ground undulation in each block and to decrease the processing time. As shown in Figure 3-4 (a), a point cloud scene is partitioned into a set of point blocks ($B_i, i = 1, 2, 3, \dots, N_b$) according to a width W_b in the XOY plane, which is determined by the size of test datasets, in the XOY plane. Figure 3-4 (b) presents a sample of point block. Points in each block are analyzed and processed separately in the following steps.



(a) Voxelization of a point cloud (Yu et al., 2015)

(b) 26 adjacent voxels of V_i

(c) Up-ward growing and thresholding

Figure 3-5 Voxelization of point clouds

Voxelization is a segmentation process which divides the point cloud space into continuous cuboid-shape blocks and represents them as voxels (Figure 3-5 (a)). Voxelization creates a consecutive data structure for point clouds, and it makes the searching and the indexing of points more efficient. Moreover, through voxelization, the impacts of ground undulation can be eliminated, and thresholds can be set to classify ground points and non-ground points. Cube-shaped voxels with the width of W_v is used in this step. Voxels that contains no points are ignored in the processing. As shown in Figure 3-5 (b), a voxel V_i has 26 adjacent voxels, with nine of them located above it ($voxels\ N_1,\ N_2, \dots, N_9$). The upward-growing process grows along with N_1 to N_9. If a point is involved in $N_1, N_2, \dots, or\ N_9$, that voxel is used as a new starting voxel to grow up. This recursive upward-growing pattern stops when there is no point in $N_1, N_2, \dots, and\ N_9$.

As illustrated in Figure 3-5 (c), a voxel segment is formed after the upward-growing, with a known voxel V_h at the top of the segment and a known voxel V_l at the bottom of the segment. Then, a ‘local height value’ variable (H_l) is defined as the elevation difference between V_h and V_l. Similarly, a ‘global height value’ variable (H_g) is defined as the elevation difference between V_h and the lowest point in the entire point cloud. Two criteria are used to label the ground voxels

(voxels that contain ground points): a local ground undulation threshold T_l to constrain the maximum ground undulation within a block and a global ground undulation threshold T_g to constrain the maximum ground undulation for the entire point cloud. If H_g of a voxel is less than T_g, and H_l of the voxel is less than T_l, the voxel will be labeled as a ground voxel. Otherwise, it is labeled as a non-ground voxel (a voxel that contain non-ground points). By filtering out the non-ground voxels, ground points are detected from the ground voxels.

3.3.2 Road Surface Detection

The region growing algorithm is improved to enhance road surface detection, mainly in two ways. First, trajectory points and road curbs are utilized as a prior knowledge. Second, voxels are used as processing units for growing and computing. The algorithm can detect road surfaces from road intersection scenes as well as straight road scenes. Figure 3-6 shows the three steps involved in this algorithm: seed voxel selection, voxel growth, and voxel labeling.

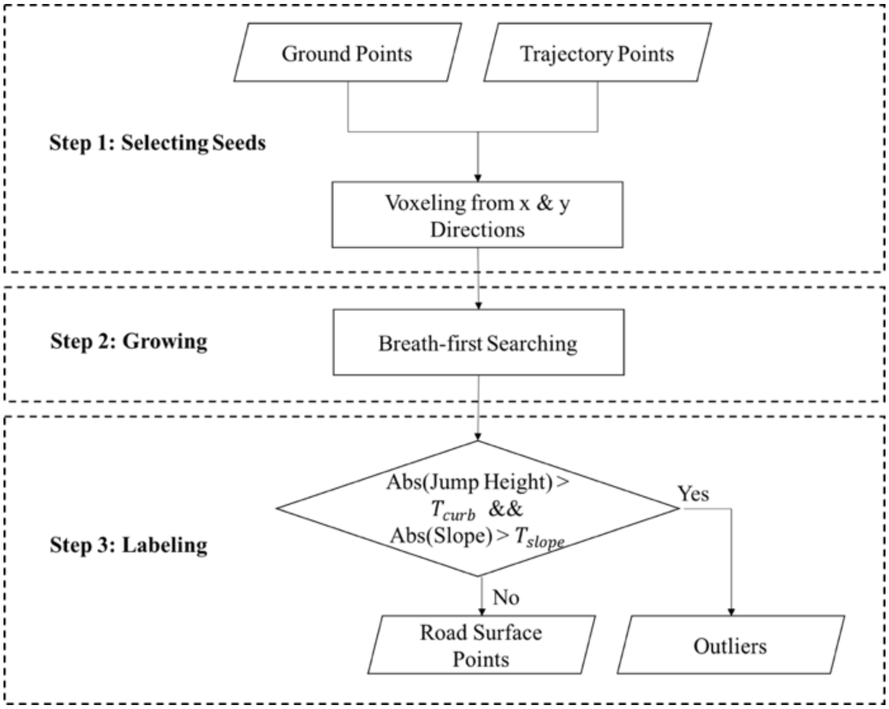
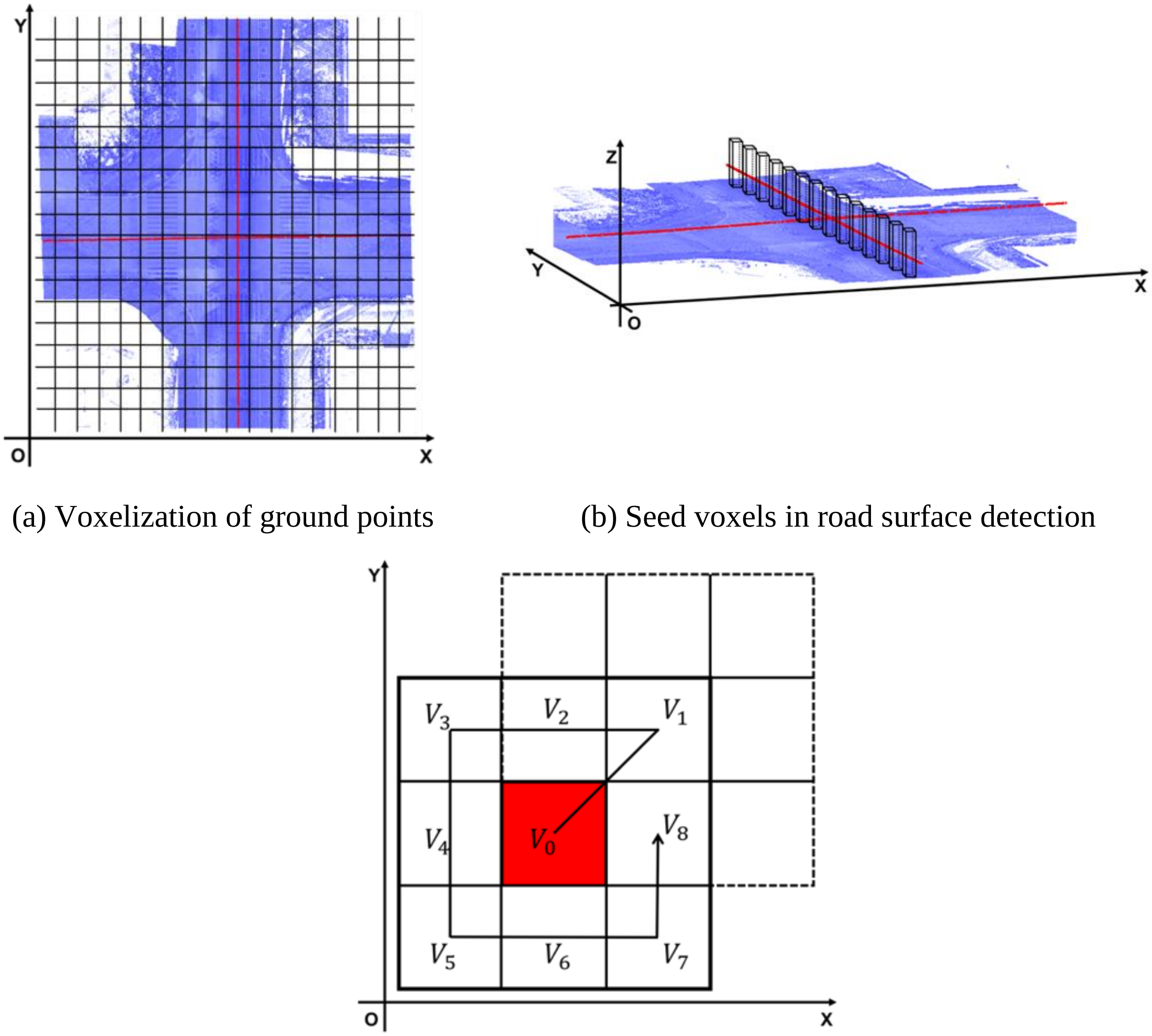


Figure 3-6 Workflow of curb-based road surface detection



(a) Voxelization of ground points

(b) Seed voxels in road surface detection

(c) Breadth-first searching

Figure 3-7 Principle of the region growing

Trajectory points are important indicators of the road surface in seed selection because they mostly exist above road surfaces. Nevertheless, trajectory points cannot be directly used as seed points, since they all have a higher elevation than nearby road surface points. Thus, the processing unit should contain trajectory points and ground points simultaneously. To achieve this, the shape of voxels is modified. As shown in Figure 3-7 (a) and (b), the ground points (blue) and the trajectory points (red) are gridded into cuboid-shaped voxels, which have a height equal to the point cloud space. Constructing these voxels generates the link between road surface points and the trajectory points. Those voxels that contain at least one trajectory point and one road surface point are selected as the original seed voxels.

The Breath-first Searching (BFS) algorithm (Moore, 1959) is adopted in the voxel growth step. As shown in Figure 3-7 (c), a seed voxel V_0 has eight adjacent voxels, named $V_1, V_2, \dots, V_8$. The BFS algorithm exhaustedly searches V_1 to V_8 in sequence. If any of these adjacent voxels does not contain a road curb segment, it is pushed into a queue. Once V_1 to V_8 are all searched and labeled, the BFS algorithm picks up the next voxel in the queue as a new seed voxel and repeats the same searching pattern.

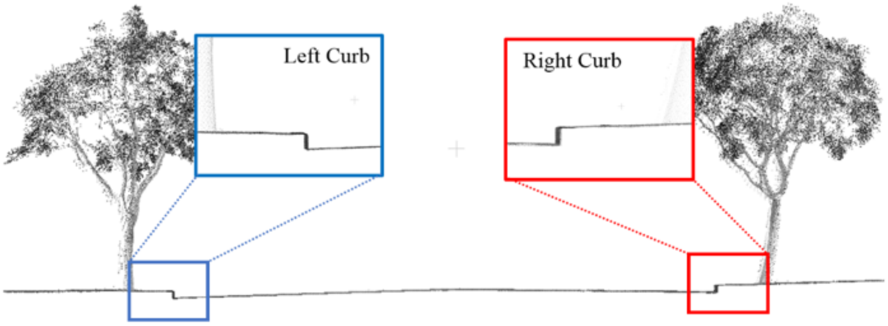


Figure 3-8 Close-up illustration of road curbs

Figure 3-8 shows the close-up views of the road curbs and demonstrates that the road curbs are vertical or mostly vertical to road surfaces and have obvious elevation jumps. Therefore, road curbs can be detected through elevation jump and slope thresholds. Road curb voxels (voxels that contain road curb points) are estimated based on the following two observations.

1) Elevation jump

If part of a road curb is covered by a voxel, the elevation jump in the voxel should meet the following observation (Guan et al., 2015a):

$$T_{curb\_min} \leq E_c \leq T_{curb\_max} \, , \tag{3-1}$$

where E_c denotes the elevation jump in the voxel, and T_{curb_min} and T_{curb_max} is the threshold of the elevation jump. According to Guan et al. (2015a), street design and construction manuals in many countries document that the height of road curbs generally ranges from 10 cm to 25 cm, and T_{curb_min} and T_{curb_max} is predefined as 8 cm and 30 cm, respectively.

2) Slope

The slope observation is defined as following (Guan et al., 2015a):

$$\arctan\left[\frac{(z_{i+1} - z_i)}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}\right] \geq T_{slope} \, , \tag{3-2}$$

where (x_i, y_i, z_i) and $(x_{i+1}, y_{i+1}, z_{i+1})$ are the coordinates of two adjacent points in a voxel, and T_{slope} is the slope threshold. According to Guan et al. (2015a), the T_{slope} is predefined as $2\pi/3$.

The growth stops when the following conditions occur: first, there is no seed voxel in the queue; second, the distance from a searched voxel to the original seed voxel is greater than the pre-defined threshold. The first condition indicates that the searching is completed. As an additional stop condition, a distance is set to restrain the searching radius of the BFS algorithm. Because of the ground's undulation, if the growth from an original seed with higher elevation is not limited, road curbs located in lower positions might not be detected.

3.4 Module II: Lane Marking Extraction

Lane marking refers to road markings that indicate lane edges. Typical lane markings include single lines, double lines, and dashed lines. Since lane markings are critical information for lane centerline generation, they are extracted from the road surface points using four adapted algorithms. The multi-thresholding and the density filtering (Yu et al., 2015a) are employed for road marking point extraction. Furthermore, lane markings are extracted through the Conditional Euclidean clustering (Yu et al., 2015b) and the geometric feature filtering (Holgado-Barco et al., 2017).

3.4.1 Road Marking Extraction and Refinement

The extraction of road marking points contains three steps. Road surface points are firstly segmented into point sets by the multi-range partitioning. Then, an intensity threshold is determined by the Otsu filter (Otsu, 1975) for each point set and extract candidate road marking points, since the Otsu method has been widely utilized as the default image thresholding method in commercial software such as MATLAB (Guan et al., 2015a). Finally, the extracted candidate road marking points were refined by the density filter. The workflow of this algorithm is shown in Figure 3-9.

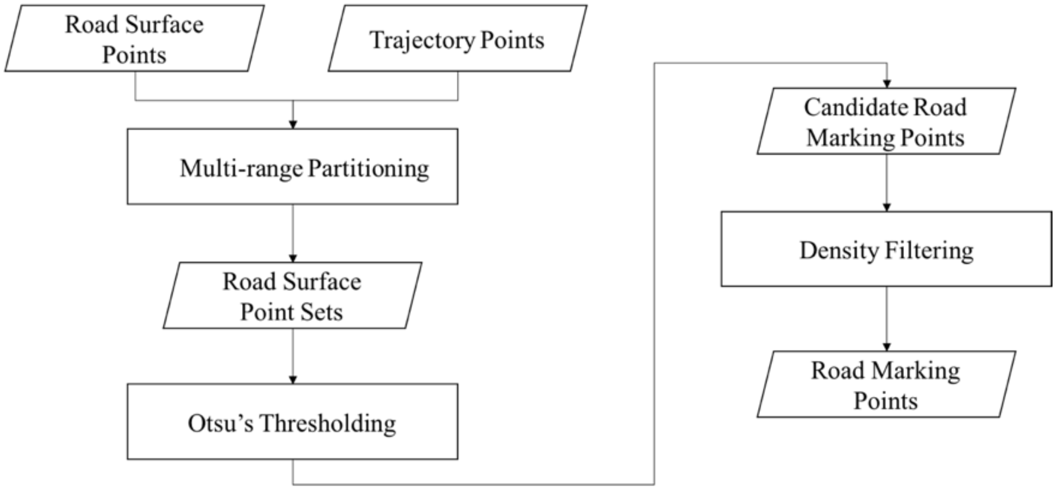
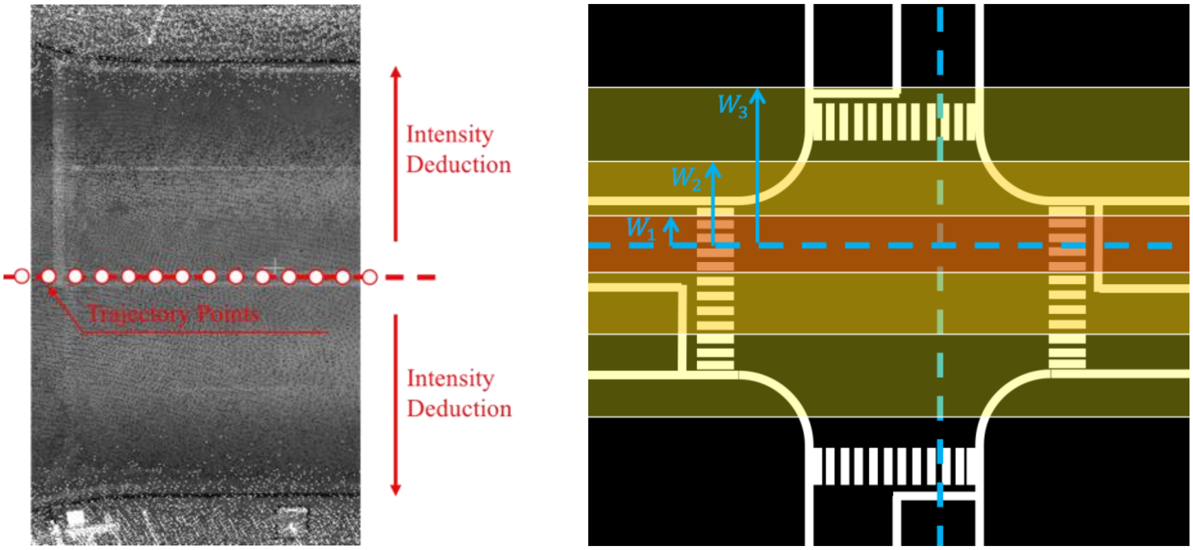


Figure 3-9 Workflow of road marking extraction

Generally, road markings are painted by highly reflective material on the road surfaces. Thus, road marking points have higher intensities than surrounding road surface points. However, the intensity of a reflected laser pulse depends not only on the properties of the scanned surface but also on the distance from the laser scanner to the scanned surface and the incidence angle of the laser pulse. As shown in Figure 3-10 (a), the overall intensity of a road surface point cloud fades from the scanning center to its two sides. Consequently, road marking points closer to the scanning center have higher intensities than road marking points farther away from the scanning center. Due to the uneven distribution of intensities, road surface point clouds are partitioned into point sets based on the distance from points to the trajectory. As shown in Figure 3-10 (b), centered at a trajectory, multiple buffer regions are created with a width W_i, defined as

$$W_i = w_s * i \ (i = 1, 2, \dots, n) \ , \tag{3-3}$$

where w_s is the original buffer width and n is the number of buffers. The partitioning starts from the smallest buffer to the largest one.



(a) Intensity deduction in MLS point clouds (b) Road surface segmentation

Figure 3-10 Multi-thresholding road marking extraction

The intensity threshold for road marking extraction is separately by Otsu’s thresholding algorithm in each point set. The algorithm exhaustively searches the threshold that maximises the inter-class variance (or minimises the intra-class variance). The scale of intensity should be normalized to 0-255. The number of point at level i is denoted by n_i and the total number of point is $N = n_0 + n_1 + \cdots + n_{255}$. The algorithm is detailed by Otsu (1979) through Eqs. (3-4) to (3-11).

A probability distribution can be generated from the gray-level histogram:

$$P_i = n_i/N, \quad P_i \geq 0, \quad \sum_{i=0}^{255} P_i = 1\,. \tag{3-4}$$

Suppose that road marking points are extracted from a point set by a threshold at gray level t. The occurrence probabilities of road pavement ω_s and of road marking ω_m, respectively, are

$$\omega_s = \omega(t) = \sum_{i=0}^{t} P_i\,, \tag{3-5}$$

and

$$\omega_m = 1 - \omega(t) = \sum_{i=t+1}^{255} P_i\,, \tag{3-6}$$

where $\omega(t)$ is the zeroth-order cumulative moment up to the *t*th level in the histogram. The corresponding mean levels are

$$\mu_s = \mu(t)/\omega(t) = \sum_{i=0}^{t} iP_i/\omega_s\,, \tag{3-7}$$

and

$$\mu_r = \frac{\mu_T - \mu(t)}{1 - \omega(t)} = \sum_{i=t+1}^{255} iP_i/\omega_m\,, \tag{3-8}$$

where

$$\mu(t) = \sum_{i=0}^{t} iP_i\,, \tag{3-9}$$

and

$$\mu_T = \sum_{i=0}^{255} iP_i \, , \tag{3-10}$$

are the first-order cumulative moment of the histogram up to the *t*th level and the total mean level of the road pavement, respectively. The optimal intensity threshold t^* is selected by searching the maximum inter-class variance σ_I^2 through

$$t^* = \arg \max_{0 \le t \le 255} \sigma_I^2\,(t) = \arg \frac{[\mu_T \omega(t) - \mu(t)]^2}{\omega(t)[1 - \omega(t)]} \, . \tag{3-11}$$

With the generated threshold, candidate road marking points can be extracted from the point set.

Extracted candidate road marking points might contain false extraction noise. The noise is caused by small objects that have similar intensities as road markings on road surfaces. To eliminate the noise, the spatial density filtering is conducted. The filter works because unlike road marking points, noise points distribute dispersedly and irregularly. For each candidate road marking point $p(x, y, z)$, its nearby points within a range of R_t are searched. Suppose that the number of the nearby points is k, the spatial density $D(p)$ of p is defined as following (Yu et al., 2015a):

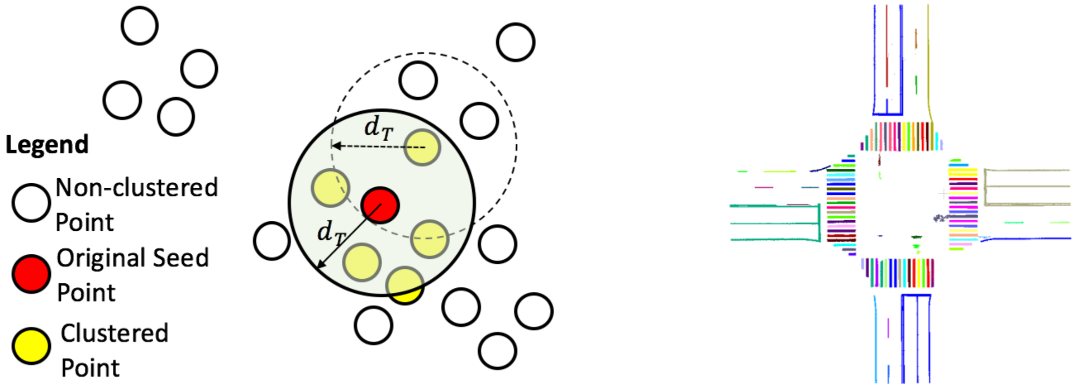
$$D(p) = 1 + \sum_{i=1}^{k} \exp(-\frac{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}{(k/3)^2}) \, . \tag{3-12}$$

Since road marking points have higher spatial density than the noise, the noise can be eliminated by defining a spatial density threshold.

3.4.2 Lane Marking Extraction

The extracted road marking points are distinct and have no semantic meanings. Thus, the road marking points are firstly clustered through the Conditional Euclidean clustering. Then, the minimum bounding rectangle is derived for each point cluster to extract its geometric features.

Lane markings are finally extracted by the geometric filtering based on the design parameters of road markings in the study area.



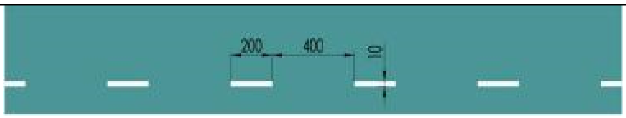
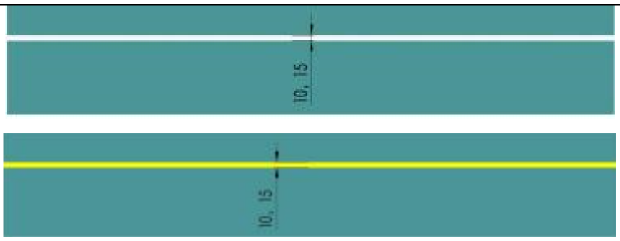
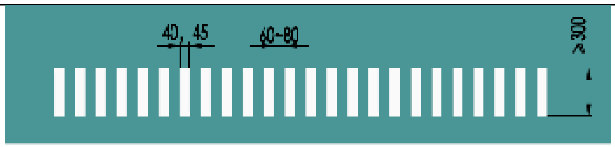
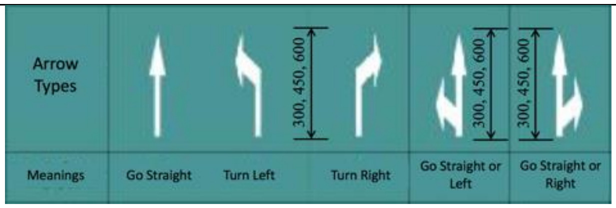
(a) Principle of the conditional Euclidean clustering

(b) A sample of clustered road markings

Figure 3-11 Road marking clustering

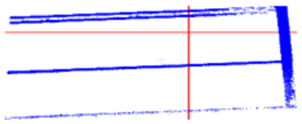
The extracted road marking points are isolated and have no topological relationships. In other words, they have no semantic meanings. To extract lane marking points from road marking points, clustering should be applied. By assuming that nearby points belong to the same road marking, the Conditional Euclidean clustering is employed. Initially, all road marking points are labeled as non-clustered points and a Euclidean distance threshold d_T should be determined according to the density of point clouds. As shown in Figure 3-11 (a), the algorithm starts from randomly selecting an unlabeled point as a seed point and constructing a 3D searching space centered at the seed point with the range d_T. Nearby points within the searching space are labeled and pushed into a queue. When all the nearby points are searched, the algorithm picks up the next point in the queue and processes it with the same procedure. A cluster is generated when there is no point in the queue. Then, the algorithm randomly selects another point as a new seed point and generate the next cluster. Consequently, semantic road marking clusters are generated, as shown in Figure 3-11 (b).

Table 3-2 Design parameters of road markings in China (GB51038-2015)

No.	Road Marking Type	Legend (Unit: cm)	Comment
1	Dashed Line		Allowed to Pass (Speed<60km/h)
2	White/Yellow Line		Boundaries of Lanes
3	Zebra Crossing Line		Pedestrians Might Cross
4	Turning Arrow	 Arrow Types 300, 450, 600 300, 450, 600 300, 450, 600 Meanings Go Straight Turn Left Turn Right Go Straight or Left Go Straight or Right	Presenting the Moving Direction

When road marking points are clustered, geometric features of point segments can be calculated and utilized for lane marking extraction. Generally, there are five types of road marking in the test datasets: solid line, dashed line, zebra crossing line, Chinese character, and turning arrow. The road marking painting standards, which are documented in the Code for Layout of Urban Road Traffic Signs and Markings (GB51038-2015), is shown in Table 3-2. A minimum bounding rectangle is generated to extract the shape features of each road marking cluster. According to the length L and width W of the minimum bounding rectangles, lane markings are extracted using the geometric parameters listed in Table 3-3.

Table 3-3 Parameters for lane marking extraction

Road marking type	Range of L and W	Sample
Dashed lines	$150\ cm \leq L \leq 250\ cm$ $\&\ 5\ cm \leq W \leq 20\ cm$	
Single lane lines	$400\ cm \leq L \leq 2500\ cm$ $\&\ 5\ cm \leq W \leq 20\ cm$	
Lane lines connected by a stop line	$600\ cm \leq L \leq 2500\ cm$ $\&\ W \geq 500\ cm$	
Outliers	Otherwise	

3.5 Module III: Transition Line Generation

This section describes the algorithms for transition line generation. As shown in Figure 3-12, lane node structure, which represents lane geometries in road intersections, is firstly constructed by a combination of lane marking node structure generation and 'exit' and 'entrance' node pairing. Moreover, the cubic Catmull-Rom spline is employed to generate transition lines for paired nodes (Catmull & Rom, 1974; Guo et al., 2016).

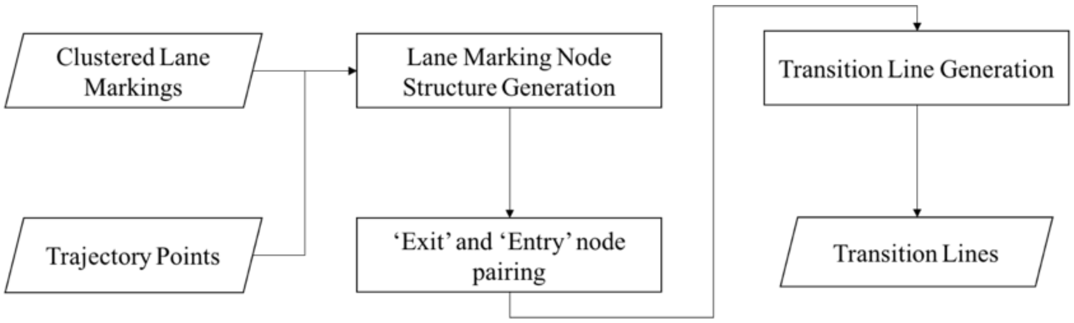
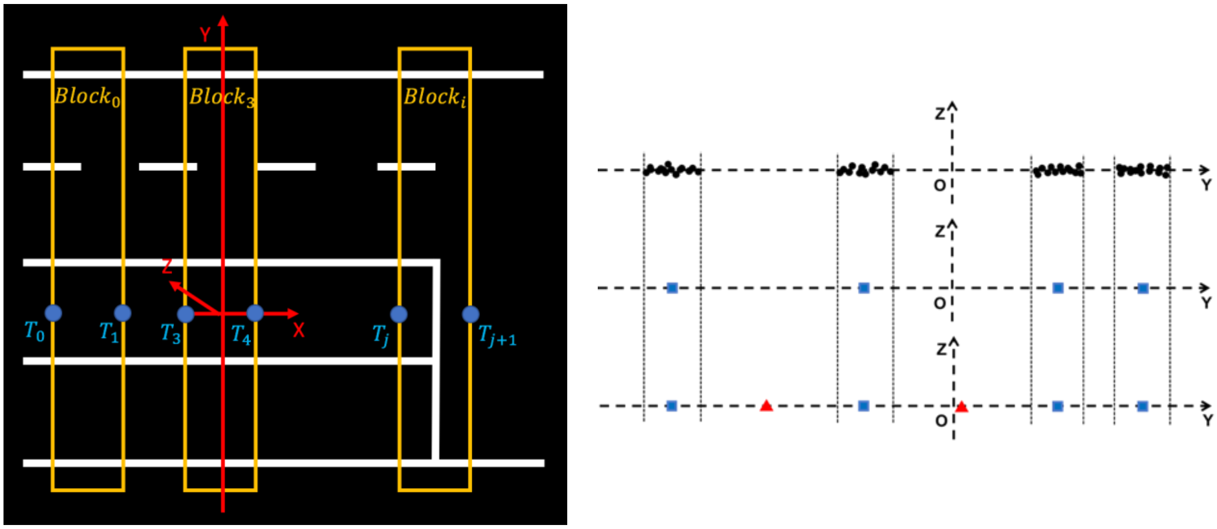

Clustered Lane Markings
Trajectory Points
Lane Marking Node Structure Generation
'Exit' and 'Entry' node pairing
Transition Line Generation
Transition Lines

Figure 3-12 Workflow of transition line generation

3.5.1 Lane Node Structure Construction

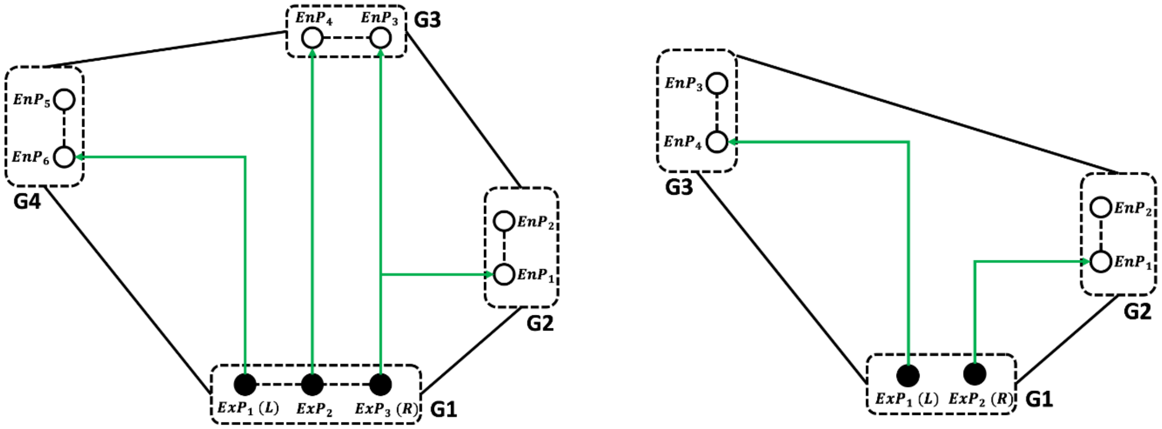
The lane node structure construction algorithm utilizes lane marking points and trajectory points as input data. The trajectory points are resampled at an interval I_T to generate a point set T_j $(j = 0, 1, \dots, L)$. As shown in Figure 3-13 (a), T is used to partition the lane marking point cloud into blocks ($Block_i$ $(i = 0, 1, \dots, N)$). The length of blocks is restricted by a threshold T_B. For each block, a Cartesian coordinate system is confirmed, with an X axis towards the front of the vehicle, a Y axis towards the left of the vehicle, and a Z axis towards the top of the vehicle. The coordinate origin is set at the midpoint of the line segment that connects T_j and T_{j+1}.



(a) Lane marking partitioning (b) Node structure generation

Figure 3-13 Lane node structure construction

Figure 3-13 (b) presents the processing of a block $Block_i$. Firstly, lane marking points (black) in $Block_i$ are projected onto the YOZ plane of the coordinate. Then, by exhaustedly searching the projected lane marking points, central points MP_k $(1 \leq k \leq 4)$ (blue) of lane markings are computed. If the width of a lane marking segment is greater than a threshold T_M, it is labeled as a stop line (the white line where vehicles stop for traffic). Furthermore, candidate lane centerline nodes (red) are generated by calculating the midpoint of MP_k and MP_{k+1}. If the range between MP_k and MP_{k+1} is less than a threshold R_T, the candidate lane centerline node calculated from them is removed. In addition, by assuming that lanes are straight near road intersections, the least square fitting is employed to generate lane centerlines from generated nodes. If a lane centerline has an intersection with a stop line, it is labeled as 'exit' lane, and its vertex is labeled as 'exit' node. Otherwise, the lane is labeled as 'entry' lane, and its vertex is labeled as 'entry' node.



(a) Pairing 'entry' and 'exit' nodes in a cross-intersection

(b) Pairing 'entry' and 'exit' nodes in a T-intersection

Figure 3-14 Pairing 'entry' and 'exit' nodes

When the node structure of a road intersection is generated, all proper 'entry' nodes for an 'exit' node are then determined. As shown in Figure 3-14 (a), 'entry' nodes and 'exit' nodes on four branch roads in a cross-intersection are grouped as G_1, G_2, G_3, and G_4. The convex hull (black lines) of the four groups is firstly generated by the Gift Wrapping algorithm, which is described later, to build topological relationships for them (Jarvis, 1973; Cormen et al., 2001). Then, the algorithm searches the 'exit' nodes (ExP_i $(1 \leq i \leq 3)$) in G_1, and all the 'entry' nodes (EnP_i $(1 \leq$

$i \leq 6)$) in G_2, G_3 and G_4. Corresponding 'entry' nodes EnP_3 and EnP_4 for the 'exit' node ExP_3 and ExP_2, respectively, are in the non-adjacent node group G_2. Additional connections are generated from the rightmost 'exit' node ExP_3 to the 'entry' node EnP_1 and from the leftmost 'exit' node ExP_1 to the 'entry' node EnP_6. The algorithm processes the 'exit' nodes in G_2, G_3 and G_4 in the same pattern.

In T-intersections, two branch roads that have the same direction follow the same pairing rules as branch roads in cross-intersections. However, pairing rules for the other branch road should be additionally discussed, because vehicles cannot go straight ahead on this branch road. As shown in Figure 3-14 (b), 'exit' nodes of the branch road are labeled as solid points ExP_j ($1 \leq j \leq 2$), and 'entry' nodes are labeled as points EnP_j ($1 \leq j \leq 4$). Links are constructed from the rightmost 'exit' ExP_2 to 'entry' node EnP_1 and from the leftmost 'exit' ExP_1 to 'entry' vertex EnP_4. Through the Gift Wrapping algorithm, proper 'entry' lanes for a 'exit' lane can be determined at T- and cross-intersections.

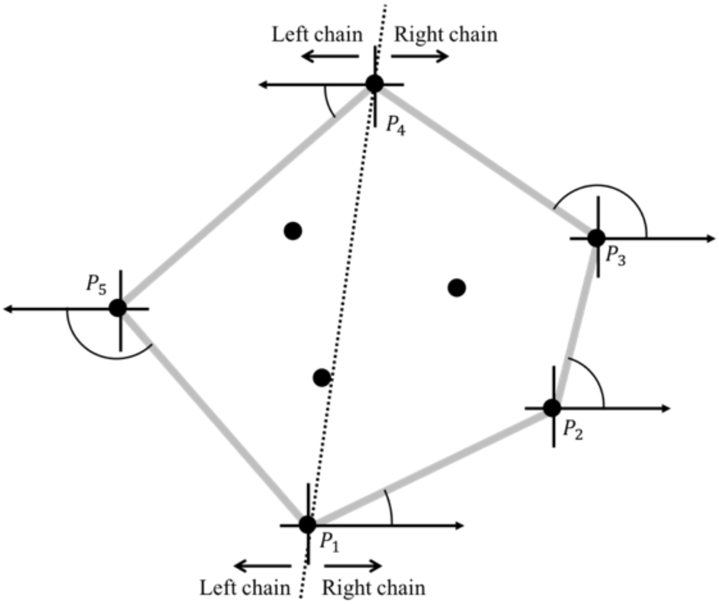
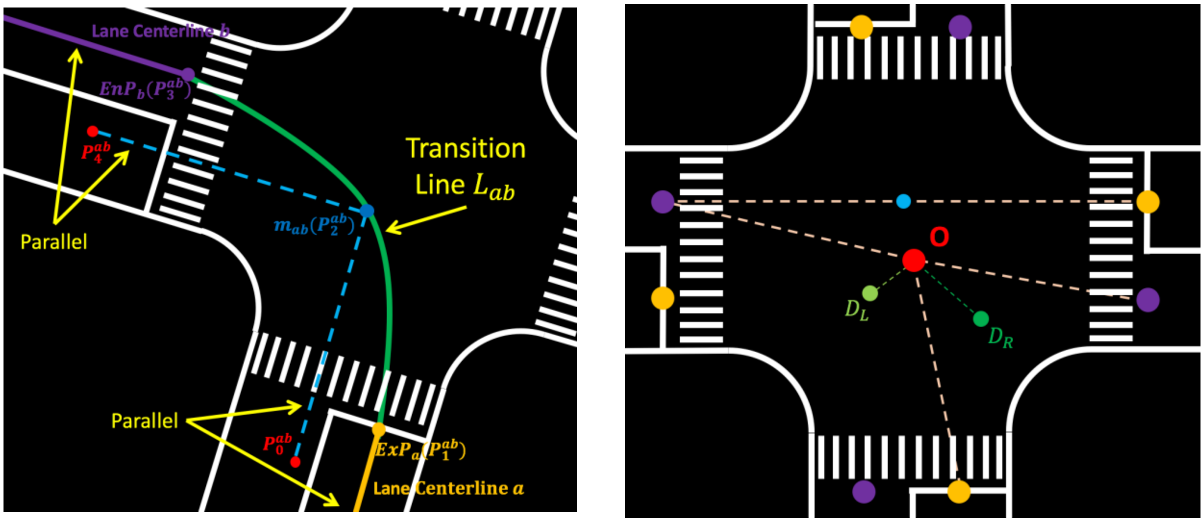


Figure 3-15 Principle of the gift wrapping algorithm (Cormen et al., 2001).

As shown in Figure 3-15, to generate the convex hull CH of a set of points $(P_1, P_2, \dots, P_h)$, the Gift Wrapping algorithm starts with $i = 1$ and P_1, the lowest point, and selects P_{i+1} that has the smallest polar angle with respect to p_1. In case of a tie, the point farthest from P_1 is selected. Letting $i = i + 1$, P_3 has the smallest polar angle with respect to p_2. Using the same pattern, the algorithm can reach the highest point, say p_h, and the right chain of CH is constructed. Then, starting at p_h, p_{h+1} that has smallest polar angle with respect to p_h from the negative x-axis is selected. The left chain can be constructed by repeating the same pattern until original point p_0 is reached again.

3.5.2 Transition Line Generation

To generate a safe path from an 'exit' node to an 'entry' node and to keep the continuity at the nodes, the cubic Catmull-Rom spline is employed (Catmull & Rom, 1974; Guo et al., 2016). Five control points P_k^{ab} $(0 \le k \le 4)$ on a transition line L_{ab} (green curve line), which connects lane centerline a (yellow line) and b (purple line), are shown in Figure 3-16 (a). The 'exit' node ExP_a (yellow dot) of a is set as P_1^{ab}, and the 'entry' node EnP_b (purple dot) of b is set as P_3^{ab}. An intermediate point m_{ab} (blue dot), which is described later in this section, is set as P_2^{ab}. P_0^{ab} is set to a location that makes line $\overline{P_o^{ab} P_2^{ab}}$ (blue dashed line) parallel to a to ensure the continuity at P_1^{ab}, because the tangent vector at a control point on a cubic Catmull-Rom spline is determined by the adjacent control points on two sides of it. The final control point P_4^{ab} is set with the same pattern according to b.



(a) Calculating a transition line (b) Intermediate point generation

Figure 3-16 Transition line generation

When five control points P_k^{ab} $(0 \leq k \leq 4)$ are determined, for $0 \leq t \leq 1$, the cubic Catmull-Rom spline curves are defined by Equation 3-13 (Catmull & Rom, 1974):

$$r_k^{ab}(t) = \sum_{i=0}^{3} f_i(t) p_{k+i}^{ab}, k = 0,1, \tag{3-13}$$

where $f_i(t)$ $(0 \leq i \leq 3)$ are the basis functions of cubic Catmull-Rom splines, and they are defined by the follow equations (Catmull & Rom, 1974):

$$\begin{cases} f_0(t) = \frac{1}{2}(-t + 2t^2 - t^3) \\ f_1(t) = \frac{1}{2}(2 - 5t^2 + 3t^3) \\ f_2(t) = \frac{1}{2}(t + 4t^2 - 3t^3) \\ f_3(t) = \frac{1}{2}(-t^2 + t^3) \end{cases}. \tag{3-14}$$

In addition, cubic Catmull-Rom spline curves have three properties (DeRose & Barsky, 1988). First, the spline curves have symmetry. Thus, P_k^{ab} $(i = 0, 1, \dots, 4)$ and P_{5-k}^{ab} $(i =$

$0, 1, \dots, 4$) define the same curves in a different parameterization. Second, the spline curves have geometric invariance. The shapes of the spline curves are not affected by the choice of coordinate system. Third, the spline curves have C^1 continuity, and the shapes of the spline curves can be modified when control points are changed. Thus, the shaped of transition lines can be adjusted based on the angles of road intersections.

The intermediate point is defined according to Guo et al. (2016) to complete the transition line generation in road intersections. As can be seen in Figure 3-16 (b), the center point O of an intersection is calculated by averaging all the ‘exit’ nodes (yellow) and ‘entry’ nodes (purple). For a left-turn transition line, the intermediate point is defined by an offset distance D_L in the bisector of the angle between $\overrightarrow{ExP_iO}$ and $\overrightarrow{OEnP_j}$. The intermediate point for a right-turn transition line is determined in the same pattern with a larger D_R, since the study area follows the right-hand traffic. For a straight-ahead transition line, the intermediate point is defined as the midpoint (green) of line segment $\overline{ExP_iEnP_j}$. According to the type of transition lines, different intermediate points are defined to generate them.

3.6 Validation Methods

3.6.1 Accuracy Assessment of Road Markings

The performance of the road marking extraction method is assessed by thematic accuracy, which refers to the differences between the labeled attributes of remotely sensed features and the true attributes of reference features (Congalton and Green, 2008). The confusion matrix for the binary classification is illustrated in Table 3-4, where TP represents true positive, TN represents true negative, FP indicates false positive, and FN indicates false negative. In this study, TP is the number of road marking points which are correctly classified, whereas FP is the number of outliers that are falsely classified as road marking points. TN referes to the number of correctly classified outliers, whereas FN refers to the number of road marking points which are misclassified as outliers.

Table 3-4 Confusion matrix for binary classification

Class\Classified	**as Positive**	**as Negative**
Positive	*TP*	*FN*
Negative	*FP*	*TN*

Furthermore, the thematic accuracy is measured by three criteria: recall, precision, and F$_1$-score (Van Rijsbergen, 1979; Sokolova et al., 2006). The recall, which indicates the integrity of the extracted road markings, is expressed as

$$Recall = \frac{C_p}{R_{cpt}} = \frac{TP}{TP + FN}, \tag{3-15}$$

where C_p is the number of points belonging to the actual road markings, and R_{cpt} is the number of road marking points in reference data. The precision denotes the percentage of valid road markings, and it is expressed as

$$Precision = \frac{C_p}{R_{crt}} = \frac{TP}{TP + FP}, \tag{3-16}$$

where R_{crt} indicates the number of points extracted by the proposed method. The global score is evaluated by F$_1$-score, which is obtained by

$$F_1 - score = \frac{2 * recall * precision}{recall + precision}. \tag{3-17}$$

3.6.2 Accuracy Assessment of Lane Centerlines and Transition Lines

The generated lane centerlines and transition lines are assessed by the Buffer-overlay-statistics (BOS) method (Tveite, 1999). The method firstly generates buffers in various sizes around test lines and reference lines and then compares them through overlaying and statistics.

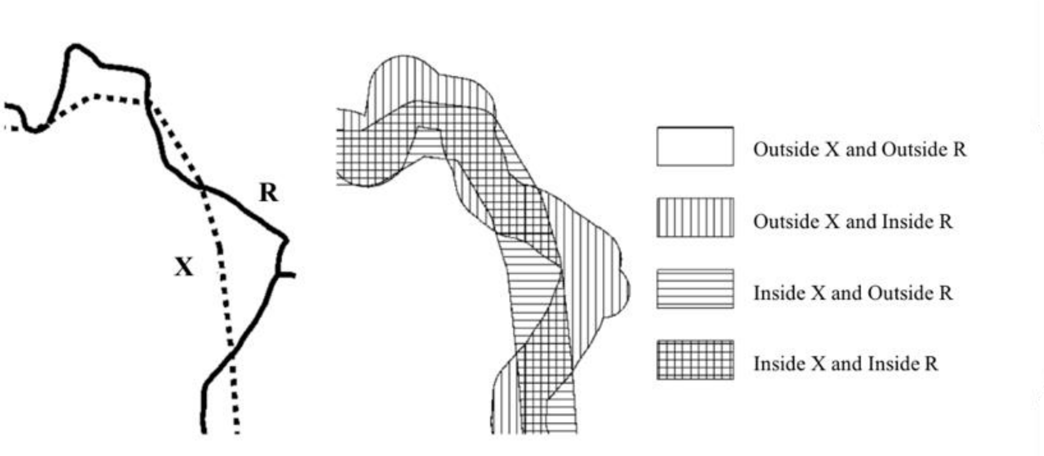


Figure 3-17 Elements used in the BOS method (Tveite, 1999)

The elements used in the BOS method for line feature assessment are presented in Figure 3-17. In the left side of Figure 3-17, there are two line-based datasets, where X is a test dataset, and R is a reference dataset. To the right is the buffering and overlaying result. The method generates four types of result areas: areas that are inside the buffer of X (XB) and the buffer of R (RB) ($XB \cap RB$), areas that are inside XB and outside RB ($XB \cap \overline{RB}$), areas that are outside XB and inside RB ($\overline{XB} \cap RB$), and areas that are outside XB and RB ($\overline{XB} \cap \overline{RB}$). Simultaneously, four types of line segments are formed: line segments of R inside XB, line segments of R outside XB, line segment of X inside RB, and line segments of X outside RB.

To present a quantitative accuracy assessment of lines features, buffering and overlaying is conducted iteratively. For a number (n) of buffer sizes $B_i, (1 \leq i \leq n)$, conduct the following four steps. Firstly, perform buffering on the line features in X and R with B_i, and call the generated buffer polygons XB_i and RB_i, respectively. Secondly, overlay X with RB_i to generate a mixed dataset XRB_i. Overlay R with XB_i to generate a mixed dataset RXB_i. Thirdly, calculate the sum length of the line segments from X outside RB_i in XRB_i and the total length of line features in X. The recall C_i of X is computed by (Tveite, 1999):

$$C_i = \frac{Length(R\ inside\ XB_i)}{Length(R)}. \tag{3-18}$$

Finally, calculate the sum length of the line segments from R inside XB_i in RXB_i and the total length of line features in R. The miscoding M_i of X is computed by (Tveite, 1999):

$$M_i = \frac{Length(X\ outside\ RB_i)}{Length(X)}. \tag{3-19}$$

3.7 Chapter Summary

The details of the test datasets and the entire method of transition line generation using MLS point clouds is presented in this chapter. The method contains three modules. In module one, the voxel-based upward-growing is employed to detect ground points from input point clouds. Subsequently, the region growing is enhanced for road surface detection. In module two, the multi-thresholding is employed for road marking extraction. Geometric feature filtering is adapted to extract lane markings from clustered road markings. In module three, lane marking node structures are constructed through the proposed algorithm, and proper ‘entry’ lanes for ‘exit’ lanes are connected. The cubic Catmull-Rom spline algorithm is employed for transition line generation. In addition, the quantitative assessment methods are stated.

In addition, since the proposed method is a semi-automated method, some parameters might need to be manually adjusted when the method is applied to a new study area. Firstly, lane markings are extracted based on the road marking design standards in the study area. When conducting the algorithm in another study area, especially in other countries, the road marking design standards might be different and parameters should be changed accordingly. Furthermore, in lane node structure algorithm, ‘exit’ and ‘entry’ nodes are paired according to traffic rules in the study area. However, traffic rules might be different in other countries. Thus, rules for ‘exit’ and ‘entry’ node pairing should be adjusted. In addition, five control points for cubic Catmull-Rom splines are determined based on the right-hand traffic in the study area. If the method is conducted to a study area that follows left-hand traffic, the control points should be modified.

Chapter 4 Results and Validation

This chapter presents the stepwise results and validation of transition lines generated by the proposed method, organized into four sections. Section 4.1 shows the results and the analysis of the ground removal algorithm, the road surface detection algorithm, the lane marking extraction algorithm, and the transition line generation algorithm. Section 4.2 includes the accuracy assessment of the extracted lane markings and generated transition lines. Section 4.3 presents a comparative study of road marking extraction. Finally, Section 4.4 summarises this chapter.

4.1 Results and Evaluation

4.1.1 Ground Point Detection

Ground points were detected through the voxel-based upward-growing algorithm. The four parameters used in this algorithm are as follows:

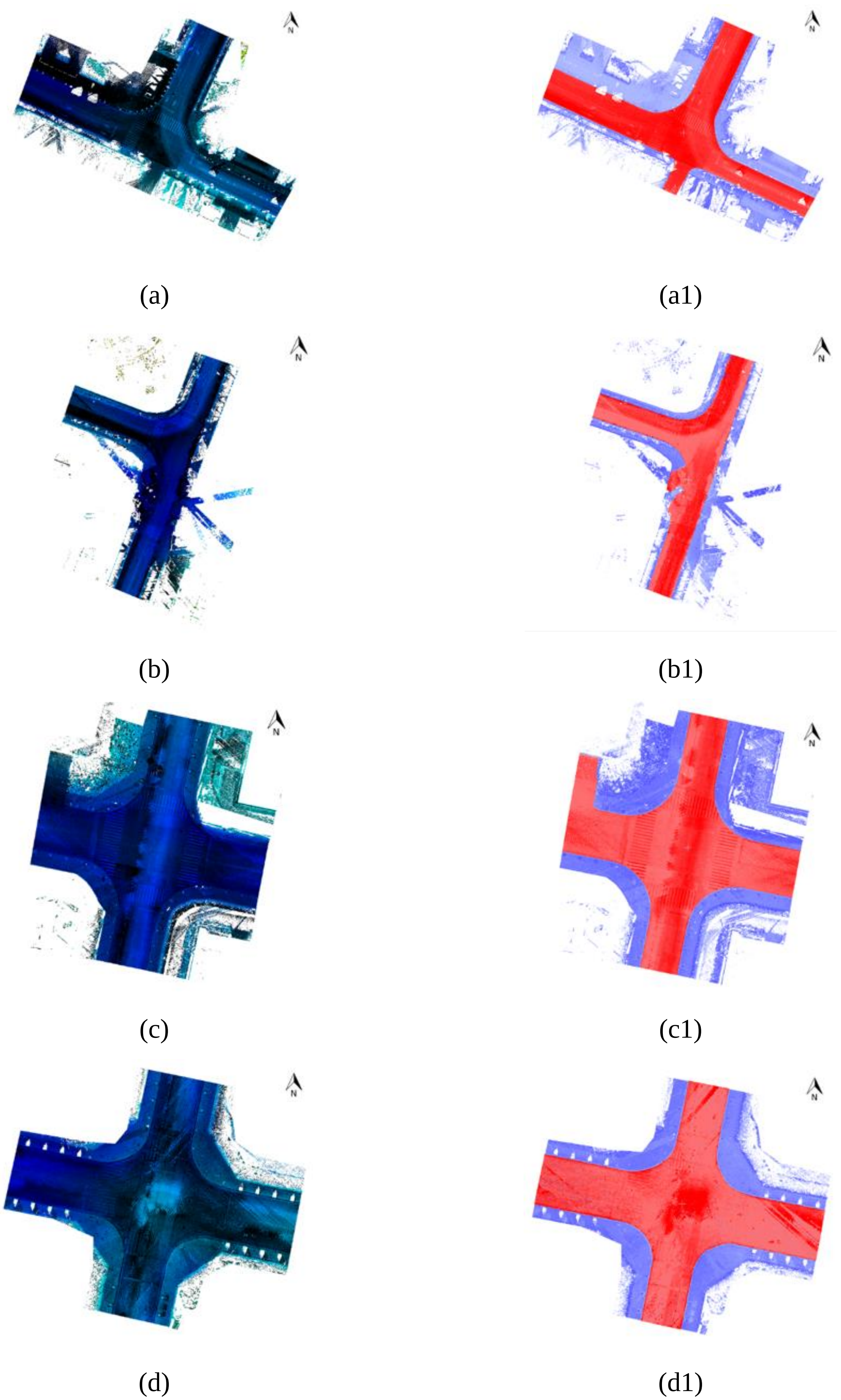
W_b: size of a block,

W_v: size of a voxel,

T_g: a global ground undulation threshold, and

T_l: a local ground undulation threshold.

According to the point density of the test datasets, the block size and the voxel size were set as $W_b = 20m$, $W_v = 0.5m$, respectively, to confirm the time complexity of the algorithm. Since the algorithm was conducted to remove roadside objects and preserve road curbs, the thresholds were set as $T_g = 3m$, $T_l = 0.3m$, respectively. Figure 4-1 (a) to (d) presents the ground point detection results.



(a) (a1)

(b) (b1)

(c) (c1)

(d) (d1)

Figure 4-1 Detected ground points (a) to (d) and road surface points (a1) to (d1)

4.1.2 Road Surface Detection

Road surface points were detected by the improved region growing algorithm. The input data were detected ground points and trajectory points. The algorithm firstly gridded the point clouds into voxels from x direction and y direction with a given width. Then, the BFS was conducted to detect and label the road surface voxels. Road curbs were detected based on elevation jump and slope. Five parameters were used in this algorithm:

W_R: size of a voxel,

T_{curb_max}, T_{curb_min}: elevation jump thresholds,

T_{slope}: slope threshold, and

T_B: searching range.

According to Guan et al. (2015a), $T_{curb\_min} = 8\ cm$, $T_{curb\_max} = 30\ cm$, and $T_{slope} = 2\pi/3$ were set. To determine the optimal parameters for W_R and T_B, and to evaluate the sensitivity of each of them, two experiments were carried out. The purpose of detecting road surface points is to improve the accuracy of the road marking extraction. Thus, road markings extracted by different parameters were evaluated to determine the optimal parameters. Test Dataset 1 was used for the experiments, since it has different and exhaustive features described in Section 3.1.2.

Table 4-1 Accuracy of road marking extraction when applying different searching ranges

Searching Range (m)	5	10	15	20	25	30	35
F_1-score (%)	86.20	90.83	93.37	93.99	94.46	94.34	94.23
Recall (%)	82.15	89.64	94.83	95.88	96.60	96.55	96.59
Precision (%)	90.68	92.05	91.96	92.17	92.42	92.23	91.98

Table 4-1 presents the F$_1$-score, recall, and precision of road marking extraction results using different searching ranges, with recall and precision recorded as well. A fixed voxel size $W_R = 10$ cm was used to explore the influences of searching ranges. Then, the road surface detection was carried out with different searching ranges, which were from 5 m to 35 m at an interval of 5 m. Generally, the F$_1$-score exponentially improved as the searching range increased from 5 m (86.20%) to 25 m (94.46%), and then slightly dropped from 25 m (94.46%) to 35 m (94.23%). The improvement of the F$_1$-score from 5 m to 25 m was caused by the increase of recall. Since the BFS searching started from the trajectory, the road surface was not completely detected when the searching range was small, and road markings close to the boundaries of the road surface were not extracted. However, when the searching range was overlarge, outliers on the pavements might be involved in road surface detection results. Accordingly, searching range T_B was set as 25 m for the BFS searching in road surface extraction.

Table 4-2 Accuracy of road marking extraction when applying different voxel sizes

Voxel Size (cm)	6	8	10	12	14	16
F$_1$-score (%)	93.18	93.98	94.46	93.29	91.59	89.66
Recall (%)	96.89	97.01	96.60	93.52	90.33	86.62
Precision (%)	89.74	91.13	92.42	93.06	92.89	92.93

Table 4-2 presents the F$_1$-score of road marking extraction results using different voxel sizes, with recall and precision recorded simultaneously. The searching range $T_B = 25$ m was fixed to explore the influences of the voxel size, which was from 6 cm to 16 cm at an interval of 2 cm. Generally, the F$_1$-score slightly increased when the voxel size increased from 6 cm (93.18%) to 10 cm (94.46%), and then significantly decreased when the voxel size was larger than 10 cm. When the voxel size was overlarge, road markings close to the boundary of the road surface were involved in the same voxel as the road curb and removed, which caused a significant loss in recall. However, more points were covered with larger voxels, which made the calculation of elevation jump and slope more accurate. Hence, $W_R = 10$ cm was optimal for the road surface detection in this study.

With the defined parameters, the detected road surface points in Test Dataset 1 to 4 are presented in Figure 4-1 (a1) to (d1). The overall performance of the improved algorithm was promising in the four test datasets. One false detection occurred in Test Datasets 1 and 2 (Figure 4-1 (a1) and (b1)), as part of the road pavement was labeled as road surface. The road surface detection method was sensitive to the condition of road curbs; thus, the defects of road curbs might cause a large area of false detection. Furthermore, a detailed examination of a detected road surface is shown in Figure 4-2, and three features of the algorithm can be concluded from it. First, all road markings can be detected on the extracted road surfaces. Second, the boundaries of road surfaces are not smoothed due to the cubic shape of voxels. Third, small holes might exist due to noise on road surfaces.

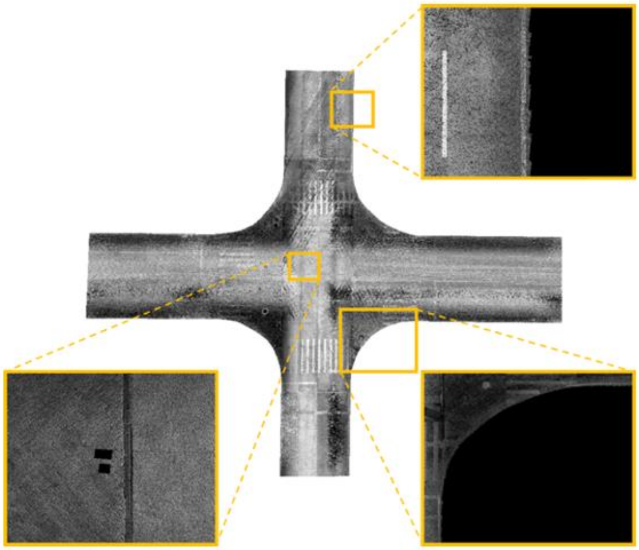
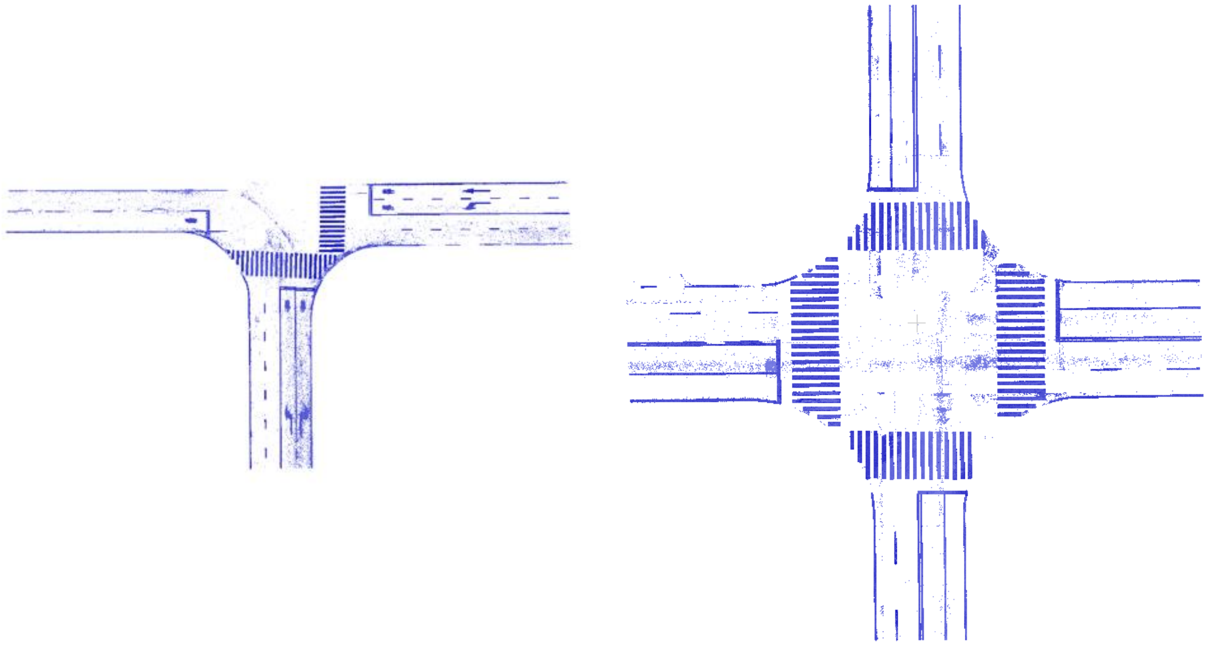


Figure 4-2 Details of a road surface detection result

4.1.3 Road Marking Extraction and Refinement

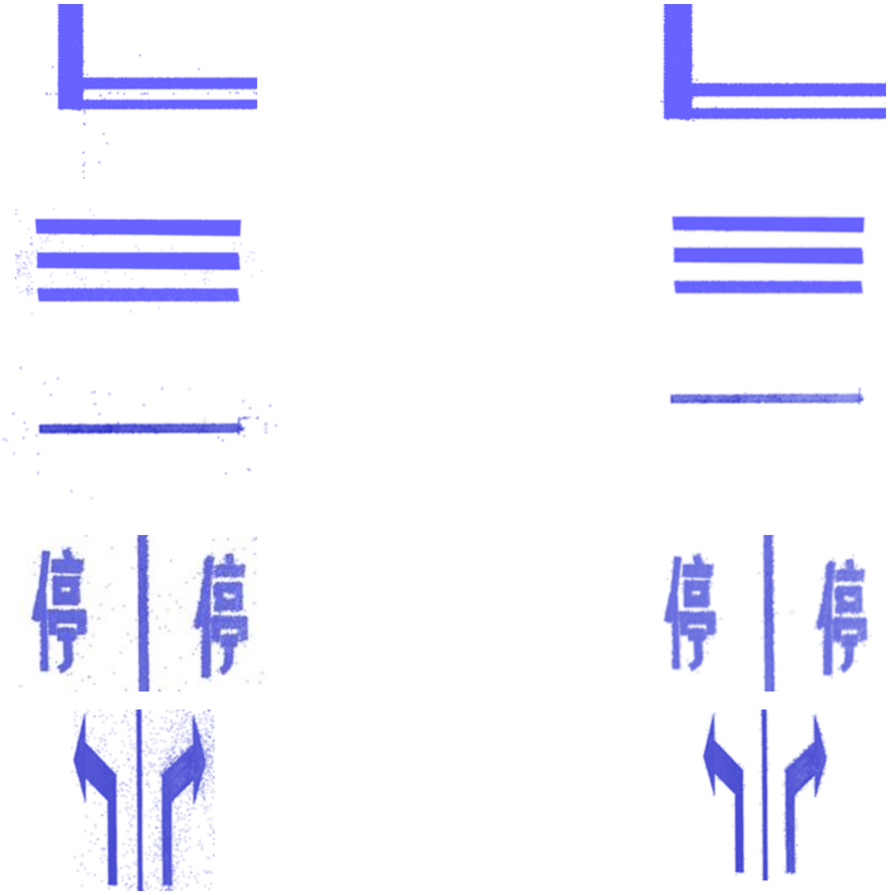
Road marking points were extracted from road surface points, using the multi-thresholding algorithm. Road surface points were segmented into point sets centered at the trajectory data with the threshold $w_s = 1m$. The intensity threshold for each point set was determined by Otsu's thresholding algorithm. As shown in Figure 4-3, the method completely extracted road markings; however, the overall quality of the extracted points was not satisfactory due to noise. Thus, to simultaneously eliminate noise and preserve the recall of road markings, the density filtering threshold $T_{SD} = 8$ was set for the test datasets. Samples of the refined road markings are shown in Figure 4-4.



(a) Candidate road marking points in Test Dataset 1

(b) Candidate road marking points in Test Dataset 3

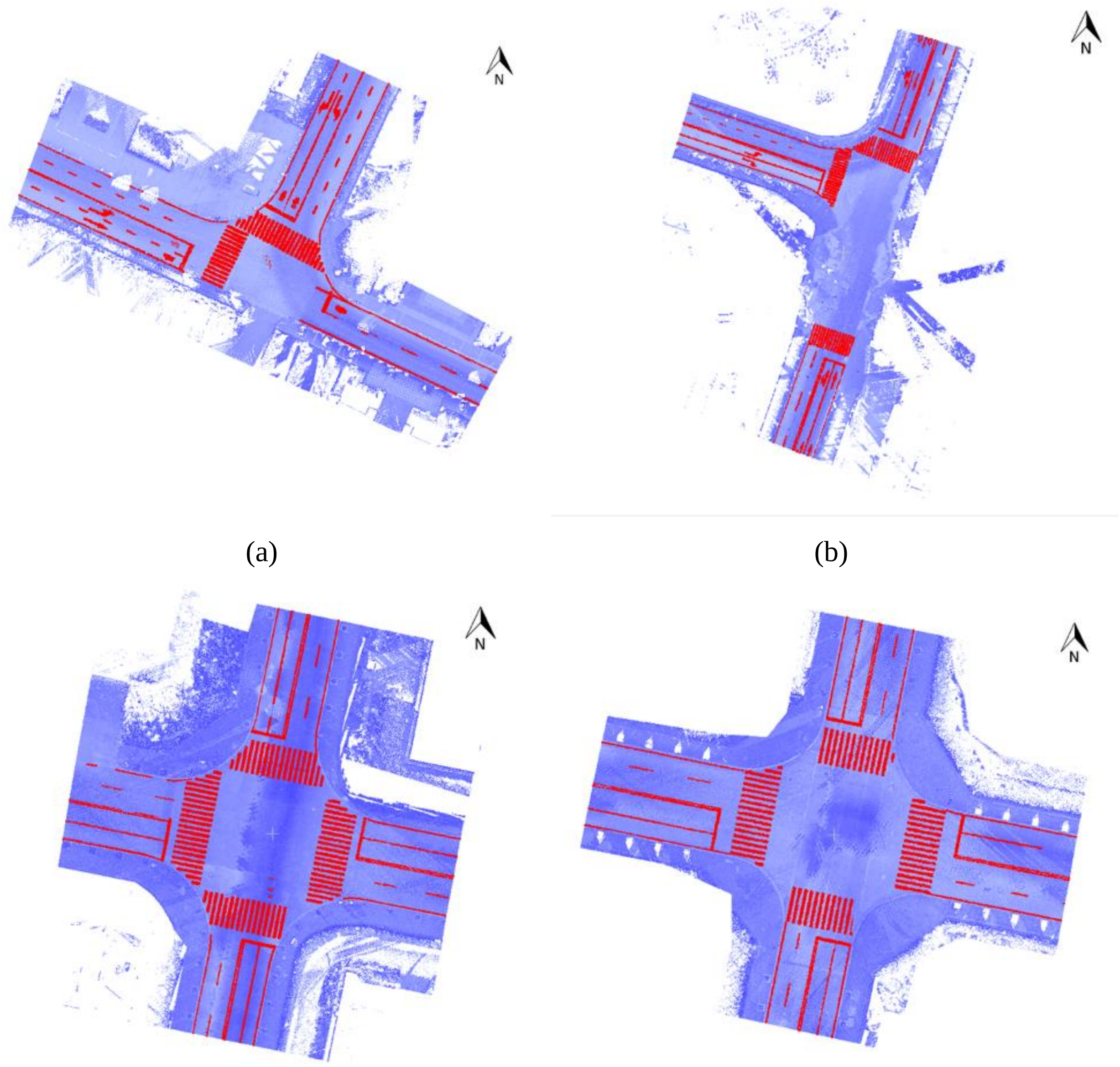
Figure 4-3 Candidate road marking points in Test Datasets 1 and 3



(a) Extracted road markings (b) Road markings after noise removal

Figure 4-4 Extracted road markings after noise removal

Road marking extraction results are shown in Figure 4-5. Generally, road markings are painted using a material that has high reflectance to laser pulses, and road marking points have higher intensities than the surrounding road surface points. However, due to the decay of road markings and obstacles on roads, some road marking points might have low intensities, and some road markings could be broken. In such cases, errors might occur in road marking extraction. Figure 4-6 presents a critical error that occurred in the results for road marking extraction in Test Dataset 2. Road features are indicated by variations in color intensity. The road marking close to the road boundary was covered by a material that had low reflectance to laser pulses, or it might be completely decayed. Since its intensity was lower than that of the road surface, the road marking was not extracted from the road surface. Such defects of the road marking could lead to a failure in lane structure generation, thus causing errors in transition line generation.



(a)

(b)

(c)

(d)

Figure 4-5 Extracted road markings after refining

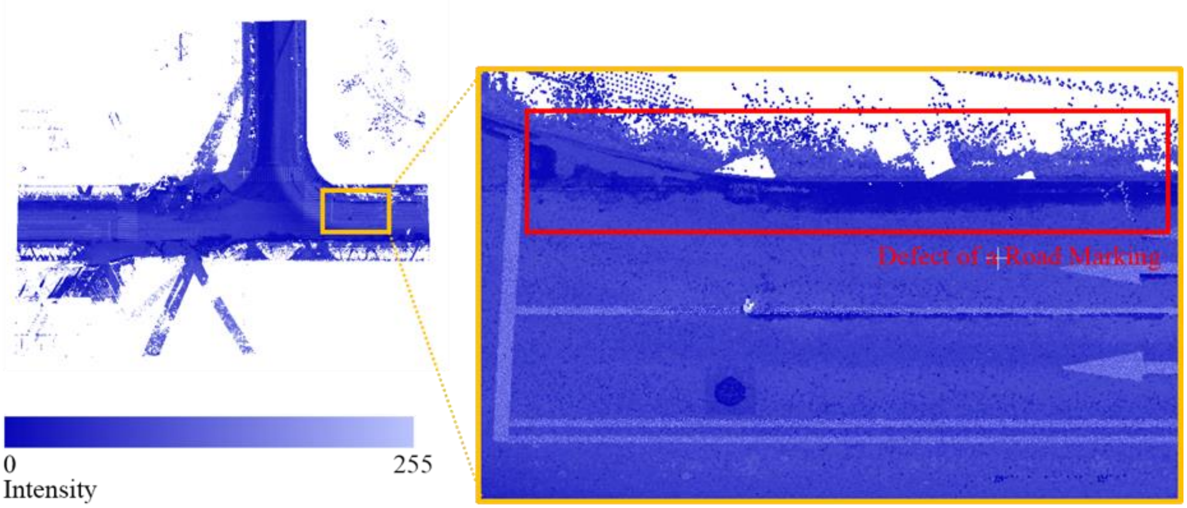
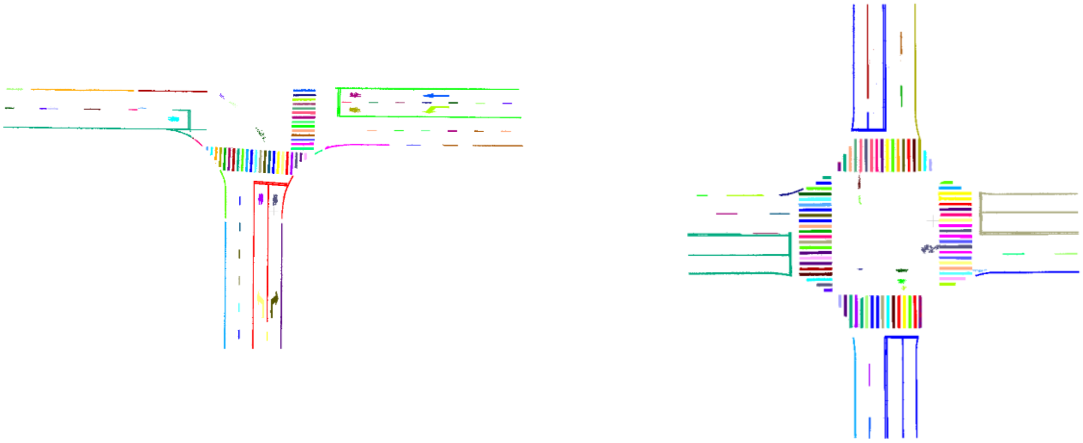


Figure 4-6 Defect of the road marking in Test Dataset 2

4.1.4 Lane Marking Extraction

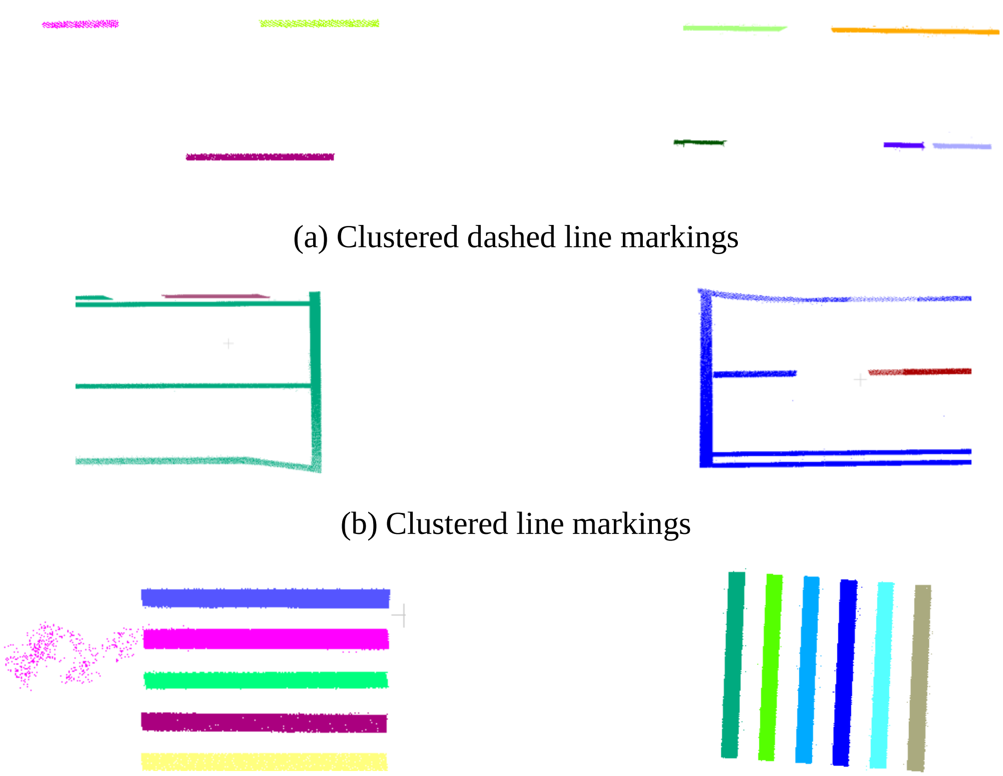
The extracted road marking points still had no topological relationships. To build such relationships among disordered points and to apply semantic meanings, the Conditional Euclidean clustering algorithm was employed to cluster road marking points. According to the point densities of test datasets, the searching range d_T was set at 0.3 cm to ensure that at least one point was within the searching range. Moreover, to generate every single zebra crossing line as a separate cluster, the d_T should be set to be less than the interval between two zebra crossing lines. Thus, $d_T = 0.1\ m$ was set for the method. The clustering results of Test Dataset 1 and 3 are shown in Figure 4-7 with clusters colored differently. In addition, as shown in Figure 4-8, points belonging to dashed line markings, single line markings, turning line markings, turning arrows, and Chinese characters are correctly clustered, whereas line markings connected by stop lines cannot be clustered separately.



(a) Clustered road markings in Test Dataset 1

(b) Clustered road markings in Test Dataset 3

Figure 4-7 Clustering results of road markings



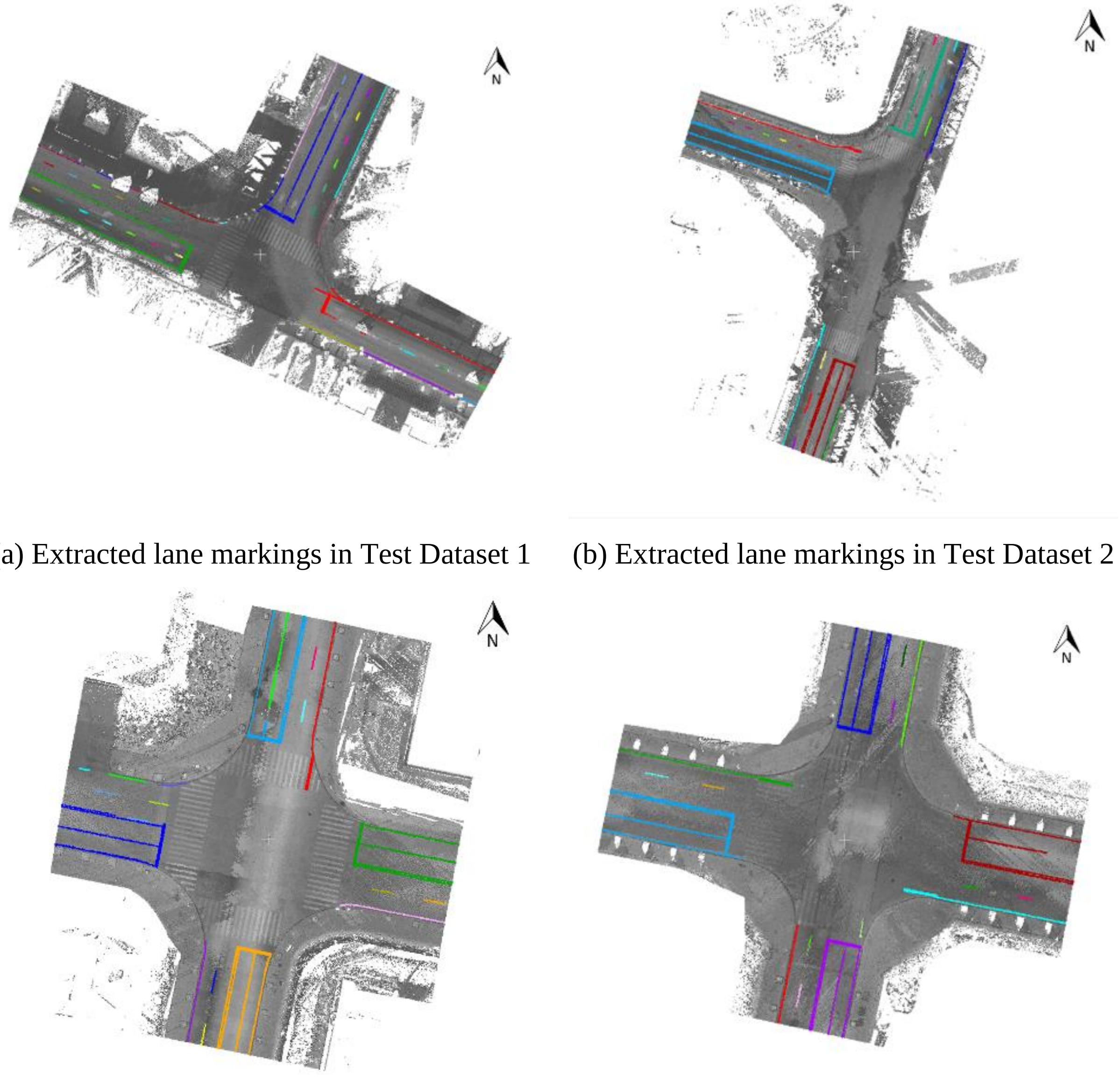
(a) Clustered dashed line markings

(b) Clustered line markings

(c) Clustered zebra crossing line markings

(d) Clustered Chinese characters and arrow markings

Figure 4-8 Clustering results of different road markings



(a) Extracted lane markings in Test Dataset 1

(b) Extracted lane markings in Test Dataset 2

(c) Extracted lane markings in Test Dataset 3

(d) Extracted lane markings in Test Dataset 4

Figure 4-9 Lane markings extracted from test datasets

Lane markings, including dashed lines, single lines, double lines, and stop lines, were extracted using the geometric feature filtering algorithm. The algorithm derived minimum bounding rectangles to extract length L and width W of clustered road markings. The filtering was conducted using the L and W range presented in Table 3-3. Most of the lane markings were preserved in the results (see Figure 4-9), whereas all turning arrows, zebra crossing lines, and Chinese characters were removed. However, as shown in Figures 4-9 (b), (c), and (d), several zebra crossing lines were also preserved, because they were close to nearby lane markings. Moreover, some of the broken lane marking segments were falsely removed due to their small size, which implied that lane markings might not be extracted when they are mostly blocked by obstacles.

4.1.5 Transition Line Generation

Lane node structures were generated by the proposed method. The resampling interval I_T was determined by the corresponding rate of the navigation system used in the MLS system and the driving speed. In this study, the corresponding rate of the GNSS system was 0.01s, and the maximum driving speed was 40 km/h. Thus, the interval of trajectory points was about 8.3 cm. Note that the over-segmentation of the lane markings might aggravate the influence of noise; the resampling interval $I_T = 6$ was defined for the test datasets. The width of a point block was about 50 cm. Additionally, the following three thresholds were used in this algorithm:

T_M, threshold to detect stop lines,

$T_{Rmin}\ and\ T_{Rmax}$, thresholds for node filtering.

The thresholds were defined according to the minimum lane width in the study area, which was 3 m. When points in a data block were projected onto the transformed coordinate system, if the length of a clustered road marking was greater than $T_M = 2m$, the marking was labelled as a stop line. To eliminate false lane centerline nodes, T_{Rmin} was set as 2 m and T_{Rmax} was set as 4 m. Figure 4-10 presents lane node structures in a road segment, with lane marking nodes in blue and lane centerline nodes in red.

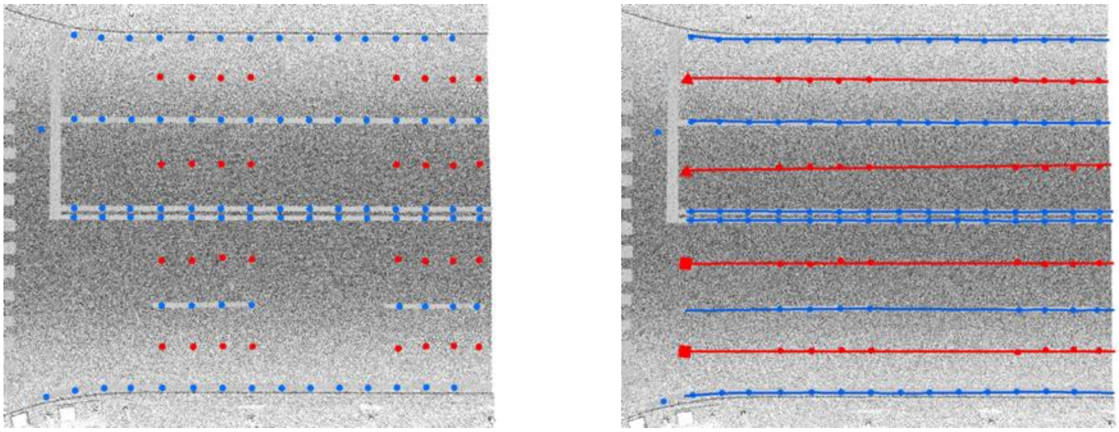
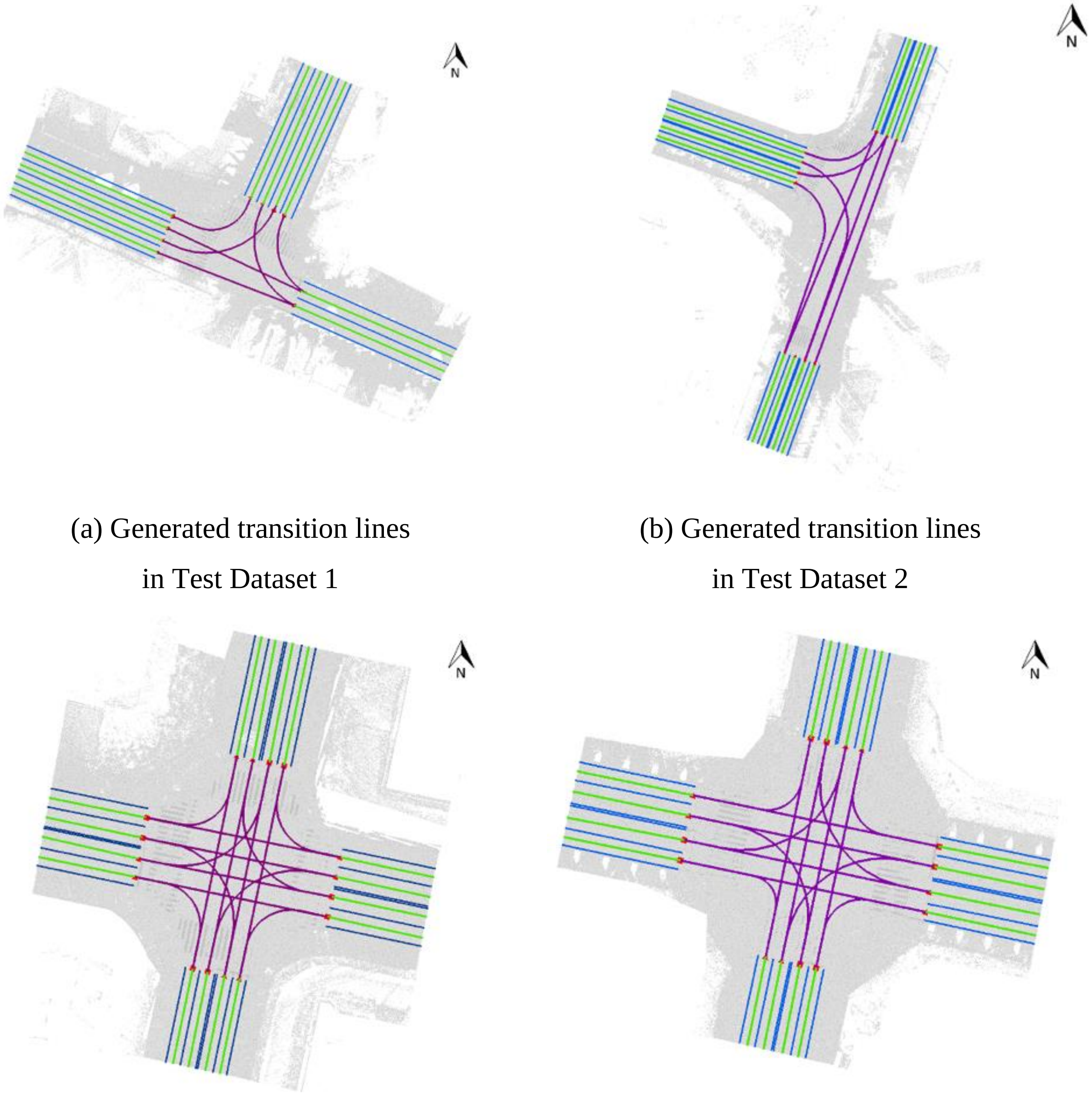


Figure 4-10 Lane node structure generation results in a road segment

Transition lines were generated by the cubic Catmull-Rom spline algorithm. As shown in Figure 4-11, test datasets are overlapped by the generated lane centerlines (green), lane markings (blue), and transition lines (purple). ‘Entry’ nodes of lane centerlines are square dots, and ‘exit’ nodes are triangle dots. In the four test datasets, most corresponding ‘entry’ and ‘exit’ nodes were correctly connected by transition lines, except that two transition lines connected wrong pairs of ‘exit’ and ‘entry’ nodes in Test Dataset 2 (Figure 4-11 (b)). Thus, the success rate of transition line generation was about 96.5%, again suggesting that road marking defects had negative effects on transition line generation.



(a) Generated transition lines in Test Dataset 1

(b) Generated transition lines in Test Dataset 2

(c) Generated transition lines in Test Dataset 3

(d) Generated transition lines in Test Dataset 4

Figure 4-11 Generated transition lines in the test datasets

HDRM generation is a post-processing process. To speed it up, automation of the process is critical. Therefore, the computing time of the proposed method in four test datasets is shown in Table 4-3. The consuming time was calculated in each procedure and in total. The computer used in this study had an AMD FX-8350 4.2MHz CPU, and 16 GB RAM. The proposed method performed very well considering the large-volume point cloud data. The voxel-based upward-growing and the improved region growing algorithms efficiently removed outliers from the test datasets. Road marking extraction and refinement procedure contributed most of the processing time because the multi-thresholding algorithm exhaustively processed every single point of the input data. By taking advantage of the Conditional Euclidean clustering algorithm, the lane marking extraction procedure did not consume much time. Furthermore, the lane marking node structure generation algorithm and the cubic Catmull-Rom spline algorithm were both efficient. Therefore, the overall efficiency of the proposed method could be further improved by modifying the multi-thresholding algorithm and the density filtering algorithm.

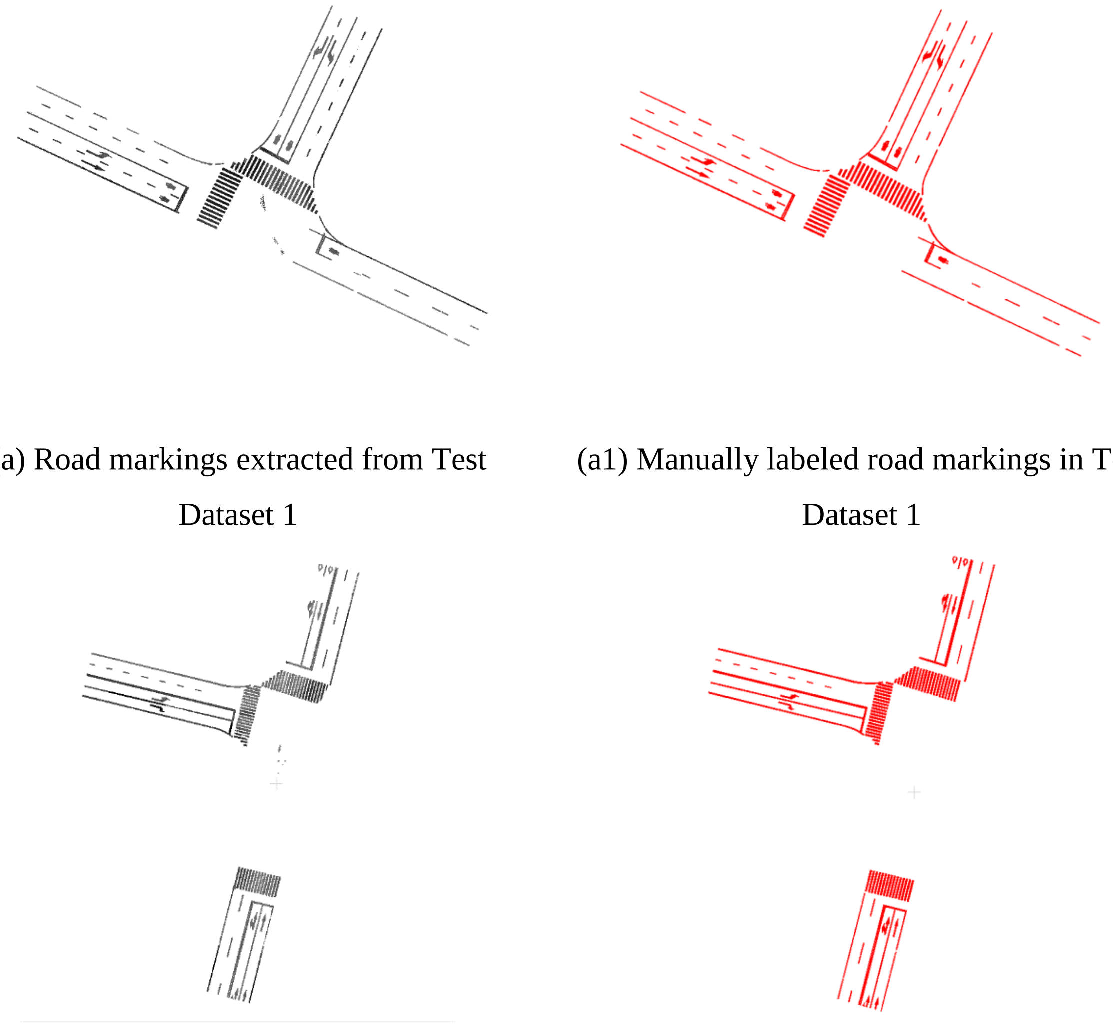
Table 4-3 Computing time of the proposed method

Dataset	**1**	**2**	**3**	**4**
Size	565MB	521MB	905MB	623MB
No. of MLS Points	19,243,445	25,537,193	17,444,202	16,094,181
No. of Trajectory Points	13748	25982	12946	16973
Time for Ground Point Detection (s)	13.1s	21.5s	12.8s	12.3s
Time for Road Surface Point Detection (s)	29.3s	35.3s	50.4s	53.9s
Time for Road Marking Extraction and Refinement (s)	309.2s	347.1s	456.9s	424.2s
Time for Lane Marking Extraction (s)	105.5s	126.5s	185.7s	193.2s
Time for Transition Line Generation (s)	15.5s	17.1s	30.6s	30.1s
Total Time (s)	472.6s	547.5s	736.4s	713.7s

4.2 Accuracy Assessment

4.2.1 Accuracy Assessment of Road Markings

Accuracy assessment of road markings was conducted by comparing the extracted road marking points from the test datasets with the manually interpreted road markings in the raw point clouds. The precision, recall, and F_1-score were calculated to evaluate the road marking extraction results.

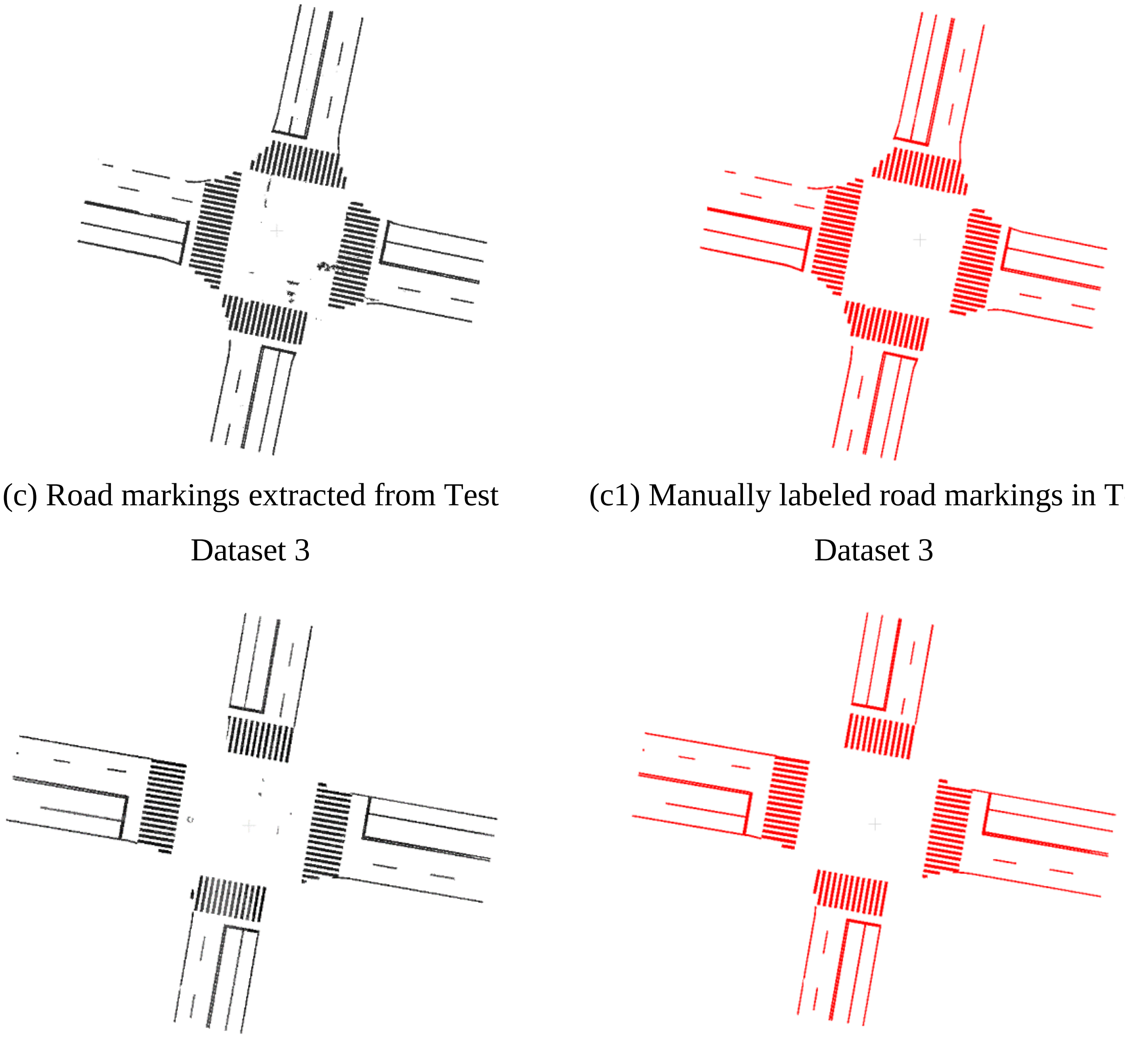


(a) Road markings extracted from Test Dataset 1

(a1) Manually labeled road markings in Test Dataset 1

(b) Road markings extracted from Test Dataset 2

(b1) Manually labeled road markings in Test Dataset 2



(c) Road markings extracted from Test Dataset 3

(c1) Manually labeled road markings in Test Dataset 3

(d) Road markings extracted from Test Dataset 4

(d1) Manually labeled road markings in Test Dataset 4

Figure 4-12 Extracted road markings and reference data

Table 4-4 Accuracy assessment of road markings using manually interpreted point clouds

Test Dataset	1	2	3	4
Precision (%)	92.42	88.67	91.71	90.38
Recall (%)	96.60	83.54	96.28	91.84
F_1-score (%)	94.46	86.23	93.94	91.10

Extracted road marking points and manually interpreted road marking points are shown in Figure 12 (a) to (d) and Figure (a1) to (d1), respectively. The results of road marking accuracy assessment using manually interpreted point clouds are listed in Table 4-4. The average precision of the test datasets was 90.80%. Generally, the loss in precision has two causes. On the one hand, false clustering might occur when zebra crossing lines and lane lines are connected. On the other hand, the density filter cannot remove all noise from candidate road marking points. The average recall of the four test datasets reached 92.07%. The loss in recall might be caused by the removal of small road marking segments when conducting the geometric feature filtering. The F_1-score of the four test datasets were all over 86%, which implied that the road marking extraction method can extract most of the road marking points in T- and cross-intersections. The proposed method performed well in Test Datasets 1, 3, and 4, whereas Test Datasets 2 contributed more to the loss of accuracy due to the broken road markings. Thus, the method should be improved in the future, to extract complete road markings in complex road intersection scenes.

4.2.2 Accuracy Assessment of Transition Lines and Lane Centerlines

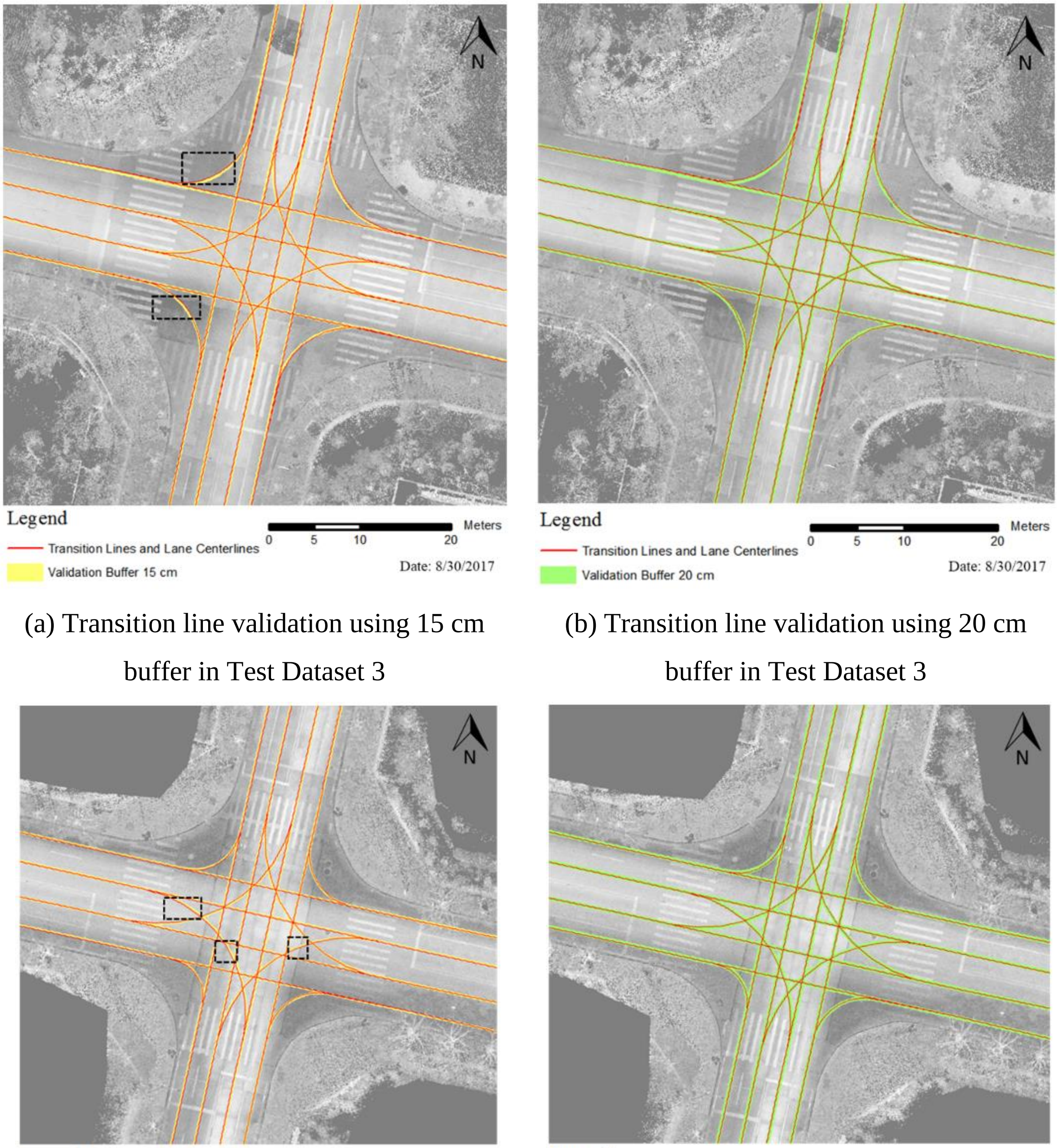
To evaluate the localization accuracy of the generated transition lines, the BOS method was applied. GRF images that were generated from raw point clouds were utilized as reference data. The gray values of pixels on the GRF images were interpolated from the intensities of raw point clouds using the IDW interpolation method (Guan et al., 2014). The resolution of the GRF images was set to be 2 cm. A detailed description of GRF image generation can be found in Guan’s (2014) study. Reference transition lines and lane centerlines were manually interpreted on the GRF images using ArcGIS v10.2.2. Centered by the manually labeled transition lines and lane centerlines, reference buffer zones with different widths, which indicate the range between the buffer zone boundaries and the manually interpreted transition lines and lane centerlines, were created and overlapped with the generated transition lines and lane centerlines.

The calculated values in both recall and miscoding of the generated transition lines and lane centerlines from Test Datasets 3 and 4 are listed in Tables 4-5. Reference buffers with the width of 15 cm and 20 cm were created to evaluate the performance of the proposed method. When conducting 15 cm BOS to the lines, the recall of Test Dataset 3 and 4 achieved 91.83% and 90.98%, respectively, and the miscoding was 7.01% and 7.82%, respectively. When conducting 20 cm BOS to the lines, the proposed method was capable of achieving 100% recall and 0% miscoding in Test Dataset 3 and 4. With the increased width of reference buffers, the miscoding values decreased, which demonstrated that most of the generated transition lines and lane centerlines were located within the precision allowable reference buffers. Therefore, the proposed method can provide a 20 cm-level localization accuracy to ensure the safety of autonomous vehicles and the precision of HDRMs.

Table 4-5 Recall and miscoding of Test Datasets 3 and 4.

	Test Dataset 3		Test Dataset 4	
Buffer Zoning (cm)	15	20	15	20
Recall (%)	91.83	100	90.98	100
Miscoding (%)	7.01	0	7.82	0

In addition, Figure 4-13 present the generated transition lines and lane centerlines in Test Dataset 3 and Test Dataset 4 within 15 cm and 20 cm width of reference buffers. The black rectangles in Figure 4-13 (a) and (c) indicate the miscoding parts in Test dataset 3 and Test Dataset 4 when the width of reference buffers was set as 15 cm. Furthermore, it is identified that generated transition lines and lane centerlines are completely located within the reference buffers with 20 cm in width.


N
Legend
Transition Lines and Lane Centerlines
🟨 Validation Buffer 15 cm
🟩 Validation Buffer 20 cm
Meters
0 5 10 20
Date: 8/30/2017

(a) Transition line validation using 15 cm buffer in Test Dataset 3

(b) Transition line validation using 20 cm buffer in Test Dataset 3

(c) Transition line validation using 15 cm buffer in Test Dataset 4

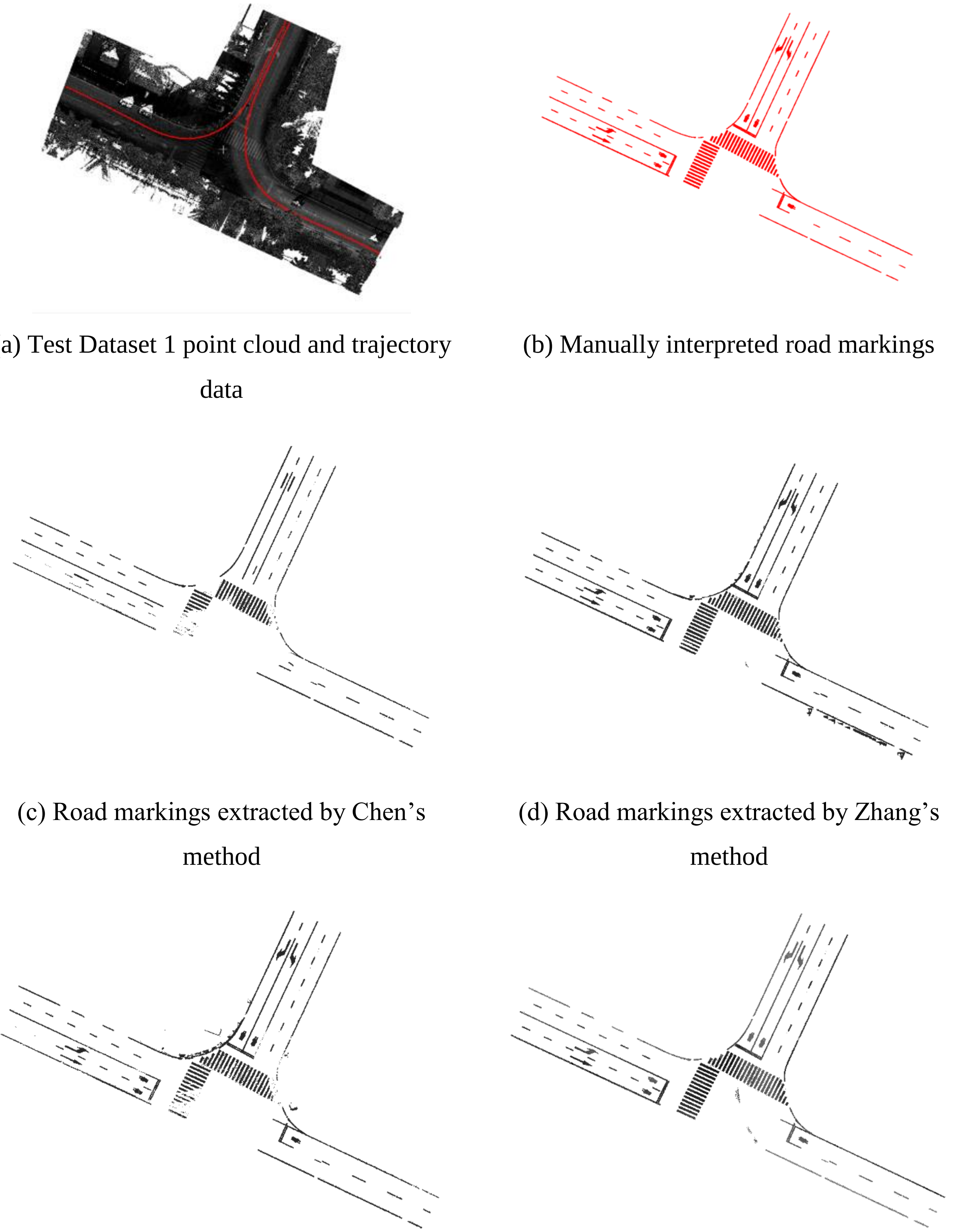
(d) Transition line validation using 20 cm buffer in Test Dataset 4

Figure 4-13 Generated transition lines and lane centerlines within reference buffers.

4.3 Comparative Study

Since road marking information was essential for transition line generation, a comparative study was conducted to compare the road marking extraction method used in this study with three other methods: Yu et al. (2015a), Zhang (2016), and Chen et al. (2009). Chen et al. (2009) conducted statistical analysis to select the intensity peaks as thresholds. The thresholds were then used to extract road markings that had a similar direction to the trajectory. Yu et al. (2015a) detected road surface points through the curb based segmenting. Then, the start point and the end point of the trajectory were used to partition the road surface into blocks, and road marking points were extracted from each block through Otsu's thresholding. In Zhang's method (Zhang, 2016), MLS point clouds were transformed to Geo-Referenced Feature (GRF) images, and road markings were detected through image processing. The high-pass filtering was firstly applied to the DTM of MLS point clouds to detect the road surfaces. Then, a GRF image was generated from the road surface pixels, and intensity deduction was corrected by the large-size high-pass enhancement. Finally, the road marking pixels were detected by applying Otsu's thresholding to the whole GRF image.

The methods were tested in Test Dataset 1 and Test Dataset 3. These two test datasets were selected for this comparative study for two reasons. First, Test Dataset 1 and Test Dataset 3 covered all road marking types in the four test datasets. Second, the trajectories in Test Dataset 1 (T-intersection) were curved (Figure 4-14 (a)), while the trajectories in Test Dataset 3 (cross-intersection) were straight (Figure 4-15 (a)). The results of the comparative study are shown in Figure 4-14 and Figure 4-15. In addition, based on point by point comparison with the manually interpreted road marking points (Figure 4-14 (b) and Figure 4-15 (b)), the performance of the methods was evaluated with precision, recall, and F_1-score (Table 4-6).



(a) Test Dataset 1 point cloud and trajectory data

(b) Manually interpreted road markings

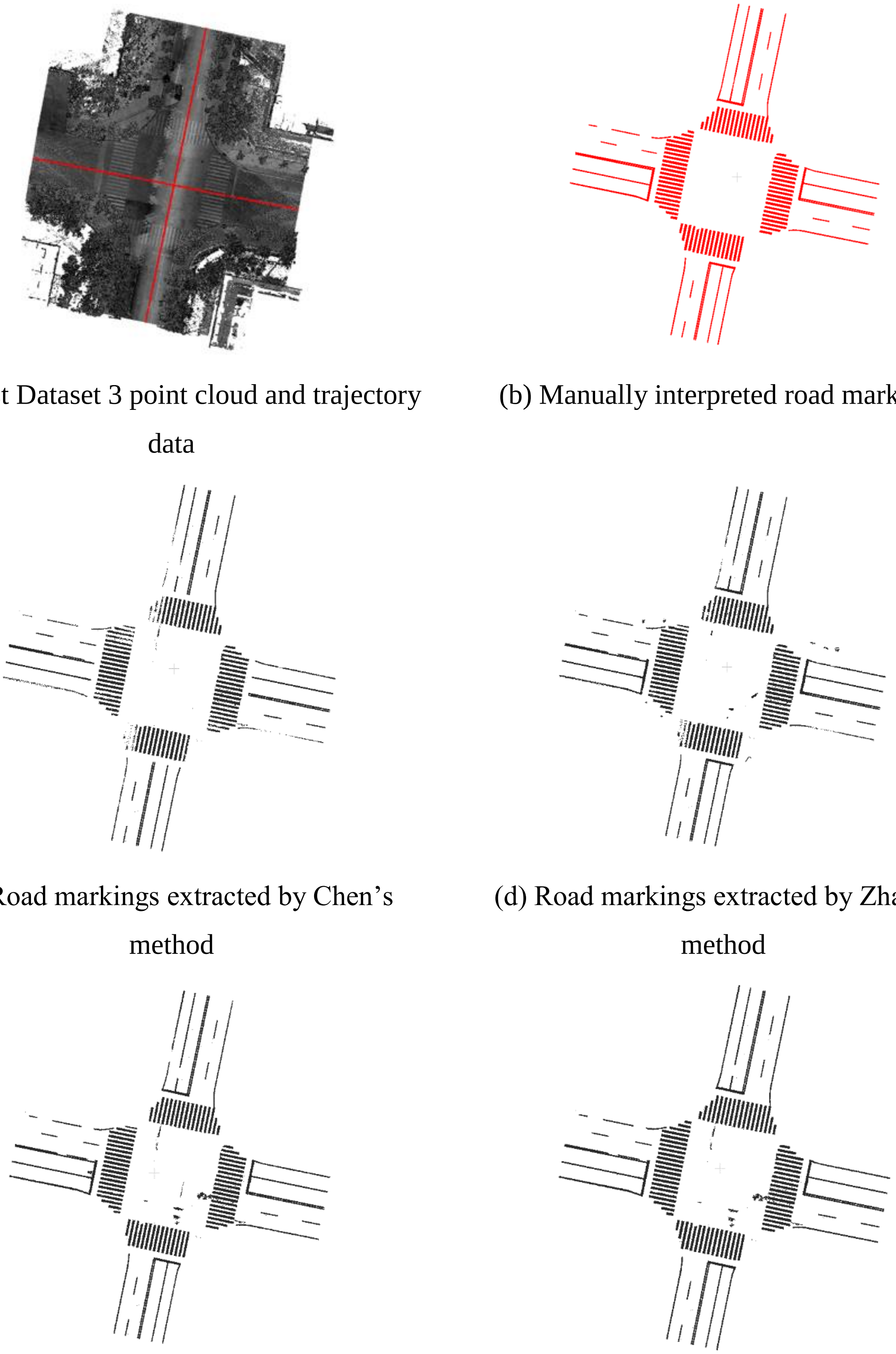
(c) Road markings extracted by Chen's method

(d) Road markings extracted by Zhang's method

(e) Road markings extracted by Yu's method

(f) Road markings extracted by the proposed method

Figure 4-14 Extracted road markings in Test Dataset 1



(a) Test Dataset 3 point cloud and trajectory data

(b) Manually interpreted road markings

(c) Road markings extracted by Chen's method

(d) Road markings extracted by Zhang's method

(e) Road markings extracted by Yu's method

(f) Road markings extracted by the proposed method

Figure 4-15 Extracted road markings in Test Dataset 3

Table 4-6 Comparison with different road marking extraction methods

Dataset	1				3			
Method	Chen et al. (2009)	Zhang (2016)	Yu et al. (2015a)	Proposed Method	Chen et al. (2009)	Zhang (2016)	Yu et al. (2015a)	Proposed Method
Recall (%)	68.90	91.38	80.71	92.42	77.45	90.57	91.66	91.71
Precision (%)	92.11	82.95	83.25	96.60	90.76	87.53	94.39	96.28
F_1-score (%)	78.83	86.96	81.96	94.46	83.58	89.02	93.00	93.94

Chen's method (Figure 4-14 (c) and 4-15 (c)) focused only on the lane markings along the direction of the vehicle trajectory. Thus, road markings, including turning arrows, stop lines, and Chinese characters, were not extracted, and that caused a signification loss of recall. Yu's method (Figure 4-14 (e) and 4-15 (e)) had low F_1-score in Test Dataset 1, since its road surface detection method and road marking extraction method are both based on the straight trajectory. When the method was applied in Test Dataset 1, road pavements that were close to the central part of the intersection were misclassified as road surface, and the road surface was not segmented correctly. That caused a loss of F_1-score in the result for Test Dataset 1. Zhang's method detected most of the road markings in the two test datasets (Figure 4-14 (d) and 4-15 (d)). However, road pavements that were smooth and had large areas were not eliminated by the high-pass filtering. As a result, road pavement points that have high intensities were misclassified as road markings. In addition, Zhang transformed MLS point clouds to geo-referenced images and detected road marking pixels on the images. Labeling road marking points through the images also caused a loss of precision.

The proposed method successfully extracted most of the road marking points in Test Dataset 1 and 3 (Figure 4-14 (f) and 4-15 (f)). Unlike Chen's method, it had the ability to extract all types of road markings on road surfaces. Unlike Zhang's geo-referenced feature image based method, it directly extracted road marking points from MLS point clouds and did not cause accuracy losses due to data transformation. Moreover, it performed better in road surface detection;

thus, noises caused by road pavement points were rare. Compared with Yu's method, the proposed method made an over 10% improvement in F_1-score when trajectories were curved in Test Dataset 1. However, some errors still occurred in the results. The errors in the extracted road markings were mainly caused by road marking decay and constant road marking extraction thresholds. Although road marking extraction thresholds were separately determined in each data block, intensity deduction still happened in the data blocks. As a result, road marking points that had low intensities due to the road marking decay might not be extracted. Simultaneously, road surface points that had high intensities could be falsely extracted as road marking points. In addition, noises that were close to the road markings were not eliminated by the density filtering, thus reducing precision.

4.4 Chapter Summary

This chapter presents the stepwise experimental results of the proposed method. According to the accuracy assessment and the comparative study of road marking extraction, it can be concluded that the proposed road marking extraction method can achieve highly accurate results, with 90.80% in precision, 92.07% in recall, and 91.43% in F_1-score, respectively. Furthermore, based on the accuracy assessment of generated transition lines and lane centerlines, it indicates that the proposed method can generate transition lines and lane centerlines with 20 cm-level positioning accuracy in road intersections. However, the accuracy of the proposed method may be influenced by defects in extracted lane markings.

This chapter also analyzes the computational efficiency of the proposed method. The proposed algorithms for road surface detection and node structure generation are highly efficient; however, the total efficiency of the proposed method could be further improved in the future by modifying the algorithms for road marking point detection and refinement.

Chapter 5 Conclusions and Recommendations

5.1 Conclusions

MLS systems have enormous potential in the construction of HDRMs because of the following advantages: flexibility in collecting time, complete coverage of street scenes, and 3D high resolution in point clouds. However, the unique data structure and the high density of point clouds makes developing advanced methods for the generation of road maps from the MLS data challenging. Currently, road surface information such as road markings and road cracks can be extracted from the MLS point clouds, whereas transition lines in road intersections are mainly generated from public GPS trajectory data. In addition, nowadays, there is still no commercial software which specializes in modeling and interpreting MLS point clouds for transportation related applications. Therefore, the construction of HDRMs requires new software and high-end workstations.

To provide a credible approach for HDRM construction, this study develops a semi-automated method for transition line generation in road intersections from MLS point clouds. Ground points are detected by the voxel-based upward-growing, followed by the curb-based region-growing for road surface point detection. Then, candidate road marking points are extracted from road surface points using the multi-thresholding. The density filtering is employed for noise removal in candidate road marking points. Next, semantic clusters of road markings are segmented using the Conditional Euclidean clustering, and subsequently, lane markings are extracted with the geometric feature filtering. Moreover, lane centerlines are generated from the lane marking node structure, followed by the generation of transition lines through the Catmull-Rom spline.

Four test datasets have been used in this study to evaluate the performance of the proposed method. Assessment of the test datasets contains two parts: the accuracy assessment of extracted lane marking points and the accuracy assessment of generated lane centerlines and transition lines. In the validation of lane marking extraction using manually interpreted lane marking points, the method can achieve 90.80% precision, 92.07% recall, and 91.43% F_1-score, respectively. The success rate of transition line generation is 96.5%. Moreover, the BOS method validates that the proposed method can generate lane centerlines and transition lines within a 20-cm range of reference path. Detailed analysis of the experimental results demonstrates that the proposed method can successfully generate most of the transition lines, whereas defects of road markings could cause a loss in accuracy.

This thesis concludes that the proposed stepwise methodology can generate transition lines and lane centerlines in T-intersections and cross-intersections from MLS point clouds to provide highly accurate navigation and localization services for autonomous vehicles. It also provides a reliable solution to overcome the challenges in the generation of HDRMs for worldwide automotive manufacturers and mapping companies such as HERE, Google, TomTom, and Baidu.

5.2 Contributions

As stated in the objectives (Chapter 1), the transition line generation method proposed in this study mainly contributes to the construction of HDRMs using MLS point clouds. The detailed contributions are, namely, an improved curb-based region growing algorithm for road surface detection and a semi-automated node structure generation algorithm:

(1) The curb-based region growing algorithm detects road surface points by using trajectory points as seeds and then iteratively searching nearby areas. This algorithm is

feasible for different types of curbed urban roads, including straight roads, curved roads, and road intersections. Furthermore, it reduces accuracy loss by detecting road surface points directly from point clouds.

(2) A semi-automated node structure generation algorithm is presented. The generated lane centerline nodes facilitate the construction of transition lines. By utilizing lane marking point clouds as input data, it does not require GPS trajectory covers all lanes, and that improves the efficiency of the data collection in road intersections.

5.3 Limitations and Recommendations

Although the proposed method in generating transition lines in road intersections has provided satisfactory results, according to the Chapter 4, some limitations still exist. Therefore, limitations are clarified and recommendations are made to guide the future work. The details can be noted as follows:

(1) ‘Exit’ and ‘Entry’ nodes are paired mainly based on lane information and traffic rules in this study because turning arrow markings cannot be extracted through the geometric filtering. Thus, more works are still needed to make the algorithm more completed. Some approaches in computer vision, such as machine learning and deep learning, could be adopted to classify road markings in MLS point clouds.

(2) Defects of road markings have negative effects on the lane marking node structure generation algorithm due to the segmentation process. Thus, more works are still needed to improve it.

(3) The proposed method is only proved in small T- and cross-intersections. However, road marking standard and lane pattern could be more complex in large road intersections.

Moreover, point density and intensity is influenced by the scanning distance. Therefore, experiments in large road intersections are needed to test the application range of the proposed method.

(4) The efficiency of the proposed method could be furtherly improved. As mentioned in Chapter 4, the road marking extraction and refinement takes more than half of the total computational time. To reduce the time complexity of the algorithm, a more efficient approach for computing the distance from a point to a trajectory could be developed in the future.

(5) The generated lane centerlines and transition lines should be assessed by test driving. Although generated line features are tested and validated to be correct and complete through manual interpolation of UAV images, the safety and comfort level cannot be assessed. Thus, test driving should be conducted in the future to give a complete validation of transition lines.

References

Abraham, H., Lee, C., Brady, S., Fitzgerald, C., Mehler, B., Reimer, B., & Coughlin, J. F., 2017. Autonomous vehicles, trust, and driving alternatives: a survey of consumer preferences. In *Transportation Research Board 96th Annual Meeting, Washington, DC,* 8-12.

Barber, D., Mills, J., & Smith-Voysey, S., 2008. Geometric validation of a ground-based mobile laser scanning system. *ISPRS Journal of Photogrammetry and Remote Sensing, 63*(1), 128-141.

Bétaille, D., & Toledo-Moreo, R., 2010. Creating enhanced maps for lane-level vehicle navigation. *IEEE Transactions on Intelligent Transportation Systems, 11*(4), 786-798.

Bétaille, D., Toledo-Moreo, R., & Laneurit, J., 2008. Making an enhanced map for lane location based services. *In Intelligent Transportation Systems, 2008,* 711-716.

Buehler, M., Iagnemma, K., & Singh, S., 2009. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic,* Springer, 125-163.

Cabo, C., Kukko, A., García-Cortés, S., Kaartinen, H., Hyyppä, J., & Ordoñez, C., 2016. An algorithm for automatic road asphalt edge delineation from mobile laser scanner data using the line clouds concept. *Remote Sensing, 8*(9), 740-760.

Cao, L., & Krumm, J., 2009. From GPS traces to a routable road map. *In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems,* 3-12.

CATMULL, E., AND ROM, R., 1974. A class of local interpolating splines. In *Computer Aided Geometric Design*, R. E. Barnhill and R. F. Riesenfeld, Eds. Academic Press, Orlando, 317-326.

Chen, X., Stroila, M. and Wang, R., 2009. Next generation map marking: geo-referenced ground-level LiDAR point clouds for automatic retro-reflective road feature extraction. *17th ACM SIGSPATIAL International Conference on Advances in Geographic Information System*, Seattle, WA, USA, 4-6.

Cheng, M., Zhang, H., Wang, C., & Li, J., 2017. Extraction and classification of road markings using mobile laser scanning point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. 10*(3), 1182-1196.

Clode, S., Rottensteiner, F., Kootsookos, P., & Zelniker, E., 2007. Detection and vectorization of roads from lidar data. *Photogrammetric Engineering & Remote Sensing, 73*(5), 517-535.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2001. "33.3: Finding the convex hull". *Introduction to Algorithms* (2nd ed.). MIT Press and McGraw-Hill, 955–956.

Congalton, R., G., and Green, K., 2008. *Assessing the Accuracy of Remotely Sensed Data: Principles and Practics*. Boca Raton, FL: CRC Press, 55-59.

Danescu, R., & Nedevschi, S., 2009. Probabilistic lane tracking in difficult road scenarios using stereovision. *IEEE Transactions on Intelligent Transportation Systems, 10*(2), 272-282.

de Lima, D. A., & Victorino, A. C., 2016. A hybrid controller for vision-based navigation of autonomous vehicles in urban environments. *IEEE Transactions on Intelligent Transportation Systems*, *17*(8), 2310-2323.

DeRose, T. D., & Barsky, B. A., 1988. Geometric continuity, shape parameters, and geometric constructions for Catmull-Rom splines. *ACM Transactions on Graphics (TOG), 7*(1), 1-41.

Du, J., & Barth, M. J., 2008. Next-generation automated vehicle location systems: Positioning at the lane level. *IEEE Transactions on Intelligent Transportation Systems, 9*(1), 48-57.

Du, J., 2004. CAMP enhanced digital mapping project—Final report, U.S. Department of Transportation, Washington, DC. [Online]. Available: http://www.nhtsa.gov/DOT/ NHTSA/NRD/Multimedia/PDFs/Crash%20Avoidance/ 2004/FinalRept_111904.pdf

El-Sheimy, N., 2005. An overview of mobile mapping systems. In *FIG Working Week*, 16-21.

Fagnant, D. J., & Kockelman, K., 2015. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, *77*, 167-181.

Freeman, W. T., & Adelson, E. H., 1991. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(9), 891-906.

Gao, T., & Aghajan, H., 2009. Self lane assignment using egocentric smart mobile camera for intelligent GPS navigation. *In Computer Vision and Pattern Recognition Workshops, 2009,* 57-62.

Glennie, C., 2007. Rigorous 3D error analysis of kinematic scanning LIDAR systems. *Journal of Applied Geodesy, 1*(3), 147-157.

Glennie, C., & Lichti, D. D., 2010. Static calibration and analysis of the Velodyne HDL-64E S2 for high accuracy mobile scanning. *Remote Sensing, 2*(6), 1610-1624.

Guan, H., Li, J., Cao, S., & Yu, Y., 2016. Use of mobile LiDAR in road information inventory: A review. *International Journal of Image and Data Fusion, 7*(3), 219-242.

Guan, H., Li, J., Yu, Y., Chapman, M., & Wang, C., 2015a. Automated road information extraction from mobile laser scanning data. *IEEE Transactions on Intelligent Transportation Systems, 16*(1), 194-205.

Guan, H., Li, J., Yu, Y., Ji, Z., & Wang, C., 2015b. Using mobile LiDAR data for rapidly updating road markings. *IEEE Transactions on Intelligent Transportation Systems, 16*(5), 2457-2466.

Guan, H., Li, J., Yu, Y., Wang, C., Chapman, M., & Yang, B., 2014. Using mobile laser scanning data for automated extraction of road markings. *ISPRS Journal of Photogrammetry and Remote Sensing, 87*, 93-107.

Guan, H., Yu, Y., Li, J., & Liu, P., 2016. Pole-Like road object detection in mobile LiDAR data via supervoxel and bag-of-contextual-visual-words representation. *IEEE Geoscience and Remote Sensing Letters, 13*(4), 520-524.

Guo, C., Kidono, K., Meguro, J., Kojima, Y., Ogawa, M., & Naito, T., 2016. A low-cost solution for automatic lane-level map generation using conventional in-car sensors. *IEEE Transactions on Intelligent Transportation Systems, 17*(8), 2355-2366.

Gwon, G. P., Hur, W. S., Kim, S. W., & Seo, S. W., 2017. Generation of a precise and efficient lane-level road map for intelligent vehicle systems. *IEEE Transactions on Vehicular Technology, 66*(6), 4517-4533.

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.

Hervieu, A., & Soheilian, B., 2013. Semi-automatic road/pavement modeling using mobile laser scanning. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2,* 31-36.

Hillel, A. B., Lerner, R., Levi, D., & Raz, G., 2014. Recent progress in road and lane detection: a survey. *Machine Vision and Applications, 25*(3), 727-745.

Holgado-Barco, A., Riveiro, B., González-Aguilera, D., & Arias, P., 2017. Automatic inventory of road cross-sections from mobile laser scanning system. *Computer-Aided Civil and Infrastructure Engineering*, *32*(1), 3-17.

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

Jarvis, R. A., 1973. On the identification of the convex hull of a finite set of points in the plane. *Information processing letters, 2*(1), 18-21.

Jiang, Y., Gao, F., & Xu, G., 2010. Computer vision-based multiple-lane detection on straight road and in a curve. *In Image Analysis and Signal Processing, 2010,* 114-117.

Jwa, Y., and Sohn, G., 2012. A piecewise catenary curve model growing for 3d power-line reconstruction. *Photogrammetric Engineering and Remote Sensing, 78*(12): 1227-1240.

Kaasalainen, S., Jaakkola, A., Kaasalainen, M., Krooks, A., & Kukko, A., 2011. Analysis of incidence angle and distance effects on terrestrial laser scanner intensity: Search for correction methods. *Remote Sensing, 3*(10), 2207-2221.

Kheyrollahi, A., & Breckon, T. P., 2012. Automatic real-time road marking recognition using a feature driven approach. *Machine Vision and Applications, 23*(1), 123-133.

Kumar, P., McElhinney, C. P., Lewis, P., & McCarthy, T., 2014. Automated road markings extraction from mobile laser scanning data. *International Journal of Applied Earth Observation and Geoinformation*, *32*, 125-137.

Lemmens, M., 2011. *Geo-information: technologies, applications and the environment*. Springer Science & Business Media, 101-120.

Li, L., Zhang, D., Ying, S., & Li, Y., 2016. Recognition and Reconstruction of Zebra Crossings on Roads from Mobile Laser Scanning Data. *ISPRS International Journal of Geo-Information, 5*(7), 125-141.

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.

Lim, S., Thatcher, C. A., Brock, J. C., Kimbrow, D. R., Danielson, J. J., & Reynolds, B. J., 2013. Accuracy assessment of a mobile terrestrial lidar survey at Padre Island National Seashore. *International Journal of Remote Sensing, 34*(18), 6355-6366.

Liu, J., Cai, B., Wang, Y., & Wang, J., 2013a. Generating enhanced intersection maps for lane level vehicle positioning based applications. *Procedia-Social and Behavioral Sciences, 96*, 2395-2403.

Liu, Z., Wang, J., & Liu, D., 2013b. A new curb detection method for unmanned ground vehicles using 2D sequential laser data. *Sensors, 13*(1), 1102-1120.

Luettel, T., Himmelsbach, M., & Wuensche, H. J., 2012. Autonomous ground vehicles—Concepts and a path to the future. *Proceedings of the IEEE, 100*, 1831-1839.

Luo, H., Wang, C., Wen, C., Cai, Z., Chen, Z., Wang, H., ... & Li, J., 2016. Patch-based semantic labeling of road scene using colorized mobile lidar point clouds. *IEEE Transactions on Intelligent Transportation Systems, 17*(5), 1286-1297.

Marshall, G., F., 2011. *Hand Book of Optical and Laser Scanning*. CRC Press, 1-69.

Mathibela, B., Newman, P., & Posner, I., 2015. Reading the road: road marking classification and interpretation. *IEEE Transactions on Intelligent Transportation Systems, 16*(4), 2072-2081.

McElhinney, C. P., Kumar, P., Cahalane, C., & McCarthy, T., 2010. Initial results from European Road Safety Inspection (EURSI) mobile mapping project. *In ISPRS Commission V Technical Symposium*, 440-445.

Ministry of Housing and Urban-Rural Development of the People's Republic of China & General Administration of Quality Supervision, Inspection and Quarantine of the People's Republic of China, 2015. *Code for Layout of Urban Road Traffic Signs and Markings* (GB 51038-2015).

Moore, E. F., 1959. The shortest path through a maze. In *Proc. Int. Symp. Switching Theory, 1959*, 285-292.

National Highway Traffic Safety Administration., 2013. *Preliminary statement of policy concerning automated vehicles*. Washington, DC, 1-14.

Otsu, N., 1979. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics, 9*(1), 62-66.

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

RIEGL., 2017. RIEGL VQ-450, Retrieved from: http://www.riegl.com/nc/products/mobile-scanning/produktdetail/product/scanner/31.

Riveiro, B., González-Jorge, H., Martínez-Sánchez, J., Díaz-Vilariño, L., & Arias, P., 2015. Automatic detection of zebra crossings from mobile LiDAR data. *Optics & Laser Technology, 70*, 63-70.

Schreiber, M., Knöppel, C., & Franke, U., 2013. Laneloc: Lane marking based localization using highly accurate maps. In *2013 IEEE Intelligent Vehicles Symposium (IV),* 449-454.

Shi, W., Miao, Z., & Debayle, J., 2014. An integrated method for urban main-road centerline extraction from optical remotely sensed imagery. *IEEE Transactions on Geoscience and Remote Sensing*, *52*(6), 3359-3372.

Shujia, F., 2015. The Roadmap for Autonomous (Self-Driving) Vehicles in Ontario, Canada [White paper]. Retrieved July 2015, from Ontario Good Roads Association: https://www.ogra.org/files/Roadmap%20for%20AVs%20in%20Ontario%20-%20White%20Paper.pdf.

Smadja, L., Ninot, J., & Gavrilovic, T., 2010. Road extraction and environment interpretation from LiDAR sensors. *IAPRS Archieves, 38*, 281-286.

Soilán, M., Riveiro, B., Martínez-Sánchez, J., & Arias, P., 2016. Traffic sign detection in MLS acquired point clouds for geometric and image-based semantic inventory. *ISPRS Journal of Photogrammetry and Remote Sensing*, *114*, 92-101.

Soilán, M., Riveiro, B., Martínez-Sánchez, J., & Arias, P., 2017. Segmentation and classification of road markings using MLS data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *123*, 94-103.

Sokolova, M., Japkowicz, N., & Szpakowicz, S., 2006. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. *In Australian Conference on Artificial Intelligence, 4304,* 1015-1021.

Talaya, J., Bosch, E., Alamús, R., Serra, A., & Baron, A., 2004. GEOVAN: The mobile mapping system from the ICC. In *Proceedings of the 4th International Symposium on Mobile Mapping Technology*, 1-7.

Thorpe, C., Herbert, M., Kanade, T., & Shafter, S., 1991. Toward autonomous driving: the CMU navlab. II. architecture and systems. *IEEE Expert*, *6*(4), 44-52.

Toledo-Moreo, R., Bétaille, D., & Peyret, F., 2010. Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps. *IEEE Transactions on Intelligent Transportation Systems*, *11*(1), 100-112.

Tveite, H., 1999. An accuracy assessment method for geographical line data sets based on buffering. *International Journal of Geographical Information Science, 13*(1), 27-47.

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

Van Rijsbergen, C. J., 1979. *Information Retrieval* (2nd ed.). Butterworth, 112-140.

Wang, L., 2016. Semi-automated Generation of High-accuracy Digital Terrain Models along Roads Using Mobile Laser Scanning Data. Retrieved from theses, University of Waterloo: https://uwspace.uwaterloo.ca/handle/10012/10276

Wang, H., Luo, H., Wen, C., Cheng, J., Li, P., Chen, Y., ... & Li, J., 2015. Road boundaries detection based on local normal saliency from mobile laser scanning data. *IEEE Geoscience and Remote Sensing Letters, 12*(10), 2085-2089.

Wegner, J. D., Montoya-Zegarra, J. A., & Schindler, K., 2013. A higher-order CRF model for road network extraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1698-1705.

Wen, C., Li, J., Luo, H., Yu, Y., Cai, Z., Wang, H., & Wang, C., 2016. Spatial-related traffic sign inspection for inventory purposes using mobile laser scanning data. *IEEE Transactions on Intelligent Transportation Systems*, *17*(1), 27-37.

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

Wu, S. J., Chiang, H. H., Perng, J. W., Chen, C. J., Wu, B. F., & Lee, T. T., 2008. The heterogeneous systems integration design and implementation for lane keeping on a vehicle. *IEEE Transactions on Intelligent Transportation Systems, 9*(2), 246-263.

Xu, S., Wang, R., & Zheng, H., 2017. Road Curb Extraction from Mobile LiDAR Point Clouds. *IEEE Transactions on Geoscience and Remote Sensing, 55*(2), 996-1009.

Yan, L., Liu, H., Tan, J., Li, Z., Xie, H., & Chen, C., 2016. Scan line based road marking extraction from mobile LiDAR point clouds. *Sensors, 16*(6), 903-924.

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., Fang, L., & Li, J., 2013. Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *79*, 80-93.

Yen, K. S., Ravani, B., and Lasky, T. A., 2011. *Lidar for Data Efficiency* [Research Report]. Retrieved September 2011, from Washington, DC: Washington State Department of Transportation: https://www.wsdot.wa.gov/research/reports/fullreports/778.1.pdf.

Yin, D., Du, S., Wang, S., & Guo, Z., 2015. A direction-guided ant colony optimisation method for extraction of urban road information from very-high-resolution images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *8*(10), 4785-4794.

Yu, Y., Li, J., Guan, H., Jia, F., & Wang, C., 2015a. Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8*(2), 709-726.

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

Yu, Y., Guan, H., & Ji, Z., 2015c. Automated detection of urban road manhole covers using mobile laser scanning data. *IEEE Transactions on Intelligent Transportation Systems*, 16(6), 3258-3269.

Zhang, H., 2016. Rapid Inspection of Pavement Markings Using Mobile Laser Scanning Point Clouds. Retrieved from theses, University of Waterloo: https://uwspace.uwaterloo.ca/handle/10012/10343

Zhang, T., Arrigoni, S., Garozzo, M., Yang, D. G., & Cheli, F., 2016. A lane-level road network model with global continuity. *Transportation Research Part C: Emerging Technologies*, *71*, 32-50.

Zhang, T., Yang, D., Li, T., Li, K., & Lian, X., 2011. An improved virtual intersection model for vehicle navigation at intersections. *Transportation Research Part C: Emerging Technologies*, *19*(3), 413-423.

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.

Zhu, Q., & Li, Y., 2008. Hierarchical lane-oriented 3D road-network model. *International Journal of Geographical Information Science*, *22*(5), 479-505.