# Recognizing Features in Mobile Laser Scanning Point Clouds Towards 3D High-definition Road Maps for Autonomous Vehicles

by

He Zhao

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

The sensors mounted on a driverless vehicle are not always reliable for precise localization and navigation. By comparing the real-time sensory data with a priori map, the autonomous navigation system can transform the complicated sensor perception mission into a simple map-based localization task. However, the lack of robust solutions and standards for creating such lane-level high-definition road maps is a major challenge in this emerging field.

This thesis presents a semi-automated method for extracting meaningful road features from mobile laser scanning (MLS) point clouds and creating 3D high-definition road maps for autonomous vehicles. After pre-processing steps including coordinate system transformation and non-ground point removal, a road edge detection algorithm is performed to distinguish road curbs and extract road surfaces followed by extraction of two categories of road markings. On the one hand, textual and directional road markings including arrows, symbols, and words are detected by intensity thresholding and conditional Euclidean clustering. On the other hand, lane markings (lines) are extracted by local intensity analysis and distance thresholding according to road design standards. Afterwards, centerline points in every single lane are estimated based on the position of the extracted lane markings. Ultimately, 3D road maps with precise road boundaries, road markings, and the estimated lane centerlines are created.

The experimental results demonstrate the feasibility of the proposed method, which can accurately extract most road features from the MLS point clouds. The average recall, precision, and $F_1$-score obtained from four datasets for road marking extraction are 93.87%, 93.76%, and 93.73%, respectively. All of the estimated lane centerlines are validated using the “ground

truthing" data manually digitized from the 4 cm resolution UAV orthoimages. The results of a comparison study show the better performance of the proposed method than that of some other existing methods.

# Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Dr. Jonathan Li, who offers me not only professional knowledge, but also guidance in my life. Also, thanks to my thesis committee member: Dr. Michael Chapman, Professor with the Department of Civil Engineering at Ryerson University, Dr. Baoxin Hu, Professor with the Department of Earth and Space Science and Engineering at York University, and Dr. Yaoliang Yu, Assistant Professor with the Cheriton School of Computer Science at the University of Waterloo for reviewing my thesis and providing me the constructive comments that help me improve the readability of my thesis.

Thanks go to the Fujian Key Laboratory of Sensing and Computing for Smart Cities (SCSC Lab) at Xiamen University for providing raw MLS point clouds used in this study. Special thanks go to Kecheng Zhang and Weiquan Liu at the SCSC Lab for spending time to collect UAV images for validation of my methods.

I would like to thank my friends and group mates: Haocheng Zhang, who provided me relevant background of the road marking extraction; Menglan Zhou, who offered me strong support in generating orthoimages in ArcGIS; Zilong Zhong, who provided me with valuable literatures and suggestions. I would like to express my gratitude to amazing teammates Lingfei Ma and Han Jiang, who I got to know through my MSc journey. I am truly happy about our friendship and would like to thank them for sharing me all the unavoidable ups and downs. Many thanks go to both Dr. Yongtao Yu and Dr. Haiyan Guan who were doctoral students supervised by Prof. Jonathan Li before for their insightful suggestions and intellectual

supports. Also, I would like to thank Susie Castela and Alan Anthony at the Department of Geography and Environmental Management for their administrative support.

The last paragraph goes to my dearest family. I cannot describe how grateful I am to my parents. Without their continual love, support, encouragement and patience throughout my life, this thesis could not be completed.

# Table of Contents

# List of Abbreviations

| | |
|---|---|
| 2D | Two-dimensional |
| 3D | Three-dimensional |
| ACC | Adaptive Cruise Control |
| ALS | Airborne Laser Scanning |
| DARPA | Defense Advanced Research Projects Agency |
| DMI | Distance Measurement Indicator |
| DTM | Digital Terrain Model |
| FN | False Negative |
| FP | False Positive |
| GIS | Geographic Information System |
| GNSS | Global Navigation Satellite System |
| GPS | Global Positioning System |
| IMU | Inertial Measurement Unit |
| LiDAR | Light Detection and Ranging |
| LDW | Lane Departure Warning |
| LKAS | Lane Keeping Assist System |
| MLS | Mobile Laser Scanning |
| PCA | Principal Component Analysis |
| PCL | Point Cloud Library |
| PS | Phase Shift |
| POS | Position and Orientation System |
| RADAR | Radio Detection and Ranging |

| | |
|---|---|
| RANSAC | Random Sample Consensus |
| SLAM | Simultaneous Localization and Mapping |
| TLS | Terrestrial Laser Scanning |
| TOF | Time of Flight |
| TP | True Positive |
| UAV | Unmanned Aerial Vehicle |
| UTM | Universal Transverse Mercator Projection |
| WGS84 | World Geodetic System 1984 |

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

Since the 1980s, an increasing number of companies have worked towards autonomous vehicle technology due to its great potential and vast development prospects. From the early exploration to nowadays' fleet of driverless cars, autonomous vehicles have always attracted attention. Today, as one of the most well-known autonomous vehicle designer, Google has brought its driverless cars from laboratory into commercial research. In 2013, both Audi and Toyota published their autonomous vehicle projects, jumping into the new trend of commercial autonomous vehicle development (Hsu, 2013). According to a report by BCG (2017), wider commercialization of fully autonomous vehicle is expected to be launched in the next ten years, and by 2035, 12 million fully autonomous units could be sold a year globally. The market for partially and fully autonomous vehicles is expected to leap from about $42 billion in 2025 to nearly $77 billion in 2035.

In recent years, with more and more driving assistance systems flooding into the market, such as lane departure warnings (LDWs), lane keeping assist systems (LKAS), and adaptive cruise control systems (ACC), driving becomes easier than before, and people are more likely to enjoy their driving experience (Thorsten et al., 2012). However, fully autonomous driving in real and complex urban areas remains a significant but exclusive goal. One primary limitation that in the way of this promising field is the lack of precise localization and navigation methods.

The precise localization of an autonomous vehicle is crucial for driving safety and efficiency. Autonomous vehicles could conduct a series of catastrophic behaviors with inaccurate localization and planning, such as driving on the wrong side of the road, or illegally changing lanes. Sometimes autonomous systems might identify sidewalks and maintenance lanes to be driveable, or believe a destination is inside of a large obstacle (Thorsten et al., 2012).

Autonomous vehicles need to know their precise location. To help these "robots" make sense of the world, several sensors are commonly employed in autonomous technology, including image-based sensors like cameras and remote sensing devices like Radio Detection and Ranging (RADAR) and Light Detection and Ranging (LiDAR) (Thorsten et al., 2012). Also, Global Positioning System (GPS) antennas are mounted to measure the vehicles' position and to estimate its motion. Generally, a "sense-plan-act" procedure is widely used by most autonomous systems (Siciliano & Khatib, 2008). With good perception, autonomous vehicles can perceive their local environments, thus achieving reasonable planning and acting.

Nevertheless, all of these sensors have limitations. For instance, cameras are better at capturing data in sunny days than fog and rains. The ambient condition sensitivity becomes its primary limitation. Unlike camera-based systems, radar uses signals' time of flight to detect objects and compute the distance to objects. Radio waves are useful tools that enable radar systems to gather data regardless of weather conditions. However, one critical shortcoming of radar is its reflectivity limitation, that is, only metallic objects can be successfully detected, while non-metallic features become invisible to radar sensors. Particularly when it comes to pedestrian detection or road boundary detection, radar systems are considered as incompetent. Moreover, GPS signals weaken or become corrupted under dense canopy, urban structures and other shelters, resulting in a delayed reaction of navigation systems (Urmson, 2008). Most importantly, in the

field of precise trajectory design and detailed road information registration, sensors are not always reliable. In reality, autonomous vehicles need to know which area of the lane can be safely driven on, and when making lane-changing decisions, the system needs to extrapolate specific trajectories. Therefore, if some maps with pre-set information of the environment are available, the performance of autonomous navigation and can be drastically improved.

Traditional maps commonly have an error range of approximately 10 m. Different from traditional 2D maps, high-definition maps are created from MLS data with the error of 10-to-20 cm, containing billions of 3D point cloud data which could represent detailed 3D road network topologies. These high-precision maps can give autonomous vehicles greater ability to maneuver smoothly, while the on-board sensors are responsible for another part of the task: to ensure the driving safety. It is well-known that the detection range of sensors is limited due to the presence of corners, ramps, and obstacles like moving vehicles. If an autonomous vehicle can obtain real-time data from a priori map, it can not only "expand" its detection range, knowing whether there are obstacles, or sharp curves in front of it, but also allows sensors to respond more quickly.

By combining real-time sensory detection with a priori knowledge, a priori map with rich metadata can transform the complicated sensor perception mission into a simple map-based localization task. Today, the idea of map-aided localization has been widely accepted by automobile manufacturers. Compared with traditional localization methods like GPS / Inertial Navigation System (INS) integration method and the Simultaneous Localization and Mapping (SLAM) (Tao, 2013), map-aided localization is capable of long-distance path planning and never suffers from the problems that sensor-based odometry faces, such as wrong time integration and loop closure (Pink, 2009).

However, map-aided localization still faces several challenges. First, MLS point cloud data can only be operated by specialists due to its massive data volume and high-complexity. The size of a MLS dataset representing 1 km of road length can be over 1 GB. Moreover, the inconsistent distribution of point density makes MLS data difficult to process. Hence, there is no commercial software with comprehensive functions that can handle such types of data. Most existing methods for creating 3D road maps still require significant human input. To solve this problem, this study attempts to develop an automated approach for creating 3D high-definition road maps.

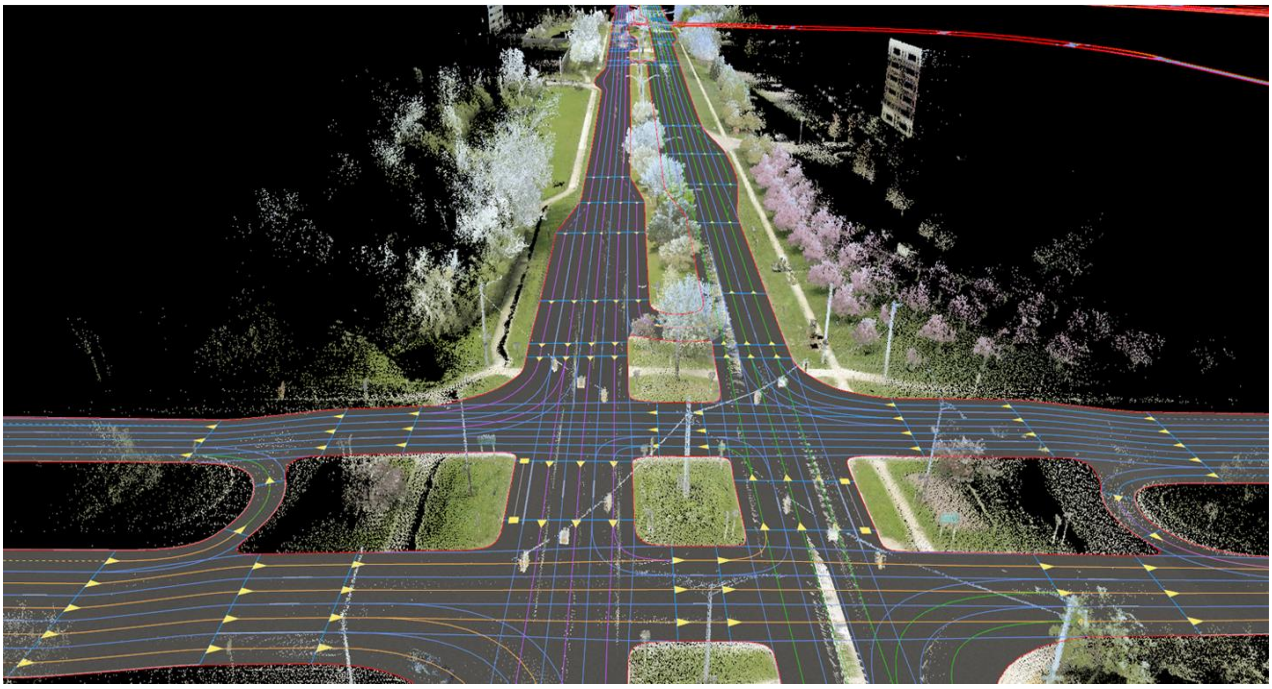


Figure 1.1: A 3D high-definition road map (HERE, 2016)

Figure 1.1 shows a sample of high-definition road maps in the 3D coloured point cloud environment, which clearly indicates the lane lines (highlighted in blue), road edges (highlighted in red), lane centerlines (highlighted in orange), and driving directions (presented with arrows). However, the detailed procedures for creating such maps cannot be found in literature.

## 1.2 Objectives of the Thesis

As map-aided localization has been employed by increasing autonomous vehicle designers and manufacturers, a great number of consumer-grade 3D high-definition road maps are required. These regional specific 3D high-definition road maps tend to widely facilitate autonomous navigation and change people's trip mode. However, there are no or few established methods for creating 3D high-definition road maps for autonomous vehicles. In this study, a semi-automated approach for creating 3D high-definition road maps is proposed. The intended objectives are listed below:

1) To develop a semi-automated algorithm for extracting curbs from massive MLS point clouds, generating road boundaries and extracting road surfaces.
2) To develop a semi-automated algorithm for road marking extraction by semantically and topologically analyzing road characteristics, providing lane-level navigation for autonomous vehicles.
3) To estimate lane centerlines and design appropriate trajectories for autonomous vehicles.
4) To provide prototype procedures for creating 3D high-definition road maps using MLS point clouds.

## 1.3 Structure of the Thesis

The thesis is organized into five chapters.

Chapter 1 briefly introduces the development of autonomous vehicles, the working principle of autonomous vehicles, as well as the motivation of this study – the requirement for 3D high-definition road maps. Finally, the objectives and structure of this thesis are given.

Chapter 2 introduces some background information of this study, such as the working principle of MLS system, the components of a 3D high-definition road map, and the basic elements for lane-level navigation. This chapter also presents the literature review which related to this thesis, including road surface extraction techniques, road marking extraction techniques, and other existing road maps.

Chapter 3 describes the study area, datasets, related software and platforms, followed by an explicit description of the proposed approach. In this chapter, the stepwise data processing algorithms are described in detail, including the voxel-based upward growing algorithm, the segmentation-based curb detection algorithm, the intensity analysis, the statistical outlier removal algorithm, the conditional Euclidean clustering, the RANSAC-based road edge refinement algorithm, and the lane marking extraction algorithm based on road design specification. Finally, this chapter will end with an accuracy assessment method that is adopted to evaluate the performance of the proposed algorithms.

Chapter 4 presents the output of the proposed method. This chapter also discussed the accuracy of the experimental results and the efficiency of the proposed algorithms. Besides, two comparative studies are introduced.

Chapter 5 summarizes the results and findings of this study, as well as the contributions and limitations. Finally, recommendations for future studies are proposed.

# Chapter 2

# Background and Related Work

This chapter presents the background information and literature review which related to this study. Section 2.1 briefly introduces the principle of mobile laser scanning. Section 2.2 describes the elements of lane-level navigation and 3D high-definition road maps. Section 2.3 provides the literature review of existing methods for road surface extraction, road marking extraction, and navigation map generation. Section 2.5 summarizes this chapter.

## 2.1 Introduction to Mobile Laser Scanning

Mobile laser scanning is a mapping technology that is used for collecting 3D geo-referenced data (point clouds) of physical environments by laser scanners mounted on mobile vehicles (Schwarz and El-Sheimy, 2007; Marshall, 2011). MLS systems send out and receive near-infrared laser beams to rapidly scan surfaces of objects and calculate the range using Time of Flight (TOF). Another range measurement technique of MLS system is Phase Shift (PS), which utilize continuous laser illumination and amplitude modulation of the signal to determine the range at high frequency (Kuuko, 2013). Compared with conventional mobile mapping systems using optical or digital cameras, MLS systems can obtain 3D coordinates and intensity properties of the scanned features, which makes it more suitable for long-distance and detailed mapping tasks (Olsen, 2013; GIM, 2013).

During the data collection missions, the GPS and Inertial Measurement Unit (IMU) constantly calculate real-time geodetic coordinates (WGS84) of the vehicle at the phase centre of the GPS antenna, the laser scanner(s) keep transmitting and receiving laser pulses to measure 3D

points, and the camera(s) capture images with the distance measurement indicator (DMI) displacement (Puente, 2011). Eventually, real-time ranging and imaging data with accurate WGS84 coordinates can be obtained.

Figure 2.1 illustrates the principle of direct geo-referencing, in which, the position of the point of interest (P) in the local mapping frame can be determined by Eqs. (2-3) and (2-4) (Glennie, 2007). The parameters used in this calculation are listed in Table 2.1.

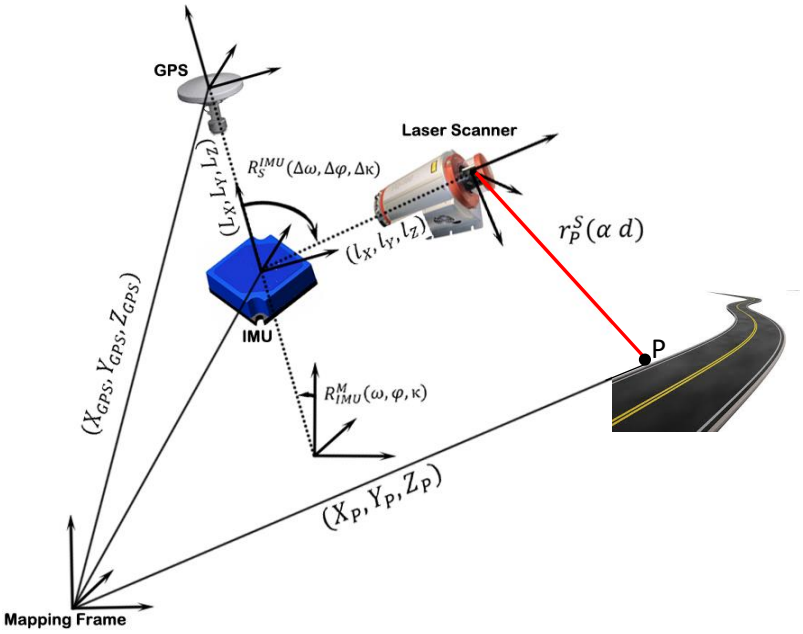



Figure 2.1 Principle of direct geo-referencing

$$\begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix} = R_M^{IMU}(\omega, \varphi, \kappa) \cdot (R_{IMU}^S(\Delta\omega, \Delta\varphi, \Delta\kappa) \cdot r_P^S(\alpha\ d) + \begin{bmatrix} l_X \\ l_Y \\ l_Z \end{bmatrix} + \begin{bmatrix} L_X \\ L_Y \\ L_Z \end{bmatrix}) + \begin{bmatrix} X_{GPS} \\ Y_{GPS} \\ Z_{GPS} \end{bmatrix} \quad (2\text{-}3)$$

$$r_P^S(\alpha\ d) = d_P^S \cdot \begin{bmatrix} \cos\alpha \\ 0 \\ \sin\alpha \end{bmatrix} \quad (2\text{-}4)$$

Table 2.1 Parameters of the direct geo-referencing

| Parameters | Description |
|---|---|
| $X_P, Y_P, Z_P$ | Coordinates of Point P in the mapping frame |
| $X_{GPS}, Y_{GPS}, Z_{GPS}$ | Coordinates of GPS antenna in the mapping frame |
| $R_M^{IMU}(\omega, \varphi, \kappa)$ | Rotation matrix between IMU body frame and mapping frame. $\omega, \varphi, \kappa$ are the roll, pitch and yaw of IMU in the local mapping frame, respectively |
| $r_P^S(\alpha\ d)$ | Range vector from the laser scanner to the Point P. α, and d are the scan angle and range measured by the laser scanner. |
| $L_X, L_Y, L_Z$ | The lever arm offsets from the IMU to the laser scanner |
| $l_X, l_Y, l_Z$ | The lever arm offsets from the IMU to the phase center of GPS |
| $R_{IMU}^S(\Delta\omega, \Delta\varphi, \Delta\kappa)$ | Rotation matrix between the laser scanner and IMU. $\Delta\omega, \Delta\varphi, \Delta\kappa$ are boresight angles which align the scanners with the IMU |

It is notable that some of these parameters are provided by the manufacturer, such as the lever arm offsets and the rotation matrix between different components, while other information cannot be acquired directly and needs to be calculated using geometrical relations of other measured vectors (Barber et al., 2008). In addition, the accuracy of the acquired MLS data highly depends on the performance of navigation solution system (GNSS receiver, IMU, DMI) and laser scanners, as most of the other errors such as boresight errors and lever-arm errors can be corrected by system calibration in advance (Wang, 2016).

## 2.2 Introduction to 3D High-definition Road Maps

The automotive industry is rapidly evolving, and the march toward autonomous vehicles has begun. However, there is much to accomplish before vehicles can really take control. 3D high-definition road map is the most powerful tool that help autonomous vehicles plan the right maneuvers and strategies. By helping give foresight to a car's computers, and adding redundancy

to the car's understanding of the situation it faces, 3D high-definition road maps could ease the burden of autonomous navigation.

Different from traditional road maps, 3D high-definition road maps can provide lane-level navigation with centimeter-level accuracy. Figure 2.2 presents a high-definition-map-based sample showing a street view with signage definitions and extraction.


Figure 2.2: Street view with signage definitions and extraction (SANBORN, 2017)

In a 3D high-definition road map, each road is decomposed into a sequence of road segments, and in each road segment, the number of lanes is constant. Adjacent lanes of a given road segment going in the same direction are implicitly connected everywhere. A 3D high-definition road map should contain abundant road information, including road reference lines, all types of road markings, road edges, traffic signs, and lane centerlines. It is noteworthy that the

centerlines of each single lane are elaborately depicted in 3D high-definition road maps, as they can be used as position vectors offering precise localization and lane-level navigation during autonomous driving. Some basic elements of lane-level navigation are listed in Table 2.2.

Table 2.2 Basic elements of lane-level navigation (CAMP, 2004)

| # | Attributes | Unit/Type | Resolution | Maximum error | Priority |
|---|---|---|---|---|---|
| L1 | Number of Lanes | Integer | 1 | 0% | Required |
| L2 | Road Types | Speed dependant | N/A | 0% | Required |
| L3 | Lane Number | Integer | 1 | 0% | Required |
| L4 | Lane Types | Normal | N/A | 0% | Required |
| | | Entrance Ramp | | | |
| | | Exit Ramp | | | |
| | | Ramp Junction | | | |
| | | Shoulder | | | |
| L5 | Lane Width | cm | 1 cm | 10 cm | Required |
| L6 | Lane Markings | Color | N/A | 100% | Required |
| | | Type | | | |
| | | Thickness | | | |
| | | Material | | | |
| L7 | Lane Marking Width | cm | 1 cm | 5 cm | Required |
| L8 | Slope | % | 0.10% | 0.20% | Required |
| L9 | Curvature | 1/m | 0.00005 | 2% | Required |
| L10 | Speed Limit | km/hr | 1 km/hr | 0% | Required |
| L11 | Minimum speed | km/hr | 1 km/hr | 0% | Optional |
| L12 | Lane Direction | Degree | 0.01 | 0.05 | Required |

## 2.3 Related Work

### 2.3.1 Road Surface Extraction

The first step of creating a 3D high-definition road map is road edge detection and road surface extraction. It is usually difficult to identify or extract specific features from large-volume and dense MLS points clouds, especially without the help from ancillary data (e.g. trajectory data). Many studies have focused on this complicated task using a variety of methods. Munoz et al. (2008) successfully detected road surface points using an Associative Markov Network. By analyzing the characteristics of MLS data in horizontal planes, their preliminary results reached a relatively high accuracy. A region adjacency graph representation method is applied by Hernandez and Matcotegui (2009) for automatically detecting road surfaces. However, their algorithm is time-consuming and difficult to achieve, as it searches for neighboring points for each single point, and the points are irregularly distributed. To simplify the approach for road detection, some researchers mainly relied on elevation, point density, and point intensity to extract road information. For example, Li et al. (2004) proposed a method which used the density of projected points as a key criterion for road extraction. Clode et al. (2004) designed a hierarchal classification technique that can identify road and non-road objects based on elevation and intensity values. However, their accuracy for road segmentation was largely affected by cars and other objects on the road. Similarly, Guan et al. (2015) removed non-ground objects from the geo-referenced image using an elevation filter and segmented road points using a point-density filter.

Different from 2D feature filtering approaches, some road segmentation methods attempted to fit lines to point clouds. MLS data is comprised of scan lines with dense points. The detection of road edges (curbs, kerbstones, guardrails) is used to determine road boundaries and segment the road surfaces. Yoon and Crane (2009) determined road seeds by detecting road curbs

and separating road surface from its surrounding environment. McElhinney et al. (2010) introduced 2D cubic splines to fit road cross sections, but it is difficult to assess the accuracy of the detected road edges. It is obvious that there will be misclassifications if some non-road points match the criteria of elevation difference (Vosselman, 2000; Roggero, 2001; Sithole, 2001; Vosselman et al., 2001). Thus, slope criteria are used to further determine road edges (Jakkola et al, 2008), combined with intensity (McElhinney et al., 2010), and vehicle proximity (Yu and Zhang, 2006). Vosselman and Liang (2009) detected kerbstones from airborne laser scanning (ALS) data by searching for small height jumps, although the presence of cars occluding kerbstones reduced the accuracy of their road extraction results. Ibrahim and Litchti (2012) proposed a method to identify curbs and road surface from terrestrial laser scanning (TLS) point clouds. The ground and non-ground points are classified based on the variation of point density and the distance to the mobile vehicle's trajectory. Subsequently, the extracted results were refined by analyzing the morphological characteristics of each points' neighboring points. Hervieu and Soheilian (2013) computed normal vectors of point clouds and angular distances to normal vectors of the ground to detect curbs and curb ramps.

Other methods for road surface extraction mainly involve 3D geometric features filtering, surface growing and voxel-based algorithms (Zhang, 2016). Yuan et al. (2008) developed an algorithm for road surface segmentation from terrestrial laser scanning data. In their approach, a fuzzy clustering method was applied to cluster raw TLS points. Then, they used straight lines to fit the linear clustered data. An approach proposed by Goulette et al. (2006) successfully identified road, trees and building facades from TLS data. The road was recognized as a horizontal plane with high point density, and the extracted road points were further used to calculate the road width. Elberink and Vosselman (2009) focused on the 3D modelling of highway infrastructure using both

airborne laser scanning data and 2D topographic maps. A seed-growing algorithm and the Hough transformation are employed in their approach to extract road polygons from 2D topographic map data. By combining the map polygons and LiDAR elevation, 3D reconstruction of highway infrastructures was achieved. RANdom Sample Consensus (RANSAC) is a widely-used algorithm for plane fitting. Lam et al. (2010) employed RANSAC to fit planes to many small regions of the raw point cloud data. Then, the Kalman filter was used to interconnect the fitted planes. Finally, the road surface is delineated by a combination of planes.

Most of the existing algorithms are complicated and require some priori knowledge and experience. In order to simplify the road extraction approach, Cabo et al. (2013) proposed the space regular voxelization, in which the raw point clouds were partitioned into voxels, and all voxels were segmented into horizontal planes. Through a tridimensional neighborhood analysis, target poles were clustered in these horizontal slices and finally reconstructed. Yu et al. (2015) also developed a voxel-based algorithm to extract roads. An octree structure was applied to contain the partitioned point clouds. Each voxel kept growing upward towards its 9-neighbour voxels until it reached a boundary, and the ground clusters can be determined if the elevation of the topmost voxel is lower than the given threshold.

Although most of the studies presented promising results, limitations inevitably existed in these approaches. For instance, some of the mentioned algorithms distinguished road points from non-road objects, but failed to generate smooth road edges. One more common limitation is that these algorithms mainly focused on a specific road type (i.e. straight roads, curve roads) or a particular study area (i.e. urban area, rural area, or highway). Probably high-accuracy experimental results can be produced in some specific conditions, but these algorithms are not suitable for all scenarios. For example, the algorithm proposed by Guan et al. (2016) performed well for straight

roads, but relatively poor for curves. Also, there would be tremendous challenges if we apply a road edge detection method for extracting road surface in rural areas, as most country roads comprises grass-soil without curbs or guardrails. Therefore, the lack of practical algorithms for large-scale areas is still a huge limitation in this field. In addition, the thresholds used in these studies are highly dependent on people's priori knowledge and experience, which limits the usage of these algorithms to a selected group of people. Therefore, further research should have been done to reduce the dependency of parameters, thus achieving fully-automated road information identification and extraction.

Table 2.3 Summary of different road surface extraction techniques

| Publication | Core Algorithms | Limitations | Advantages |
|---|---|---|---|
| Li et al. (2004) | Point density & Elevation filtering | Cannot identify the wheels of vehicles | High computational efficiency |
| Guan, et al. (2015) | Point density &Elevation filtering | Not suitable for steep terrain and curves | Straightforward & High efficiency |
| Clode et al. (2007) | Point intensity & Elevation filtering | Cannot detect boundary lines | High computational efficiency |
| McElhinney et al. (2010) | Road Edge detection | Require trajectory data & fail in rural area | High accuracy |
| Guan et al. (2014) | Road Edge detection | Require trajectory data & fail in rural area | High accuracy |
| Yang et al. (2013) | Road Edge detection | Require trajectory data & fail in rural area | High accuracy |
| Hervieu and Sohelian (2013) | Normal vector analysis | Cannot detect boundary lines & inefficient | Can Handle large area |
| Vosselman et al. (2004) | Surface growing | Cannot detect boundary lines & inefficient | Can Handle large area |
| Zhou et al. (2012) | RANSAC plane fitting | Performance depends on the model used | High efficiency & Straightforward |
| Yu et al. (2015) | Voxel-based growing | Not suitable for steep terrain and curves | High efficiency |
| Zhao & Shibasaki (2002) | Height deviation & Scan range filtering | Require trajectory data & Cannot detect lines | Use smoothness to detect road surface |

Some important road surface extraction techniques are summarised and compared in Table 2.3. In summary, road edge detection based methods tend to produce more accurate results than other techniques, as they not only extract road surfaces, but also detect road edges and generate smooth boundary lines to refine the road extraction results. In addition, other attributes except elevation and slope can also be used to detect road and road edges, such as point density (Guan et al. 2007) and point intensity (Clode et al. 2007). As to computational time, region growing (or voxel growing) approaches have proven their high efficiency, as they process data in a global scale instead of local blocks. Also, ancillary data such as trajectory points is not necessary for these algorithms, resulting in less computational time during the pre-processing steps. However, balancing the accuracy and efficiency of an approach is always a tough problem, hence different approaches are designed for different research purposes.

### 2.3.2 Road Marking Extraction Techniques

Road markings are important features in traffic management systems, as they provide necessary information for drivers and pedestrians. Meanwhile, they also have the function of navigating autonomous vehicles. In general, road marking extraction methods can be divided into two categories: (a) digital images or video based and (b) LiDAR data based.

Digital images and videos have been investigated for road marking extraction for a long time. Kheyrollahi and Breckon (2010) developed a multi-level-threshold segmentation method to identify road marking candidates. With a set of local thresholds, connected contours can be extracted from the image with inconstant luminance. Subsequently, the artificial neural network algorithm was applied to classify the extracted road markings based on their morphological characteristics. Wang et al. (2009) and Mathibela et al. (2015) also detected road markings from digital images. They both adopted an Inverse Perspective Mapping (IPM) method to transform the

viewing angle of the image from the original one to a top-view perspective. Then the Otsu thresholding (Otsu, 1979) was conducted to segment different road markings. However, either photographs or videos lack geometrical information. For example, images and videos cannot recognize the material of the road surface and capture the shape of road markings (Guan et al. 2014). Besides, the performance of videos and digital images are affected by environmental factors, such as weather conditions, the time of day, and shadow from vehicles or trees. Also, human-involved road inspection requires people to continuously take pictures on the road, which is time-consuming, labor intensive, and dangerous.

Compared with images and videos, MLS systems are more robust tools for road feature detection. Therefore, a variety of MLS-based road marking extraction algorithms have been developed in recent years. Generally, road markings have higher retro-reflectance than other objects on the road, thus making the reflectance or intensity value of a target an important information for road feature detection. Taking the study of Smadja et al. (2010) as an example, intensity information was the only attribute they used for extracting road markings. Jaakkola et al. (2008) implemented a radiometric correction for the LiDAR intensity data before extracting road markings. Chen et al. (2009) developed an algorithm focusing on lane marking extraction from MLS point clouds. Apart from intensity filter, the standard deviation of elevation was also analyzed in their approach, and finally the lane markings were classified through 2D Hough transformation. By intensity-thresholding, Butler (2011) clustered road markings and fitted convex hulls to them. A neighbourhood distance thresholding was also used to constrain the errors. Yang et al. (2012) introduced an approach that generated 2D images from the extracted LiDAR data and highlighted the outlines of road markings based on their priori knowledge of the local traffic system. In Vosselman (2009), combined with a connected components analysis, a distance

thresholding method was applied to overcome the challenge of missing intensity data. Their strategy was also used by Kumar et al. (2014). In their study, a function extracted road markings based on range thresholding. Guan et al. (2014) conducted a multi-thresholding segmentation based on the distribution of point density. The road markings were segmented according to their special point density range, the estimated mean, and the standard deviation. Table 2.4 concludes the existing road marking extraction strategies as well as their advantages and limitations.

Table 2.4 Summary of road marking detection techniques

| Publication | Method | Advantages | Limitations |
|---|---|---|---|
| Toth et al. (2008) | Intensity filtering | Efficient in simple scenes | Inaccurate |
| Smadja et al. (2010) | Intensity filtering | Efficient in simple scenes | Inaccurate |
| Jaakkola et al. (2008) | Intensity & Distance filtering | Relatively more accurate & less outliers | Require trajectory data & Correction is rough |
| Kumar et al. (2014) | Intensity & Distance filtering | Relatively more accurate & less outliers | Require trajectory data & Correction is rough |
| Guan et al. (2014) | Multi-thresholding | Use point density to locate road markings | Rely on priori knowledge |
| Yu et al. (2015) | Multi-thresholding | Use distance to trajectory to locate road markings | Performance relies on trajectory data |
| Chen et al. (2009) | Multi-thresholding | Reduce the influence of missing data (intensity) | Performance relies on trajectory data |
| Cheng et al. (2017) | 2D Intensity filtering & region growing | High accuracy | Significant human input |

As shown in Table 2.4, intensity is the most important information that can be used for extracting road markings. Distance dependant thresholding is another common solution for achieving high accuracy in road marking extraction. By calculating the distance from a point to the vehicle's trajectory, a strip area containing targeted road markings could be generated. However, trajectory data is not always reliable, as trajectory points are usually sparse and irregularly aligned, and even zigzagging due to lane changing behaviors. Thus, most of the studies tested their methods on intentionally selected areas, where trajectory points are neatly and linearly arranged. In addition, tedious pre-processing procedures should be done before people can acquire trajectory data. For example, the coordinate system transformation is necessary if different coordinate systems are used in data collection and data processing,

In structured environments like highways and viaducts, a single intensity filter can produce acceptable results, as there are no other features with high reflectance painted on the road. However, when it comes to complicated urban scenes, one or more other attributes should be employed to generate accurate road marking extraction results, such as point density, elevation, etc.

### 2.3.3 Road Map Creation

LiDAR and MLS systems have been investigated for years, and they are widely applied in road inventory survey. In addition, the usage of MLS systems has been extended to many other fields, such as power-line inspection, Digital Terrain Model (DTM) generation, and many types of 3D model reconstruction. However, the concept of 3D high-definition road maps just appeared in recent years, and there is no common standard regarding the format of such maps. Consequently, different research groups provided their own priori maps in different styles. For example, J. Levinson and S. Thrun (2010) produced a map with 2D infrared reflectivity from raw LiDAR data, which is matched against the reflectivity of the corresponding LiDAR point clouds. Urmson et al.

(2008) and Miller et al. (2011) matched the road topology network of maps with different visual features in a current view. In addition, Pink et al. (2009) utilized available aerial images as priori maps to assist the localization of an autonomous vehicle.

In summary, there is no robust and explicit solution for generating 3D high-definition road maps. Most of the previous priori maps remain 2D, and no sufficient geographic information is embedded in those maps. Therefore, the lack of consumer-grade 3D high-definition road maps is a great challenge for autonomous navigation.

## 2.4 Chapter Summary

This chapter first introduced some background information of this thesis, including the working principle of MLS system and the components of a 3D high-definition road map.

The related works of this thesis were also reviewed including the methods for road surface extraction, road marking extraction, and navigation map creation. In conclusion, different algorithms have their own advantages and limitations, as they focus on different data types and types of roads. However, no or few algorithms can produce promising results from large-scale datasets, and most of the methods still require some manual work. Therefore, creating 3D high-definition road maps remains great challenges. In Chapter 3, the algorithms for road feature extraction and 3D high-definition road map creation will be proposed.

# Chapter 3

# Methodology for Creating 3D High-definition Road Maps

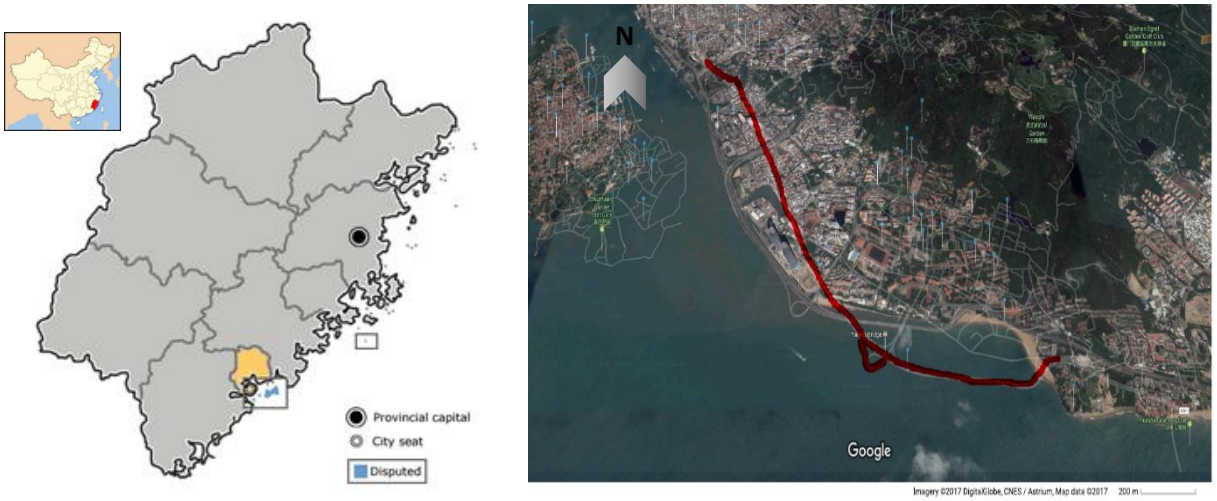
This chapter presents the methodology of this study. Section 3.1 introduces the characteristics of the test datasets and the computing platforms used in this thesis, followed by a stepwise workflow of the proposed methodology for creating 3D high-definition road maps in Section 3.2. Sections 3.3 - 3.6 detail the individual procedures of the proposed methodology, including pre-processing, road edge detection, road surface extraction, road marking extraction, and lane centerline estimation. Section 3.7 describes the accuracy assessment method of this thesis. Section 3.8 summarizes this chapter.

## 3.1 Test Datasets

The MLS datasets used in this study were collected by a research team at Xiamen University using a RIEGL VMX-450 system. The RIEGL VMX-450 system contains two 360° laser scanners with a tilted angle of 135°. The yaw offset of the scanners is ± 30°. In addition, four high-resolution optical cameras, one GNSS antenna, an IMU and a DMI are integrated into this system. RIEGL VMX-450 can scan the surface of an object within a distance up to 800 m and capture detailed features with a high precision of 5 mm at a distance of 200 m. Through using the RiPROCESS software, raw point clouds are transformed to LAS file in the World Geodetic System 1984 (WGS84) or in the Universal Transverse Mercator (UTM) coordinate system.

The MLS Surveys were conducted in Xiamen Island, Fujian, China (longitude 118°04'04"E, latitude 24°26'46"N). The data collection vehicle travels through the roads of interest at the speeds of up to 50 km/hr, resulting in the MLS point densities of 7000-8000 points/$m^2$. The MLS point

clouds collected were processed and saved in multiple LAS files. The point density was extremely high, therefore, a single LAS file representing a 100 m long segment contains millions of points. Figure 3.1 shows the geographic location of Xiamen and the trajectory of the survey area. The scanning parameters of RIEGL VMX-450 system are shown in Table 3.1.




(a) (b)

Figure 3.1: Study area in Xiamen (a) Geographic location of Xiamen, Fujian, China; (b) trajectory of the study area


Figure 3.2 RIEGL VMX-450 system

Table 3.1 Scanning parameters

| Symbol | Quantity | Value |
|---|---|---|
| $S_S$ | Line Scan speed | 400 line/sec |
| $S_G$ | Vehicle speed | 30~50 km/hr |
| $L_{Dist}$ | Scan-line-to-scan-line spacing | 9.71 cm |
| $P_{Dist}$ | Point-to-Point spacing | 5.98 cm |
| $T_{Dist}$ | Target distance | 30 m |
| $L_{Increment}$ | Scan line incremental angles | 0.1143° |
| $P_{Density}$ | Average point density | 7500 points/$m^2$ |

As the surveyed region contains important traffic roads in Xiamen and is quite busy. The average speed of the data-gathering vehicle was approximately 50 km/hr. As a result, $T_{Dist}$ was set as 30 m, and $S_G$ was set 50 km/hr.

Default values were used for other scanning parameters, including scan mode (line), scan increment angles (0.1143°), scan line start (0°) and scan line end (360°). Thus, according to the vehicle speed, the Line Scan speed $S_S$, the line distance $L_{Dist}$, the point distance $P_{Dist}$, and the average point density are estimated as 400 line/sec, 9.71 cm, 5.98 cm and 7500 points/$m^2$, respectively. In addition, all of the data was checked against reference data to assess their accuracy. By calculating their mean standard deviations of elevation accuracy and planimetric accuracy, we the data errors were controlled by ± 5 cm to meet the requirements for urban surveying (Guan et al. 2015).

A total of five test datasets were used in this thesis. Dataset 1 was acquired on September 19th, 2016, while Datasets 2 to 5 were collected on December 20th, 2013. Dataset 1 (64 MB) is a simple urban road corridor that covers a small range of Huandao Road. The length of this road segment is about 30 m. The total number of points in Dataset 1 is 1,920,753 with the point density of 8000 points/$m^2$.

Dataset 2 (23MB) is an entrance of Yanwu Bridge, covering approximately 175 m of road length with only 696,891 points. Over 50% road surface data is missing as this region is not on the surveyed trajectory, and the MLS system captured these points from a side view. Dataset 2 contains both horizontal curves and vertical curves with sparse points, bringing great challenges for the proposed algorithms. This dataset is used to test the robustness of the curb extraction and road segmentation algorithm.

Datasets 3, 4, and 5 are three segments of Yanwu viaduct, covered by 6,758,030 points, 9,058,578 points, and 8,381,952 points, respectively. The lengths of these three segments are about 176.525m, 175.453m and 176.438 m, respectively. The average point density is 7500 points/$m^2$. Dataset 3 is a one-way two-lane road, while Dataset 4 and Dataset 5 are three-lane roads with lane reduction areas (from 3 lanes to 2 lanes). The size of these three LAS files are 226 MB, 303 MB, 281 MB, respectively.

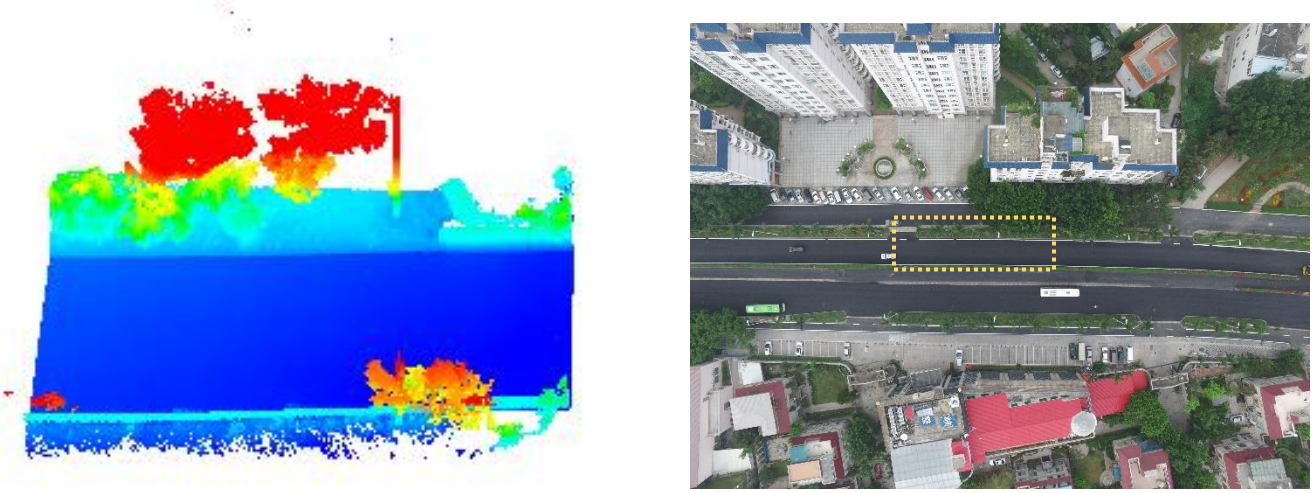
These five datasets contain a variety of road features, including different types of road markings, trees, traffic signs, and buildings. Apart from these five datasets, seven smaller datasets are selected as samples to test the performance of each step of the proposed method and to make comparison with other existing methods. UAV orthoimages are used to validate the experimental results of the proposed method. Detailed characteristics of these datasets are demonstrated in Table 3.2.

Table 3.2 Datasets used in this study

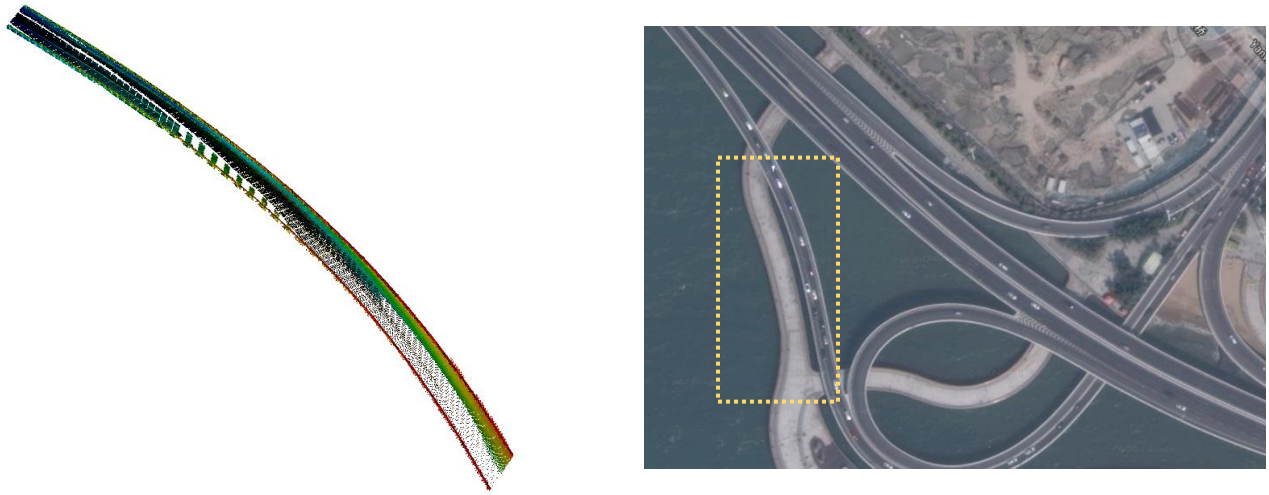
| | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 | Dataset 5 |
|---|---|---|---|---|---|
| Date | 2016/09/19 | 2013/12/20 | 2013/12/20 | 2013/12/20 | 2013/12/20 |
| Size | 64 MB | 23 MB | 226 MB | 303 MB | 281 MB |
| Number of Points | 1,920,753 | 696,891 | 6,758,030 | 9,058,578 | 8,381,952 |
| Location | Huandao Road | Yanwu Bridge | Yanwu Bridge | Yanwu Bridge | Yanwu Bridge |
| Vehicle speed (km/hr) | 30 | 50 | 50 | 50 | 50 |
| Point density ($pts/m^2$) | 7000-8000 | 7000-8000 | 7000-8000 | 7000-8000 | 7000-8000 |
| Road width (m) | 8-10 | 7-9 | 8-10 | 10-16 | 9-11 |
| Road length (m) | 30 | 175.324 | 176.525 | 175.453 | 176.438 |
| Number of lanes | 2 | 1 | 2 | 3-2 | 2-3 |

As shown in Table 3.2, most of the datasets used in this thesis are collected on Yanwu viaduct. As reported by Korosec (2017), most autonomous vehicle test sites are located in special places with extremely structured roads and sufficient protective measures. As safety is still the most significant concern when it comes to self-driving, no or few companies test their autonomous vehicles on busy urban roads. Viaducts and highways with organized road markings, traffic infrastructures, and no pedestrians would be more suitable for testing autonomous vehicles,

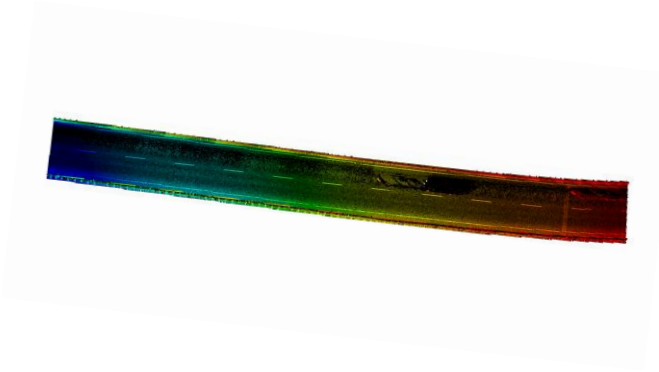
especially at the preliminary stage of autonomous navigation technology (Baidu, 2016). Modest changes to our transportation systems are expected to better support autonomous vehicles, such as reprinting road markings and building protective road facilities. Just as NG and Lin (2016) reported:" Self-driving cars won't work until we change our roads". Figure 3.3 presents the raw point clouds and UAV orthoimages of the five test datasets.



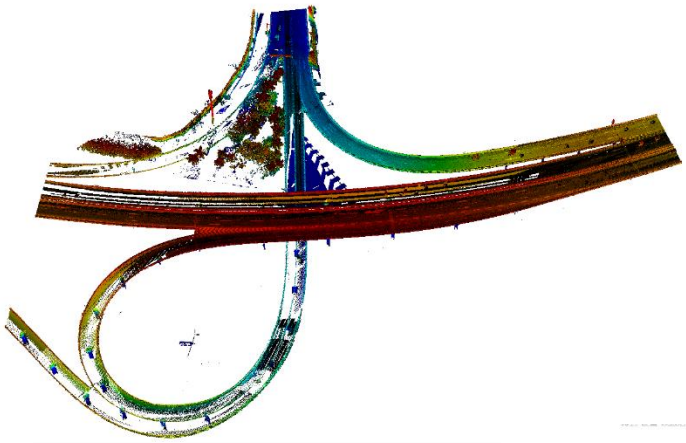
(a)



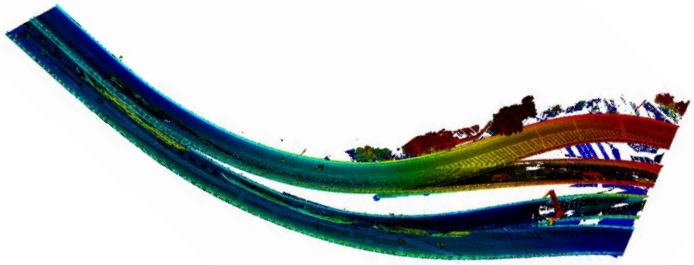
(b)



(c)



(d)



(e)

Figure 3.3 Five datasets used in this thesis

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

### 3.2 Workflow

In order to produce a solid and robust solution for creating 3D high-definition road maps, a stepwise approach is introduced in this chapter. The proposed method endeavors to extract meaningful road information from MLS point clouds, including road edges, road surfaces, road markings, and lane centerlines. These road features can be identified and extracted by semantically and morphologically analyzing data characteristics. In addition, road design standards are used to estimate optimal thresholds. The idea of using road design standards is derived from some recent news and reports. For example, as reported in Eldege (2016), new road design guidelines and specifications will change our roads and highways in the next ten years for supporting autonomous vehicles. Figure 3.4 presents the workflow of the proposed methodology.

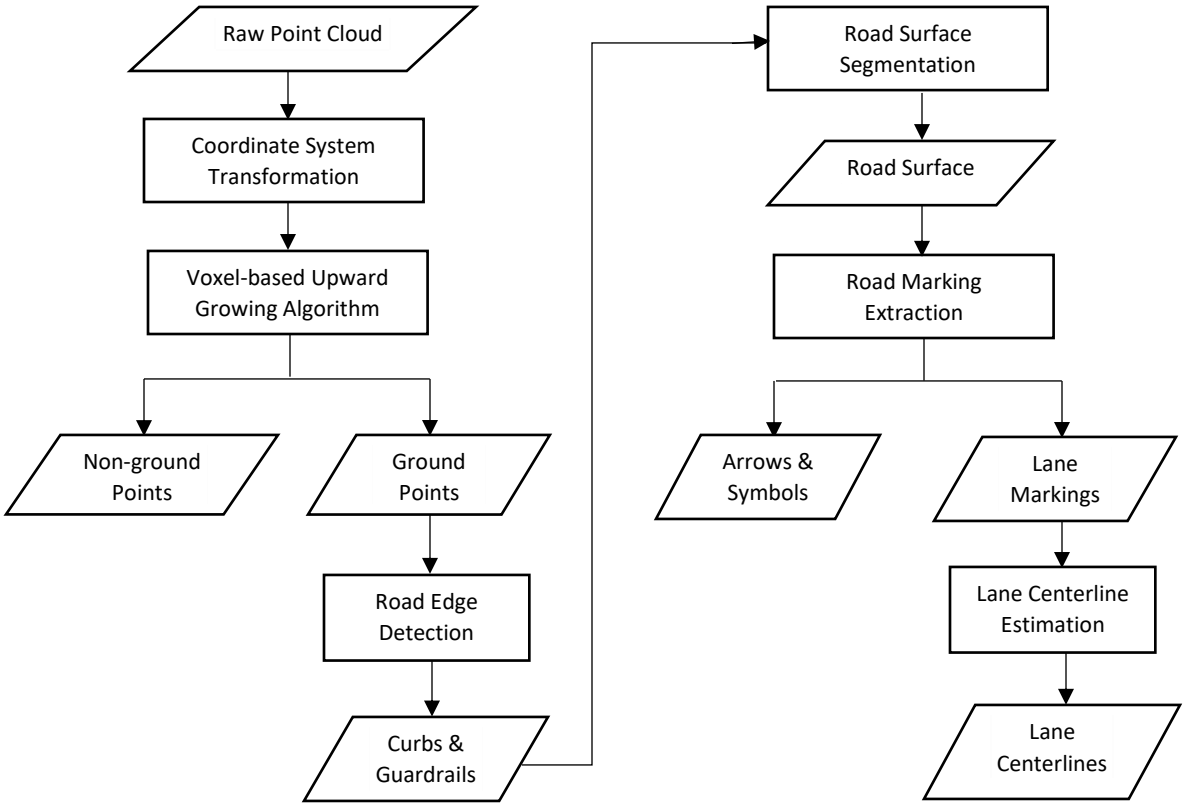



Figure 3.4 Workflow of the proposed methodology

To improve processing efficiency and reduce redundant points, some pre-processing procedures are necessary before road feature extraction. First, as the coordinate system of the MLS system is an arbitrary one with a random user-defined origin point, a coordinate system transformation function is performed to fix the orientation of the vehicle frame so that the points can be more easily interpreted. Second, a voxel-based upward growing method is applied to identify and remove non-ground points from ground points, such as canopies, buildings, traffic signs and other off-ground objects.

To distinguish road surface regions, accurate road edge points are required to represent road boundaries. Thus, a road edge detection function is applied to detect road curbs from the remaining points. To accurately extract curb points, the data is first segmented into a large number of blocks. By analyzing the height difference and slope of consecutive points in each scan line, points with a certain range of height difference and slope are extracted. Then, a quick sort algorithm is applied to sort the road curb candidates by their elevation. The points with the lowest elevation on each side of the road are considered as the bottom of curbs or guardrails and are used to represent the road boundary. Based on the position of road curb points, the region of road surface can be determined. However, misclassification inevitably exists in curb extraction results. To acquire more accurate road edges, a RANSAC algorithm is used to refine the extracted curb points, removing outliers, and fitting a linear model to the generated road boundaries.

In this study, road markings are divided into two categories and separately extracted. On the one hand, white and yellow lines whether solid or broken are classified as lane markings. It is well-known that lane markings are painted with reflective materials and are often drawn with specific distances to road edges. Thus, an intensity and distance-to-road-edge thresholding method based on road design standards is applied to distinguish these lane markings. Specifically, with the

refined road curb points as reference, a window continuously moves in each block searching for points with high retro-reflectance and apart from its corresponding reference line with a certain distance. On the other hand, arrows, symbols, and words are also important road markings that guide and control traffic. To extract these directional and textual road markings, an intensity filter is first applied to detect points with high intensity values. Next, a statistical analysis outlier removal algorithm is conducted to remove noise. After noise removal, all remaining points are organized into semantic clusters based on conditional Euclidean clustering algorithm. Afterwards, by analyzing the size and width of different clusters, arrows and words can be distinguished from lane markings. Finally, the clusters of lane markings are selected to calculate the coordinates of lane centerlines. Thus, a 3D high-definition road map with clear road boundaries, road markings, and the estimated lane centerlines is created.

Lastly, to evaluate the experimental results, the created road maps are converted into SHP files and overlapped with UAV orthoimages in ArcGIS. Post-processing, such as format conversion, can be conducted to suit other research needs. Finally, the generated 3D high-definition road maps are ready to navigate autonomous vehicles.

The programming platform of this thesis is Microsoft Visual Studio 2015. A third-party programming library Point Cloud Library (PCL) is also utilized in this thesis to realize conditional Euclidean clustering and statistical analysis. In addition, Cloud Compare v2.6.3 and Quick Terrain Reader v8.0.6.2 are used to display data and visually interpret experimental outputs. MATLAB R2016 a (9.0.0.341360) is employed to conduct 3D line fitting, and ArcGIS v10.2.2 is used to process the generated lane centerlines and conduct accuracy assessment.

### 3.3 Pre-processing

#### 3.3.1 Coordinate System Transformation

In this study, the MLS system utilized a right-handed orthogonal coordinate system with an arbitrary user-defined orientation and starting point. Such a random coordinate system makes it more difficult to depict the relative positioning of points. In order to reduce the complexity of interpretation in further processing steps, the orientation of the vehicle frame is fixed so that the x-axis faces towards the front of the vehicle, the y-axis is toward the right side of the vehicle, and the z-axis is toward the top of the vehicle. The coordinate system transformation can be carried out by:

$$X_{Di} = D + (1+k)\, R(\varepsilon_x)\, R(\varepsilon_y)\, R(\varepsilon_z) X_{Gi} \tag{3-1}$$

$$R(\varepsilon_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varepsilon_x & \sin\varepsilon_x \\ 0 & -\sin\varepsilon_x & \cos\varepsilon_x \end{pmatrix} \tag{3-2}$$

$$R(\varepsilon_y) = \begin{pmatrix} \cos\varepsilon_y & 0 & -\sin\varepsilon_y \\ 0 & 1 & 0 \\ \sin\varepsilon_y & 0 & \cos\varepsilon_y \end{pmatrix} \tag{3-3}$$

$$R(\varepsilon_z) = \begin{pmatrix} \cos\varepsilon_z & \sin\varepsilon_z & 0 \\ -\sin\varepsilon_z & \cos\varepsilon_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3-4}$$

where $X_{Di}$ and $X_{Gi}$ are the coordinates of MLS point clouds in the transformed coordinate system and original coordinate system, respectively. $\mathbf{D} = (\boldsymbol{\Delta}X, \boldsymbol{\Delta}Y, \boldsymbol{\Delta}Z)$ is the translation matrix, and k is the scaling factor denoted as the ratio between the original and the transformed coordinates. Where $\varepsilon_x, \varepsilon_y, \varepsilon_z$ are the three rotation angles of the 3D coordinate system and $R(\varepsilon_x)$, $R(\varepsilon_y)$, $R(\varepsilon_z)$ are their

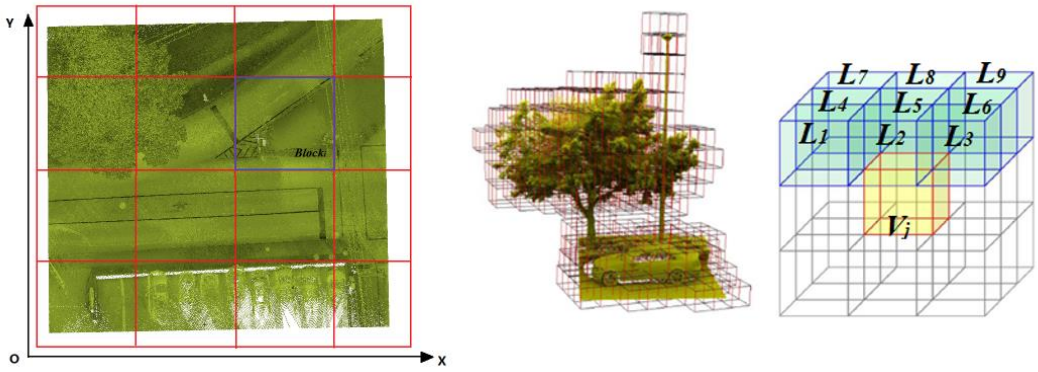
corresponding rotation matrix. To fix the orientation of x-axis and y-axis, two consecutive trajectory points are selected as reference points to calculate the rotation angle. Two points in the vertical plane are used to fix the orientation of z-axis. In addition, when dealing with curve roads, this step is not necessary.

### 3.3.2 Non-ground Point Removal

Typically, MLS data involves large-volume and high-density points. There is no doubt that it would be difficult and inefficient to process these points simultaneously. Since this study only focus on extracting road information, the non-ground points have become relatively redundant. Thus, employing a method to remove non-ground points is significant to reduce the data complexity and reduce computational time. Also, with less outliers, the performance of further road feature extraction algorithms could be improved. Based on these consideration, a voxel-based upward growing algorithm (Yu et al., 2015) is applied to generally filter out non-ground points from the raw MLS data.

Figure 3.5 shows the principle of the voxel-based upward-growing algorithm. In this algorithm, the raw data point clouds are first horizontally segmented into a number of blocks, and the width of each block is determined by the size of test datasets. Instead of processing the entire point clouds globally, this strategy can greatly reduce the influence of ground undulation and produce more accurate results. Furthermore, the points in each block will be further divided into a series of voxels. The certain width of voxels is determined by average point density using Octree Spatial Index. More voxels are needed if the average point density of datasets is higher. As shown in Figure 3.5 (c), there are totally 26 adjacent voxels for each voxel. The voxel-based upward growing algorithm can be described as follows: Each voxel will firstly grow upward toward its 9 adjacent voxels. For example, in Figure 3.5 (c), the 9 'neighbours' of voxel $V_j$ are voxel $L_{1,}$ voxel

$L_2$ voxel $L_3$ … voxel $L_9$. Then, the upward growing algorithm will regard the '9 neighbours' of voxel $V_j$ as new starting points and continue grow upward following the same pattern. This process will not stop until all voxels have no adjacent voxels above them.




(a) (b) (c)

Figure 3.5 Voxel-based upward-growing algorithm (Yu et al., 2015)

(a) Segmented raw point cloud; (b) Octree Spatial Index in a local block; (c) Voxel-based upward-growing pattern

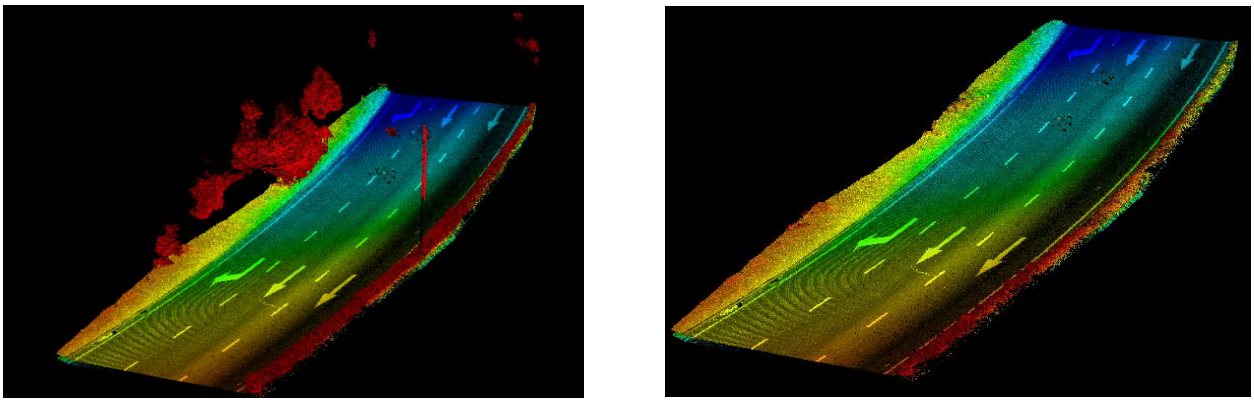
After the upward growing process, the topmost voxel with the maximum local height value in each growing region can be determined. The 'local height value' of a certain voxel is defined as the height difference between the certain voxel and the lowest voxel in its local block. And the 'global height value' of a certain voxel is defined as the height difference between the certain voxel and the lowest voxel in the entire 3D point cloud. Then the following criterion is used to judge whether a voxel belongs to ground voxel or non-ground voxel:

(1) If the global height value of a voxel is larger than a global ground undulation threshold $h_e$, or the local height value of the voxel is larger than a local ground undulation threshold $h_g$, the voxel will be designated as a non-ground voxel and removed from the point cloud.

(2) If the global height value of a voxel is smaller than a global ground undulation threshold $h_{e,}$ and the local height value of the voxel is smaller than a local ground undulation threshold $h_{g,}$ the voxel will be considered as a ground voxel and remain.

The local ground undulation threshold $h_g$ is determined by the maximum z value in a specific growing region, while the global ground undulation threshold $h_e$ is determined by the maximum z value of the entire test dataset. $h_g$ and $h_e$ are useful thresholds that can constrain the largest ground undulation in local blocks and the entire 3D point cloud data, respectively.

The voxel-based upward growing algorithm can successfully remove the majority of non-ground point, enhancing the efficiency and reducing the computational time of further processing steps. Additionally, the integrity of the ground data is completely retained without any data missing. Figure 3.6 shows an example of non-ground points removal using voxel-based upward growing method. Figure 3.6 (a) presents raw 3D point cloud data with abundant non-ground points, and Figure 3.6 (b) illustrates the point clouds after non-ground point removal. As shown in the following figures, trees, traffic signs and light poles are successfully removed from the raw data, and only some shrubs are remained.



(a) A raw 3D point cloud (b) the remained ground points

Figure 3.6 Example of voxel-based upward-growing method

### 3.4 Road Edge Detection and Road Surface Extraction

It is well-known that road curbs perform as road boundaries that separate sidewalks and green spaces from road surfaces (Guan et al., 2015). As mentioned in Chapter 2, a number of studies determined road surface regions by detecting the position of curbs. In this thesis, a revised curb-based road surface extraction method is proposed. The algorithm consists of three steps: (1) data segmentation; (2) curb detection; (3) 3D fitting.

#### (1) Data Segmentation

As the coordinate system of the MLS data has been fixed so that the x-axis is toward the front of the vehicle, the data is first partitioned into many blocks with a user-defined interval ($L_t$) along the x-axis. In addition, the data partition can also be conducted perpendicular to trajectory points if trajectory data is available. Different from some previous research which use trajectory points to segment road surface, this method can let users define the length of each block or the total number of blocks, as trajectory data are sparse and fixed points along the trajectory with a certain interval. Rather than directly processing the entire dataset, processing points in each local block can greatly reduce the negative effects of road undulation. Most importantly, with more blocks, less points will be distributed in each block. The idea is setting a proper interval so that each block can have least but sufficient points. 'Sufficient' means at least one curb candidate points can be detected on each side of the local block, otherwise the length of blocks should be increased. In this thesis, the total number of blocks usually ranges from 30 to 500 depending on the size and the characteristics of test datasets.

**(2) Curb Detection**

After data segmentation, raw point clouds are divided into a series of local blocks. For each block, the proposed algorithm detects curb points based on two criteria: slope and height difference. Based on the observation of the test datasets, pavements are higher than roadways in a neighbourhood with a 7-15 cm elevation jump. Moreover, slopes at the border of pavements and roadways are larger than that of continuous points on roadways. Figure 3.7 presents a typical road cross section shown in intensity point clouds.

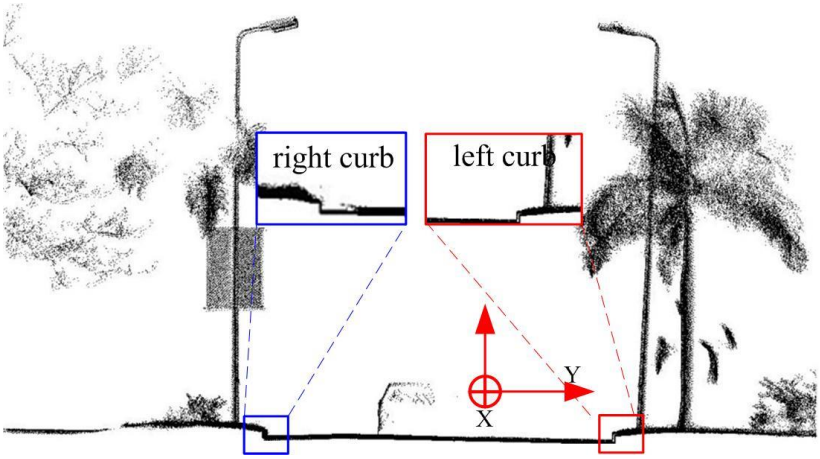



Figure 3.7 Example of a road curb in point clouds

Therefore, these two criteria are applied to determine whether a point is a curb point. The criteria can be described as follows:

$$P_i \begin{cases} curb\ candidate, & if\ (S_{slope} > S_T\ \&\ (G_{min} \le G_i \le G_{max})\ ) \\ non-curb\ point, & otherwise \end{cases} \quad (3\text{-}5)$$

where $S_{slope}$ is the slope of two consecutive points, $S_T$ is a user-defined slope threshold, $G_i$ denotes the elevation difference of a point and its adjacent point in a scan line. $G_{min}$ and $G_{max}$ are the minimum and maximum elevation difference thresholds, respectively. $S_{slope}$ is defined as

$$S_{slope} = \arctan\left(\frac{Z_{i+1}-Z_i}{\sqrt{(X_{i+1}-X_i)^2+(Y_{i+1}-Y_i)^2}}\right)$$

$$S_{slope} \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right) \quad (3\text{-}6)$$

where ($X_i$, $Y_i$, $Z_i$) and ($X_{i+1}$, $Y_{i+1}$, $Z_{i+1}$) are the coordinates of two consecutive points in a scan line. In each block, the algorithm will calculate the slope and elevation difference of any two consecutive points in each scan line. Once a point's slope and elevation difference match the given thresholds, it will be considered as a curb candidate. By this strategy, several curb candidates will be extracted in each block. Subsequently, a quick sort algorithm is performed to sort all curb candidates within the block according to their elevation. The lowest points on each side of the block are regarded as the bottom point of curbs. Figure 3.8 illustrates the principle of the proposed curb extraction method. The red points are curb candidate points in local blocks, and the blue points mean the lowest curb candidate points on each side of the block.

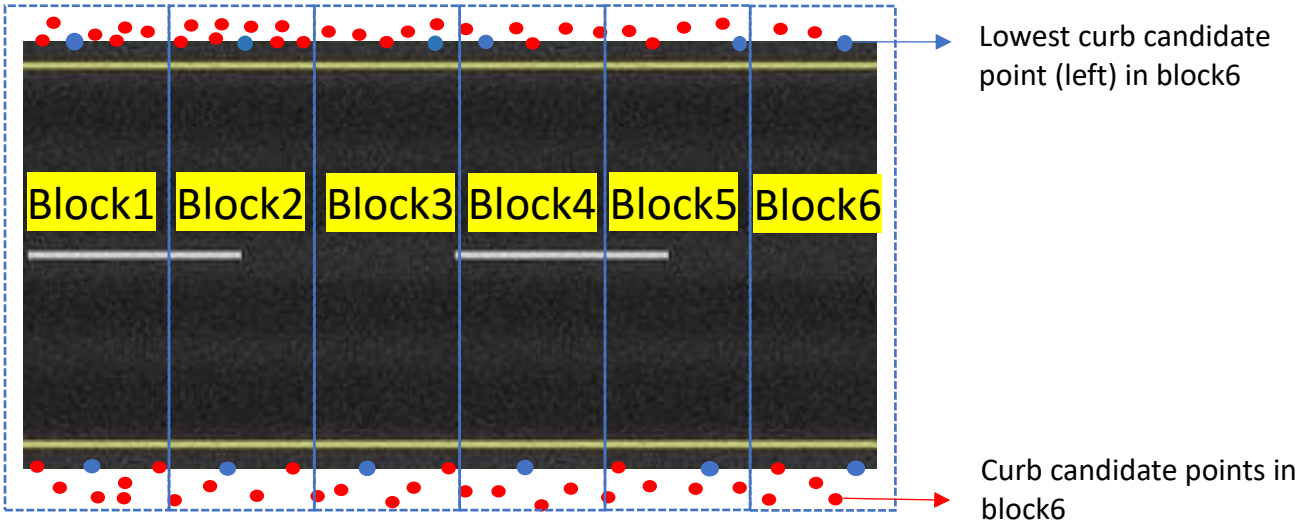


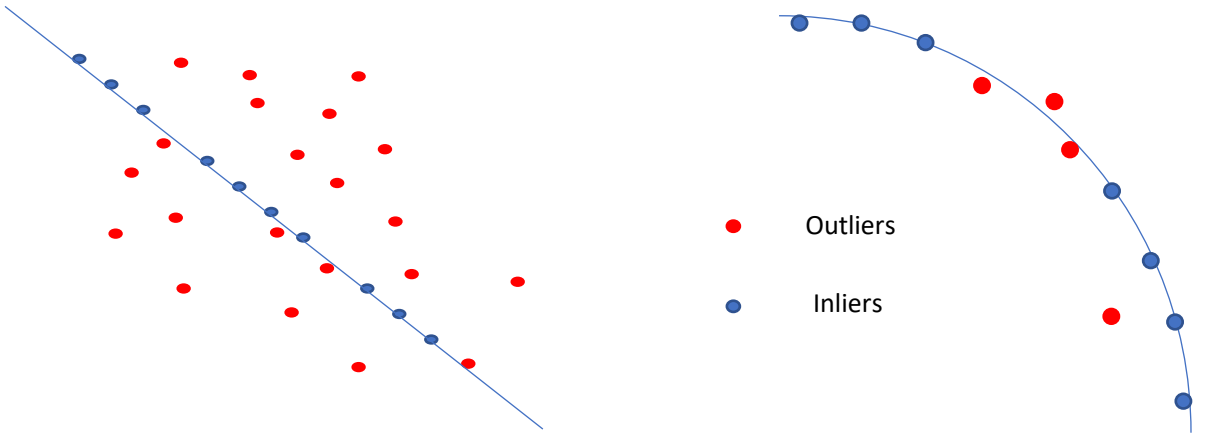

Figure 3.8 Principle of curb extraction method

**(3) Road Edge Refinement**

Different from most existing methods, in this study, the position of lane marking points are determined by calculating the distance to the extracted road edges. Therefore, the accuracy of the extracted curb points is of great significance. All of the extracted curb points are expected to be the bottom of curbs or guardrails; however, it is inevitable that some extracted points have slight deviation. Although most of the offsets are very small, the extracted curb points still need to be refined, as the performance of lane marking extraction largely relies on the accuracy of the extracted curb points. Thus, a road edge refinement algorithm based on RANSAC is proposed to remove outliers.

RANdom Sample Consensus (RANSAC) was presented by Fischler and Bolles (1981), which is an iterative algorithm that can be used to extract shapes and estimate parameters of a mathematical model from a set of data with outliers. The RANSAC algorithm achieves this goal by randomly and iteratively drawing a subset from the raw data. The selected minimal set are regarded as hypothetical inliers, and a model is defined based on these inliers. In addition, all parameters of the model are determined by the inliers. The estimated model or shape is tested against all other data to determine how many points also fit well to the candidate model, and if a point is well approximated by the model, it is considered as an inlier. A reasonably good model should contain sufficient points as inliers. To refine the candidate model, the parameters of the model are re-estimated when a series of points are classified as new inliers. Finally, to evaluate the model, a score is calculated according to the errors of inliers. As a result, models with insufficient inliers are firstly rejected, followed by models with lower scores. This procedure is repeated and after a fixed number of trials, the model which contains the most inliers and has the highest score

is extracted with its inliers. Figure 3.9 (a) shows an example of RANSAC linear model fitting. Figure 3.9 (b) illustrates the RANSAC curvilinear model fitting.




(a) RANSAC linear model fitting

(b) RANSAC curvilinear model fitting

Figure 3.9 An illustration of RANSAC

RANSAC is a robust algorithm that has been proved in many research. It has the ability to estimate parameters of a model with a high accuracy. More importantly, it can handle data containing a significant number of outliers. The robustness of RANSAC makes it a popular algorithm, which has been used in a wide range of applications. Nevertheless, a disadvantage of RANSAC is that there is no upper limit on the computational time to estimate the optimal model. It will keep searching for better models until a user-defined limitation number is reached. In this thesis, as the number of the extracted curb points is usually no more than 300 with no or few outliers, the upper bound of iterations is set at 1000. Figure 3.10 shows a demonstration of road edge refinement.

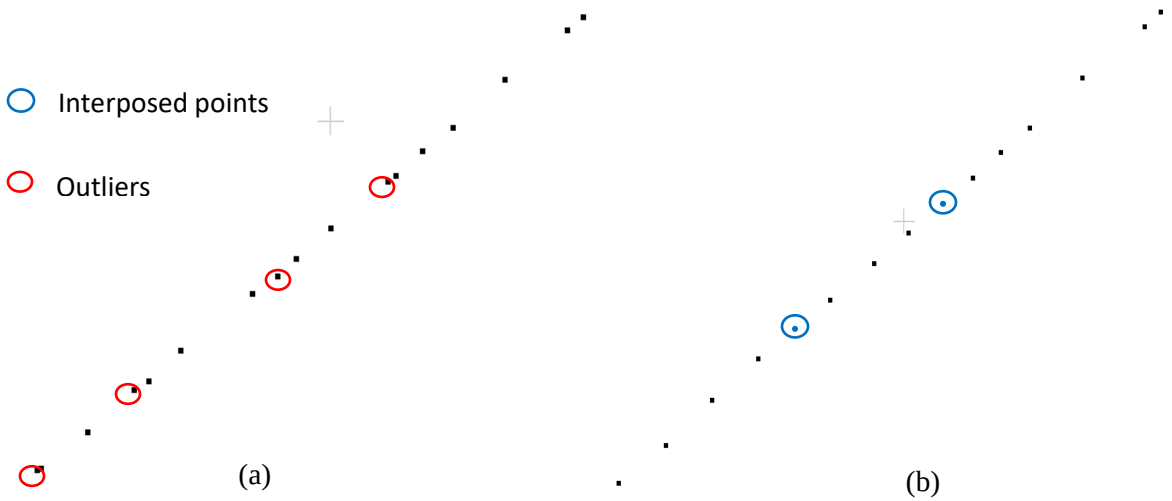



Figure 3.10 A demonstration of road edge refinement

(a) original curb points (17 points); (b) the refined curb points (15 points)

As shown in Figure 3.10, the black dots are curb candidate points displayed in CloudCompare v2.6.3. Four outliers (points in red circles) are removed from the extracted curb candidate points after this procedure. It is noteworthy that if curb points are not sufficient after refinement, a point interpolation method is applied to interpose points between consecutive curb points. The midpoints of any two consecutive curb candidates could be regarded as new curb candidates until a pair of curb points can be found in every local block. The detailed illustration of point interpolation method will be described in Section 3.6.

Lastly, for straight roads, different linear models are used to fit the extracted road edge in both XY plane and 3D space based on the RANSAC algorithm. In the XY plane, the linear equation is described as:

$$y = kx + b \tag{3-7}$$

where k denotes the slope of the line, and b is the y-intercept. In the 3D space, the parametric form of the linear equation is used to depict the line:

$$\begin{cases} x = x_0 + mt \\ y = y_0 + nt \\ z = z_0 + pt \end{cases} \tag{3-8}$$

where $P_0$ ($x_o$, $y_o$, $z_o$) is a point that is on the line, vector $\vec{v}$ (m, n, p) is the directional vector that is parallel to the line, and t is a multiplier denoting how far from the original point that the point moves. As t varies over all possible values the line can be completely covered. The extracted road edge and the fitted linear model are projected onto the XY plane and the 3D space using MATLAB. Figure 3.11 presents an example of road edge fitting.

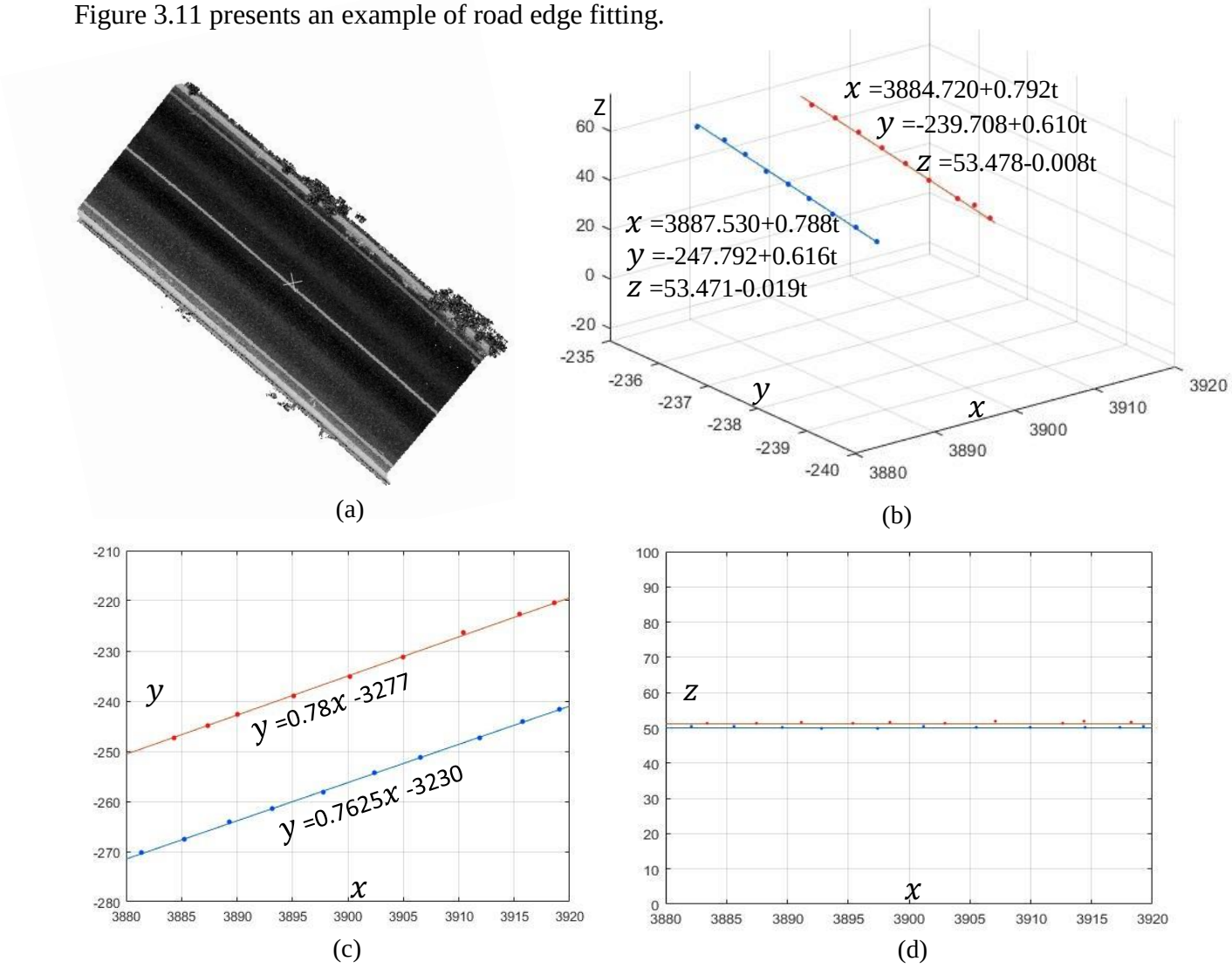
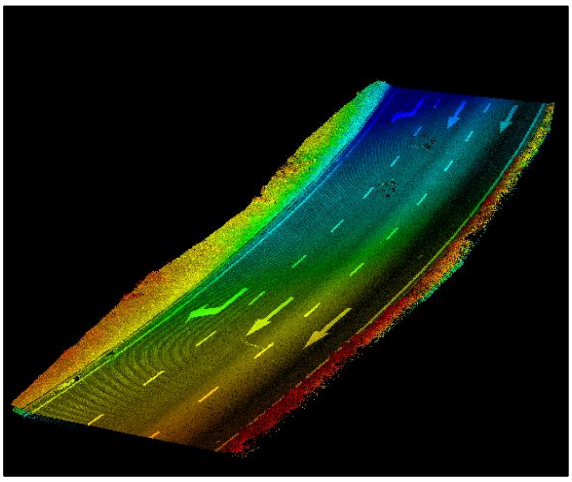


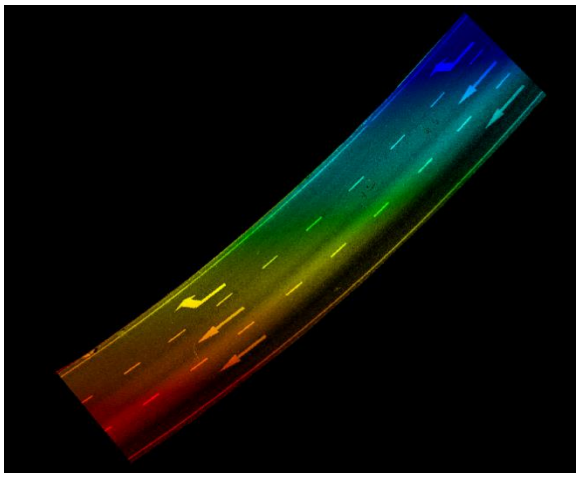

Figure 3.11 Example of road edge fitting (a) Dataset 1; (b) road edges in 3D scene; (c) the fitted road edge in XY plane; (d) the fitted road edge in XZ plane

**(5) Road Surface Extraction**

Finally, two curb points are extracted in each block with one on each side. Based on the coordinates of these curb points, linear functions that passes any two consecutive curb points can be calculated. By this analogy, any two consecutive curb points will generate a local edge line. Then in each block, the points located in the inner side of the corresponding edge line are determined as road surface and extracted. An example of the road surface segmentation is shown in Figure 3.12. Figure 3.12 (a) shows the ground points before road surface segmentation. Figure 3.12 (b) illustrates the extracted road surface.



(b) The ground points



(a) the extracted road surfaces

Figure 3.12 Example of road surface segmentation

## 3.5 Road Marking Extraction

Different road markings have different characteristics, such as width, length, shape, color, and position. Therefore, in this study, two algorithms are developed with each focusing on different types of road markings. On the one hand, a clustering algorithm based on morphological analysis is developed to extract directional and textual road markings, such as arrows, symbols, and words. On the other hand, a distance-to-road-edge thresholding algorithm based on road design standards

is used to extract lane markings, such as solid and broken lines. Figure 3.13 presents the workflow of the proposed road marking extraction methodology.

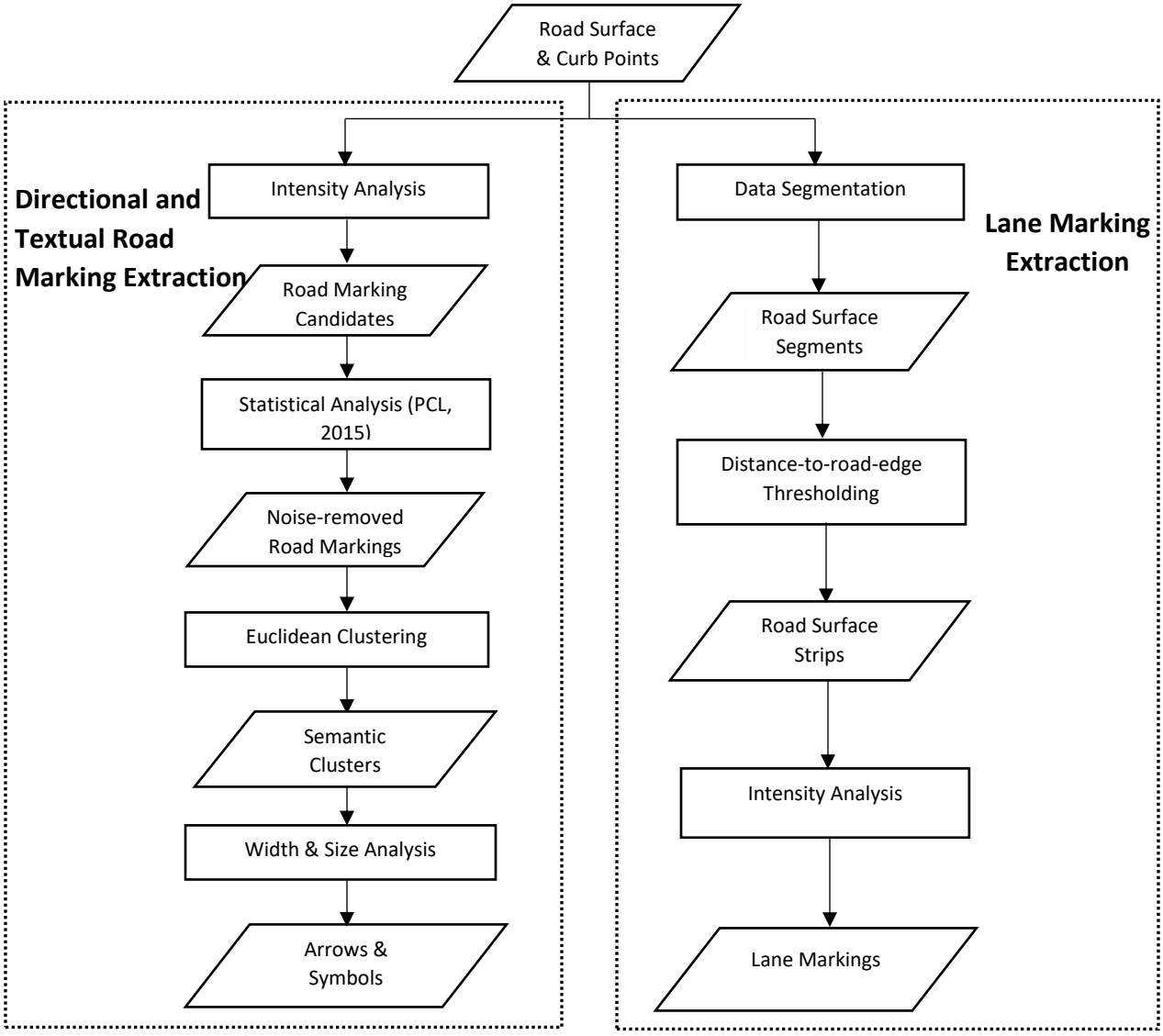



Figure 3.13 Workflow of road marking extraction methodology

### 3.5.1 Directional and Textual Road Marking Extraction

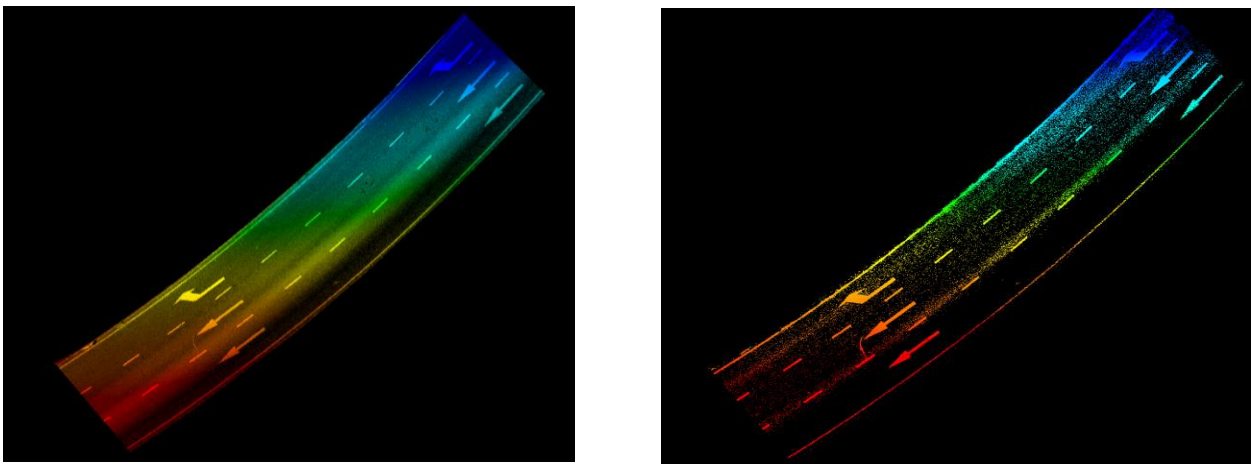
With the extracted road surface points, a directional and textual road marking recognition framework is proposed in this study, which includes three steps: (1) intensity analysis; (2) outlier removal based on statistical analysis; and (3) conditional Euclidean clustering.

**(1) Intensity Analysis**

As the wavelength of laser scanner is near-infrared, road markings have higher reflectance than the unpainted road surface. The intensity information is the key criterion that can be used for road marking detection. Therefore, an intensity thresholding algorithm is applied first, which recognizes road markings based on intensity information. The observation is defined as:

$$P_i \begin{cases} Road\ marking\ candidate, & if\ \ (I_{min} \le I_i \le I_{max}) \\ non-road\ marking\ point, & otherwise \end{cases} \tag{3-9}$$

where $I_i$ is the intensity value of a point. $I_{min}$ and $I_{max}$ are the minimum and maximum intensity thresholds, respectively. Once a point's intensity value is within the given range, the point is regarded as a road marking candidate. However, point intensity largely depends on the scanning range and the incidence angle of the laser beam, contributing to a phenomenon that point intensity gradually fade from the vehicle trajectory to its two sides (Guan et al., 2015). In consequence, local optimal thresholds are required to be adaptively estimated. Specifically, $I_{min}$ is usually set at small values in order to avoid the error of omission. Figure 3.14 (a) shows the road surface before intensity thresholding. Figure 3.14 (b) shows the extracted road markings by intensity filter.



(a) The road surfaces (b) the extracted road markings

Figure 3.14 Example of road marking extraction

**(2) Outlier Removal based on Statistical Analysis**

As shown in Figure 3.14, noise inevitably exists in MLS point clouds after road marking extraction. The noise may complicate the estimation of local point cloud characteristics, leading to erroneous values, which in turn might cause point cloud registration failures (PCL, 2015). In this study, most of the noise are isolated points and point mutations in some local areas. Thus, an outlier removal function is applied to remove outliers from the extracted road markings. The proposed outlier removal function is based on a statistical analysis filter in the PCL Library (PCL, 2015), which is a large-scale open source C++ programming library. The PCL keeps absorbing valuable algorithms in the field of LiDAR data processing every year, and the most recent version is PCL library version 1.8.0, which was released on June 17, 2016. The new PCL library can support multiple platforms with a variety of functions including data acquisition, filtering, registration, visualization, etc.

To identify outliers, the algorithm firstly attempts to find the k neighbour points from a certain point of interest. k is a user-defined threshold according to average point density. If no enough neighbors can be found within a stated radius, the point is regarded as an outlier and removed from the point cloud. Furthermore, if the k nearest points of the interested point are founded in the certain area, the mean distance from the point to its neighbors will be calculated. The algorithm assumes that the distribution of mean distance of all points should follow Gaussian distribution, and the points outside an interval defined by the global distances mean and standard deviation can be considered as outliers and trimmed from the dataset. Figure 3.15 (a) presents the principle of statistical analysis on points' neighbourhood, and Figure 3.15 (b) indicates the mean k-nearest neighbour distance.

.

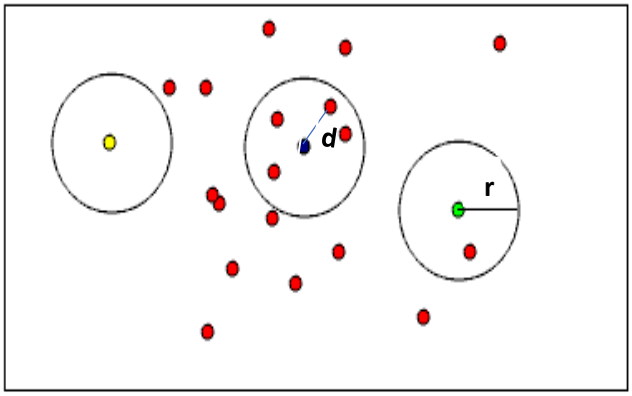
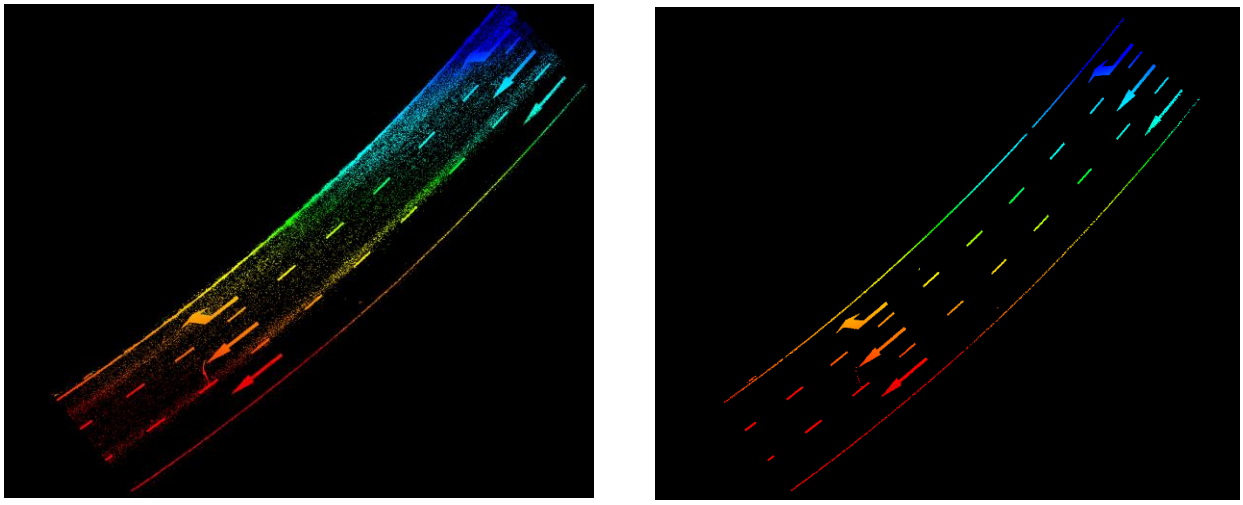



Figure 3.15 Noise removal using statistical analysis (PCL, 2015)

Figure 3.15 helps to visualize what the outlier removal function does. The user specifies a number k denoting k neighbors is required for every point within a specified radius to remain in the point Cloud. For example, if 1 neighbor is specified, only the yellow point will be removed from the point cloud. If 2 neighbors are specified then both the yellow and green points will be removed from the point cloud, and then the mean distance to neighbours will be calculated for the remaining points. Filtering out the noise can reduce the complexity of the remaining data and enhance the efficiency. Figure 3.16 (a) shows the extracted road markings by intensity filter. Figure 3.16 (b) shows the road markings after outlier removal.



(a) Road markings before outlier removal

(b) Outlier removal results

Figure 3.16 Example of outlier removal

**(3) Conditional Euclidean Clustering**

After outlier removal, road markings are primary components of the renaming data. However, points belonging to the same object are still isolated and unorganized, and there are no topological relationships between points. In other words, the extracted 3D scene is still comprised of millions of sparse and discrete points with no semantic information. Therefore, to distinguish specific objects (e.g. arrows, straight lines, etc.) and organize discrete points into semantic groups, a revised conditional Euclidean clustering method (PCL, 2015) is conducted. Figure 3.17 presents the principle of conditional Euclidean clustering method.

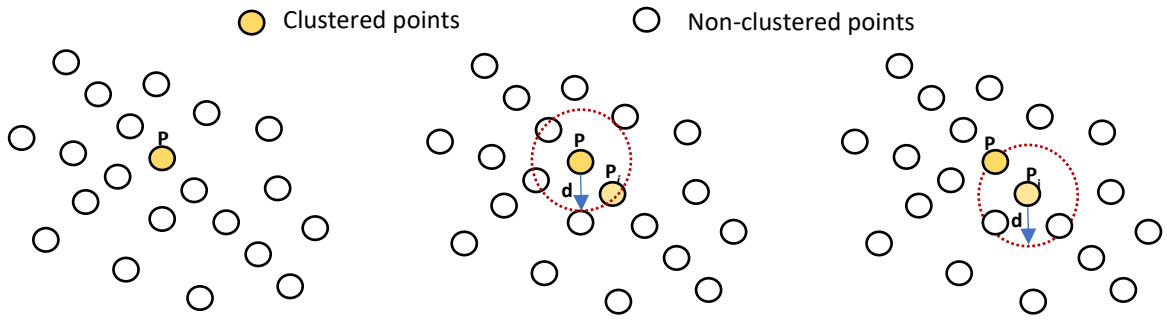



Figure 3.17 Conditional Euclidean clustering process

This method attempts to divide sparse points into different clusters according to the Euclidean distance between a certain point and its nearest points. Basically, the points will be designated into the same cluster if the Euclidean distance between them is less than the given threshold ($d_c$), which is determined by the point density and resolution of the test data. The detailed principle of the revised conditional Euclidean clustering method is described below:

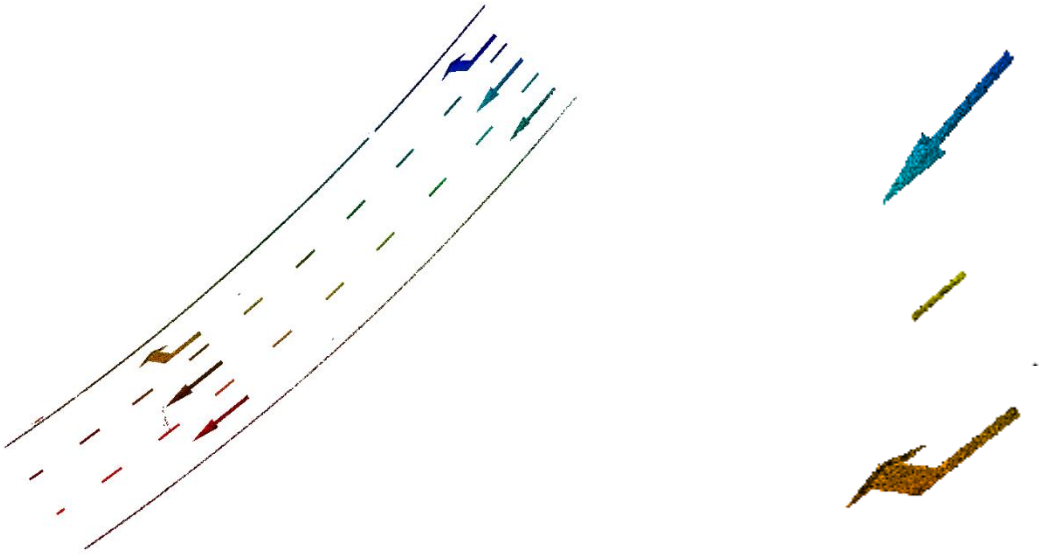
(1) Initially, the algorithm randomly selects a point P as starting point, which is also firstly classified as a clustered point. Meanwhile, an empty list of clusters C is created, as well as a queue of the points that need to be checked Q.

(2) Next, the algorithm searches for point neighbours ($P_i$, i = 1, 2, …, n) of P in a sphere with radius $d_{th}$, and adds $P_i$ to the current queen Q. For every point neighbour $P_i$, the Euclidean distance form Pi to P is calculated. Points whose Euclidean distance is less than the given threshold $d_c$ are added to the list of clusters C. If all of the point neighbors have been processed, reset Q to an empty list.

(3) The new clustered points are regarded as new starting points, and the same procedure is repeated. When no more non-clustered points can be found within the sphere, reset the list of clusters C, and randomly select a new starting point from the rest of non-clustered points.

(4) The algorithm terminates when all points have been processed.

When all of the points have been processed, the discrete and sparse points are successfully segmented into different semantic clusters. Figure 3.18 (a) presents the extracted road marking points. Figure 3.18 (b) presents three different semantic groups clustered by conditional Euclidean clustering.



(a) The extracted road markings (b) the semantic clusters

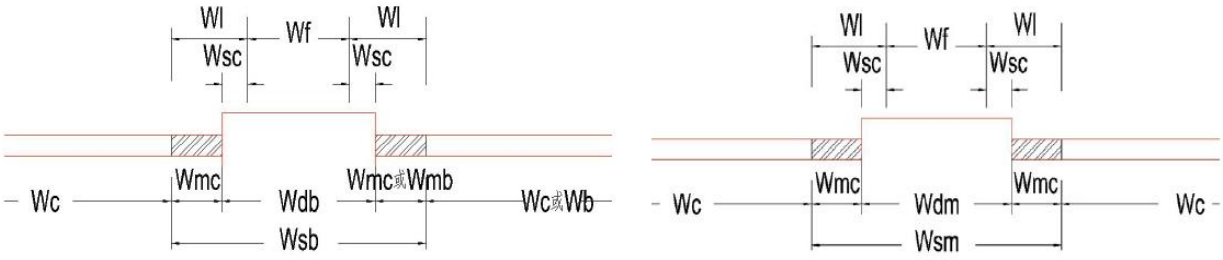
Figure 3.18 Example of conditional Euclidean clustering

Based on the prior knowledge (e.g., height, size, width) of road markings, relatively wide clusters (arrows and words) can be distinguished by removing small clusters (straight lines and noise). A width threshold denoting the difference between maximum y value and minimum y value is applied to filter out unwanted clusters.

### 3.5.2 Lane Marking Extraction

Different from directional and textual road markings like arrows and words, lane markings are extracted by local intensity thresholding and distance-to-edge thresholding, as more precise lane marking extraction results are required to provide lane-level navigation. With refined road edges as reference, the position of lane markings can be determined by calculating the distance from the point to road edges. The thresholds used in this algorithm are determined by road design standards, which regulates road marking width, lane width, marginal strip width, and shoulder width.

Different countries have their own road design standards. Since the study area of this thesis is Xiamen, China, road design standard of China is used in this study to determine the thresholds for the proposed method. Figure 3.19 presents the structure of urban median strip which is recorded in *the Code for Design of Urban Road of China* (CJJ37 -2012).




(b) Central strip (c) Left and right strip

Figure 3.19 Structure of median strips

As shown in Figure 3.19, Wmc denotes the width of marginal strip, Wsc is the safety clearance, Wl is the lateral clearance, and Wc means the lane width. Marginal strip denotes the space between the outer edge of lane markings to curbs. The main role of marginal strips is to attract drivers' attention and remind the lateral clearance. Usually, a 0.5m-marginal-strip should be set on the right side of highways and first-class roadways. The width of marginal strips on the left side of expressways and first-class roads should range from 0.5m to 0.75m. In addition, marginal strip is a component of hard shoulder or dividing strip. Table 3.3 provides detailed regulation of these parameters which are translated from *the Code for Design of Urban Road of China (CJJ37 -2012).*

Table 3.3 Minimum width of road dividing strip

| Type | | Median Strip | | Left and Right Strip | |
|---|---|---|---|---|---|
| Speed Limit (km/hr) | | ≥60 | <60 | ≥60 | <60 |
| Marginal Strip Width | Motorway (m) | 0.50 | 0.25 | 0.50 | 0.25 |
| | Bicycle lane (m) | ____ | ____ | 0.25 | 0.25 |
| Safety Clearance | Motorway (m) | 0.50 | 0.25 | 0.25 | 0.25 |
| | Bicycle lane (m) | ____ | ____ | 0.25 | 0.25 |
| Lateral Clearance | Motorway (m) | 1.00 | 0.50 | 0.75 | 0.50 |
| | Bicycle lane (m) | ____ | ____ | 0.50 | 0.50 |
| Minimum width of divider island (m) | | 2.00 | 1.50 | 1.50 | 1.50 |
| Minimum width of dividing strip (m) | | 3.00 | 2.00 | 2.50 (2.00) | 2.00 |

As shown in Table 3.3, lateral clearance is the sum of marginal strip and safety clearance. In addition, the width of marginal strip should not exceed 0.75 m and is included in the width of shoulder. In traffic terminology, shoulder denotes the space between roadway and ditch. The width

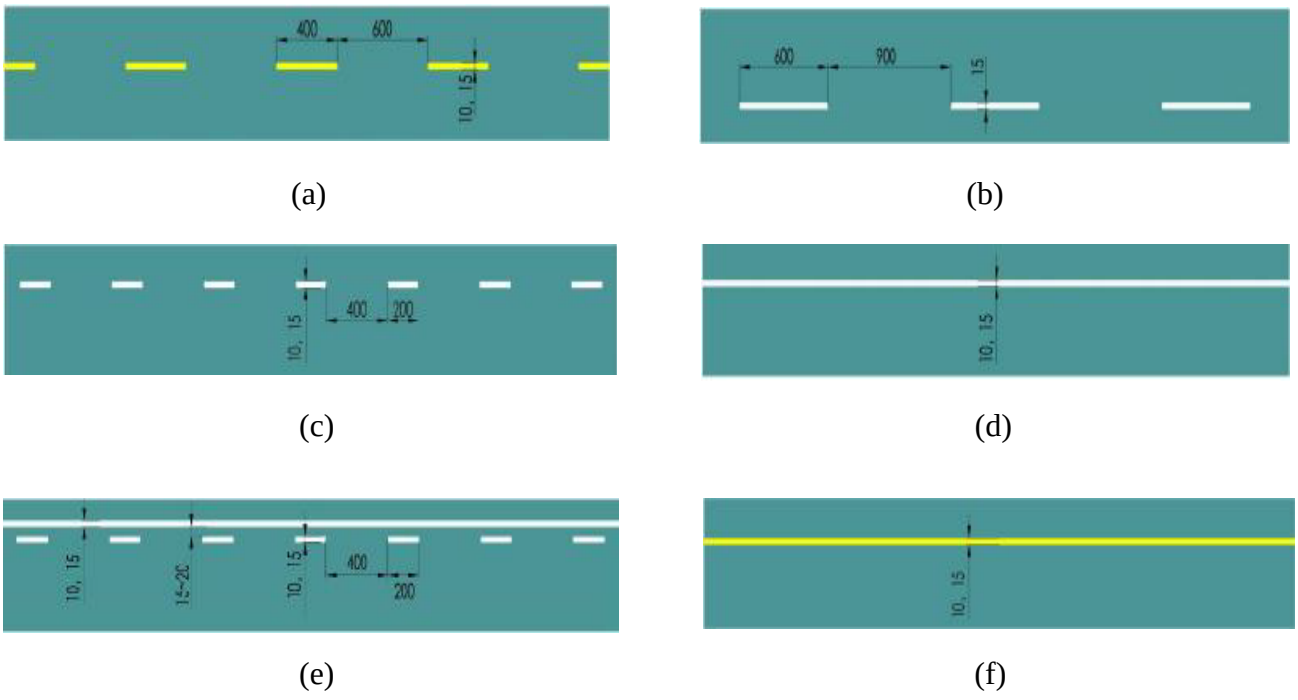
of protective shoulder on expressways should be less than 0.75m. Table 3.4 presents the width of shoulders for different types of roads recorded in *the Code for Design of Urban Road of China (CJJ37 -2012).* Figure 3.20 illustrates the position of shoulder and marginal strip.

Table 3.4 Shoulder width of different road types

| Speed Limit (km/hr) | | Highway & First-class road | | | | ($2^{nd}$, $3^{rd}$, $4^{th}$, $5^{th}$ ,$6^{th}$)-class road | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 120 | 100 | 80 | 60 | 80 | 60 | 40 | 30 | 20 |
| Width of hard shoulder on right side (m) | Recommended value (m) | 3.50 | 3.00 | 2.50 | 2.50 | 1.50 | 0.75 | — | — | — |
| | Minimum value (m) | 3.00 | 2.50 | 1.50 | 1.50 | 0.75 | 0.25 | | | |
| Width of soil shoulder (m) | Recommended value (m) | 0.75 | 0.75 | 0.75 | 0.50 | 0.75 | 0.75 | 0.75 | 0.50 | 0.25<br>0.50 |
| | Minimum value (m) | 0.75 | 0.75 | 0.75 | 0.50 | 0.50 | 0.50 | | | |


Figure 3.20 Position of shoulder and marginal strip

Furthermore, road design standards also regulate and record the length and width of lane markings. For example, according to *the Code for Layout of Urban Road Traffic Signs and Markings of China (GB 51038 – 2015)*, the width of single yellow dashed line should be 15 cm. The marking length and the distance between two consecutive line segments are 4 m and 6 m, respectively. Figure 3.21 presents some regulated width (cm) of different types of lane markings.




(a) (b)

(c) (d)

(e) (f)

Figure 3.21 Standards for road markings

(a) Single yellow dashed line (separating lanes in one direction); (b)(c) single white dashed line (separating lanes in two direction); (d)(e)(f) roadway edge lines

It is noteworthy that the width of different lane markings depends on the designated road speed limits. In this study, the speed limit of Huandao Road and Yanwu Bridge is 60 km/hr. Thus, according to *the Code for Layout of Urban Road Traffic Signs and Markings of China (GB 51038 – 2015)*, the width of these lane markings should be 15 cm. With these road design standards as priori knowledge, optimal local thresholds can be easily estimated. Figure 3.22 shows the field

surveys in Xiamen. The principle of the proposed distance-to-road-edge thresholding method is illustrated in Figure 3.23.


Figure 3.22 Field surveys in Xiamen

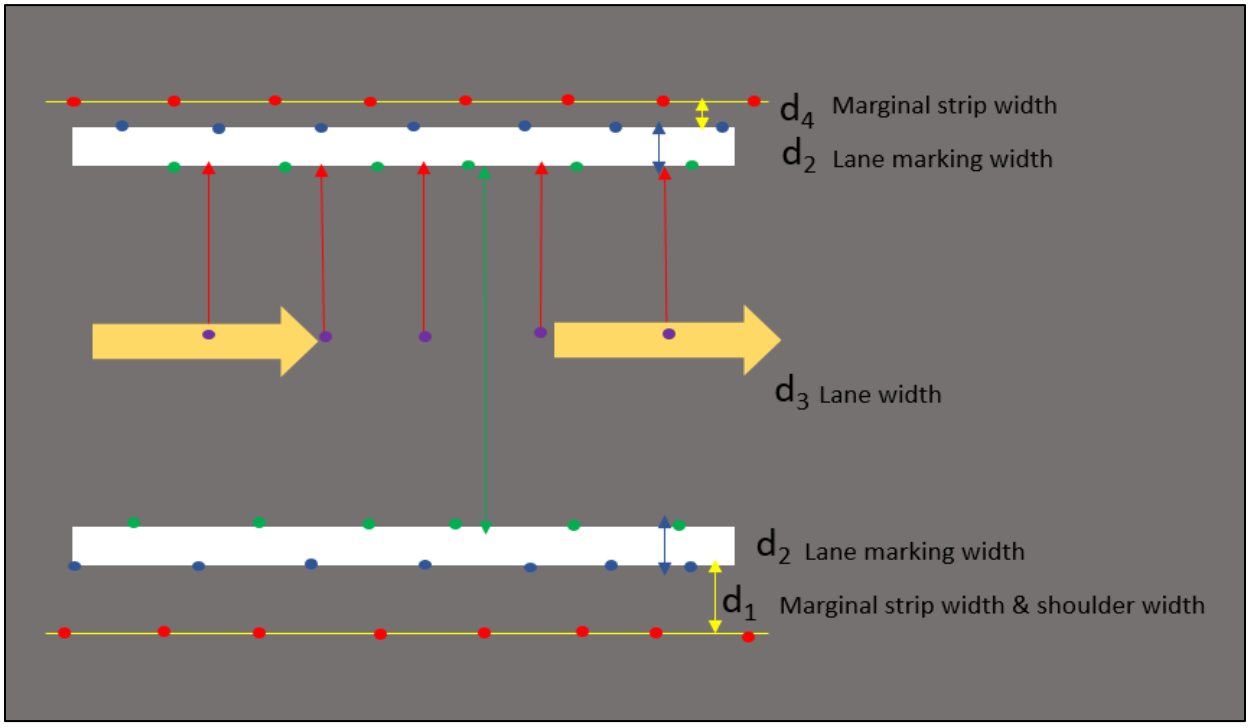



Figure 3.23 Principle of distance thresholding

As shown in Figure 3.23, the proposed algorithm detects lane markings and roughly estimates lane centerlines based on the width information recorded in *the Code for Layout of Urban Road Traffic Signs and Markings of China (GB 51038 – 2015)*, and *the Code for Design of Urban Road of China (CJJ37 -2012)*. The algorithm initially regards the refined curb points (red points) as control points and segments data into a significant number of blocks. Similar to the segmentation method mentioned in Section 3.3, every two consecutive curb points will generate one block, and a linear function is calculated based on the coordinates of the two consecutive curb points. This segmentation strategy enables the algorithm to process a minimal set of data in local blocks, thus reducing the influence of errors on the global scale. After segmentation, a searching window starts moving from the two curb points in each local block and searches for points whose distance to the generated edge line is within a certain range. Meanwhile, a point-intensity-thresholding is applied to detect points with high retro-reflectance in certain strips. The criteria for detecting lane markings can be described as follows:

$$P_i \begin{cases} Lane\ marking\ point, & if\ \ (I_{min} \le I_i \le I_{max}\ \&Dmin < D_i < Dmax) \\ non - Lane -\ marking\ point, & otherwise \end{cases} \quad (3\text{-}11)$$

where $I_i$ is the intensity value of a point. $I_{min}$ and $I_{max}$ are the minimum and maximum intensity thresholds, respectively. $D_i$ is the distance from the point to the generated road edge in the local block. $D_{min}$ and $D_{max}$ are minimum and maximum distance thresholds which are defined according to road design standards. For example, as shown in Figure 3.23, $D_{min}$ is usually set as the width of the marginal strip (d4) for the left side, and the shoulder width (d1) for right side. $D_{max}$ equals to the sum of the marginal strip width (d4) and the lane marking width (d2).

Generally, marginal strip width ranges from 0.5 m to 0.75 m for urban roads, and shoulder width ranges from 1.5 m to 2.5 m for the test roads (as shown in Table 3.4). Furthermore, the width

of lane markings in the study area is 15 cm. Thus, $D_{min}$ and $D_{max}$ are usually set at 0.5 m and 0.9 m respectively. Additionally, lane width should not be less than 3.5 m for urban roads with a speed limit of 60 km/hr.

Consequently, points with certain intensity values and within certain regions are extracted as lane markings. The nearest and farthest points on lane markings are regarded as the outer edge and inner edge of the lane marking and separately extracted. Finally, lane centerline points are roughly estimated by finding points with a certain distance (d2 + d1 + d3/2) to the right road edge. The equation of the distance from a point to a line is described as follows:

$$d = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}} \tag{3-12}$$

where d is the distance from a point to a line, (x0, y0) represents the 2D coordinates of a point, and a, b, and c are the parameters of a linear function which is estimated by two consecutive curb points in local blocks.

This method can produce promising results, especially for straight roads. It should be noted that when it comes to curves, partitioning data into smaller pieces is a useful strategy to improve the performance of the proposed method. The smaller the length of each block, the more similar the shape of the rectangle and its corresponding arc. When the area of the rectangle is approximately zero, the sides of the rectangle and the arc are almost coinciding. Figure 3.24 shows the proposed method applied to curve roads. Figure 3.24 (a) shows a curved road section segmented into 4 blocks (red rectangle). Figure 3.24 (b) shows an 8-block segmentation result of the same curve, in which, red points are the extracted curb points, yellow lines are the local road edge lines connected by consecutive curb points, blue points denote the detected lane marking

points. As shown in Figure 3.24, smoothed lane marking extraction results could be obtained by partitioning data into smaller pieces.

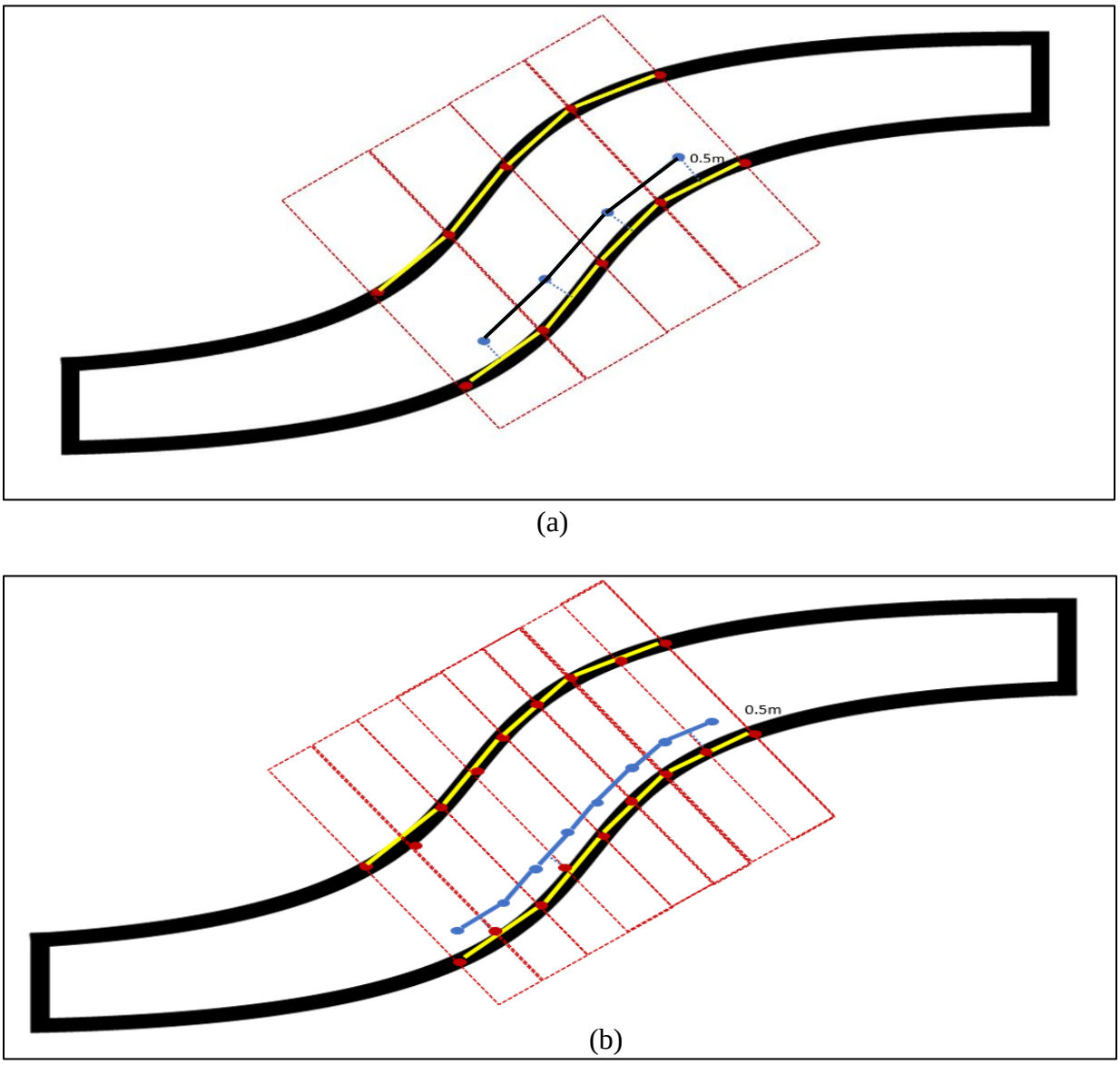



Figure 3.24 Extracting lane markings in curves using different thresholds

(a) 4 blocks; (b) 8 blocks

### 3.6 Lane Centerline Estimation

As shown in Figure 3.23, lane centerlines are roughly estimated by finding points with a certain distance to right road edges. In order to acquire more precise lane centerlines which are suitable for autonomous navigation, the lane centerline is estimated again based on the position of lane markings. To calculate the coordinates of lane centerline points, a same segmentation strategy as mentioned in Section 3.3 is applied here. The points of the extracted lane markings are segmented into hundreds of blocks, and in each block, the center points of lane markings on each side are extracted. The coordinates of center points in each block are calculated by:

$$C\ (X_C, Y_C, Z_C) \approx \left(\frac{X_{max}+X_{min}}{2}, \frac{Y_{max}+Y_{min}}{2}, \frac{Z_{max}+Z_{min}}{2}\right) \qquad (3\text{-}10)$$

where $X_{max}$, $X_{min}$, $Y_{max}$, $Y_{min}$, $Z_{max}$, $Z_{min}$ are the maximum values and minimum values of x, y, and z for each cluster in the local block, respectively. Two center points are extracted in each local block with one on each lane-marking cluster. Similarly, the coordinates of lane centerline points are estimated by calculating the coordinate mean of two extracted center points in every local block.

Generally, a point cloud is segmented into hundreds of blocks, thus producing the same number of lane centerline points. However, for broken lines and faded lane markings, a center point interpolation method is applied to interpose center points in blank areas until sufficient lane centerline points can be generated. Figure 3.25 presents the principle of the proposed lane centerline estimation method. Figure 3.26 shows a demonstration of lane centerline estimation.

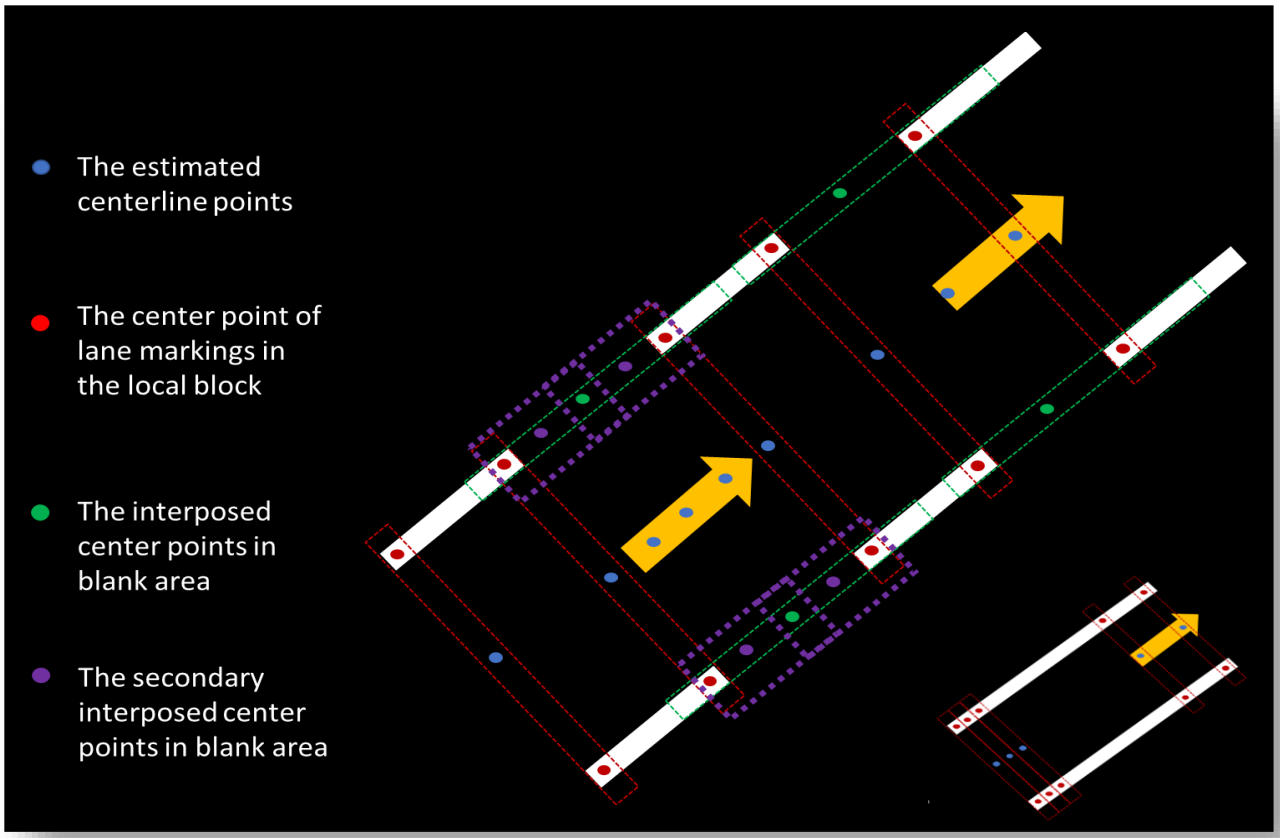
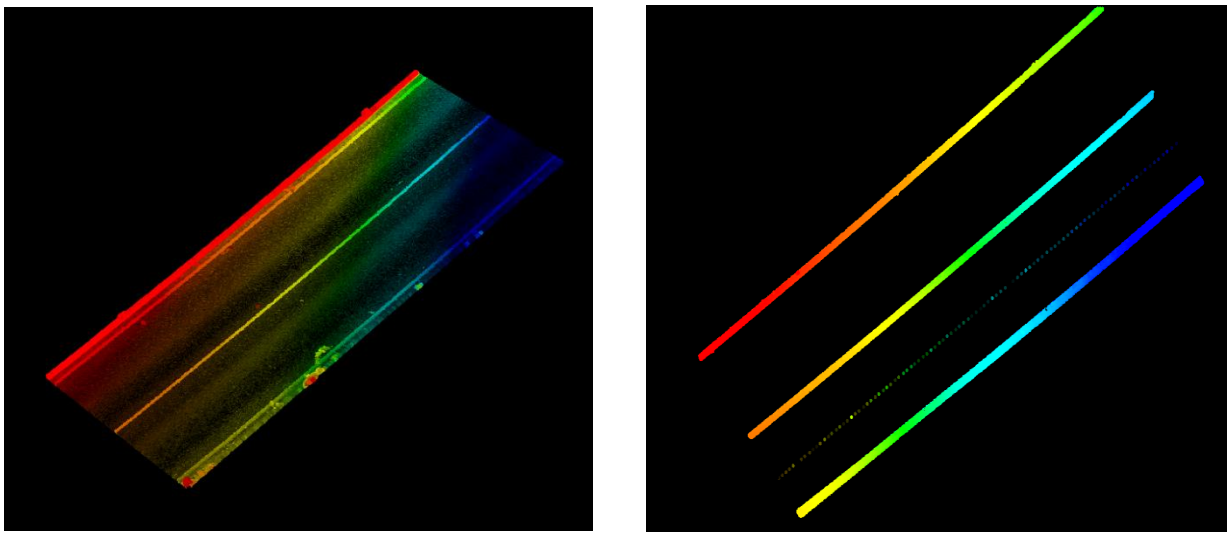



Figure 3.25 Principle of lane centerline estimation

With the estimated lane centerline points as position vectors, the optimal path for autonomous vehicles can be generated. Combined with road edges and a variety of road marking clusters, a 3D high-definition road map is created. The extracted key points are linked in ArcGIS to produce smooth road edges and lane centerlines. The final experimental results will be presented and compared in chapter 4, as well as the performance and comments of the proposed algorithms.



(a) (b)

Figure 3.26 Example of lane centerline estimation

(a) The extracted road surface; (b) the estimated lane centerline points

## 3.7 Accuracy Assessment

### 3.7.1 Accuracy Assessment of Road Markings

After extracting all useful information, an accuracy assessment is implemented to evaluate the performance of the proposed method. A set of reference pixels for each dataset is manually labeled in UAV orthoimages using ArcGIS v10.2.2. Also, the MLS point clouds are aligned to the same coordinate system (WGS84) of UAV orthoimages. With UAV orthoimages, the extracted road markings are overlapped with UAV images in ArcGIS. After pixel analysis, precision, recall, and $F_1$-score are used to represent the accuracy of the experimental results. As shown in the following equations, precision represents how many relevant road marking pixels are extracted. Recall indicates how many extracted road marking pixels are correct. $F_1$-score is a measure that combines precision and recall. It is also the harmonic mean of precision and recall in mathematics.

$$Precision = \frac{tp}{tp+fp} \tag{3-11}$$

$$Recall = \frac{tp}{tp+fn} \tag{3-12}$$

$$F_1 - score = \frac{2*Precision*Recall}{Precision+Recall} \tag{3-13}$$

As shown in Eqs. (3-11) to (3-13), tp indicates number of true positive; fp and fn represent number of false positive and negative, respectively. In this thesis, the targeted road markings are regarded as positive, and the rest of classes are negative.

### 3.7.2 Accuracy Assessment of Lane Centerlines

In this study, lane centerlines are extracted as position vectors for autonomous navigation systems. The coordinates of these lane centerlines are estimated based on the position of lane markings. As they are simulated and unreal lines without obvious characteristics in 3D scenes, there are no available reference point clouds that can be used to assess the accuracy of the estimated lane centerlines. Therefore, the estimated lane centerlines are overlapped with UAV orthoimages in ArcGIS. Buffer zones with the width of 10~30 cm are manually drawn in base maps as reference, and the experimental results are evaluated by observing whether the estimated lane centerlines are in the buffer zones. If not, the offset should be calculated. The evaluation results of road marking extraction and lane centerline estimation will be described in Chapter 4.

## 3.8 Chapter Summary

This chapter introduced the study area, datasets, computing platforms, and relative software used in this study. The rationales of every step in the approach have been presented explicitly. First, the coordinate system transformation is performed to fix the orientation of the vehicle frame. Then, the non-ground points are removed by the voxel-based upward growing

algorithm. With ground points, curb candidates are detected by their elevation difference and slope, and road surfaces are segmented based on the position of the extracted curb points. As to road marking extraction, on the one hand, arrows, words, and symbols are extracted by intensity thresholding, outlier removal and conditional Euclidean clustering. On the other hand, lane markings are extracted based on intensity and distance-to-road-edge thresholding. This chapter provides prototype procedures for extracting different meaningful road features from massive MLS point clouds and creating 3D high-definition road maps for autonomous vehicles. The accuracy assessment mechanism is also described in this chapter. In the next chapter, the experimental results will be presented, as well as the accuracy assessment results.

# Chapter 4

# Results and Discussion

This chapter presents the experimental results of the proposed method. Section 4.1 presents the results of every individual procedure, including road surface extraction, non-ground points removal, outlier removal, Euclidean clustering, road marking extraction, and lane centerline estimation. Section 4.2 demonstrates the accuracy assessment results. Section 4.3 discusses the computing complexity of the proposed algorithms. Two comparative studies are described in section 4.4. Section 4.5 summarizes the main results of this chapter.

## 4.1 Results and Evaluation

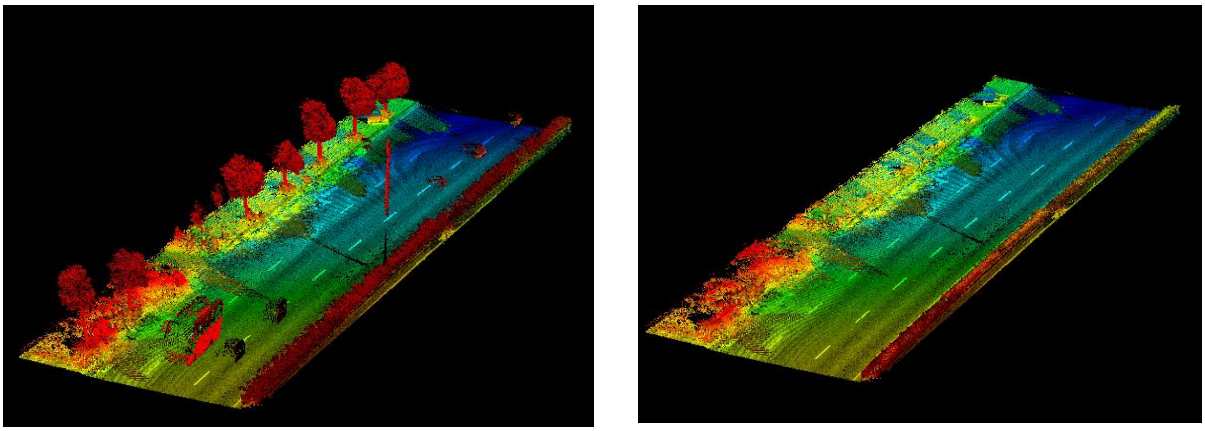
### 4.1.1 Non-ground Point Removal

As mentioned in Section 3.3.2, a voxel-based upward-growing method is adopted to remove non-ground points from raw datasets. Three parameters have great influence on the performance of the voxel-based upward-growing algorithm, which are: size of a voxel ($W_v$), local ground undulation threshold ($h_g$) and global ground undulation threshold ($h_e$). In this study, according to the point density and characteristics of test datasets, these parameters were set at different values. For Dataset 1, Dataset 2, and Dataset 4, the voxel size $W_v$ was set as $W_v = 0.4$ m, $h_g = 0.7$ m, and $h_e = 3.0$ m, respectively. For Dataset 3, $W_v$ was set at 0.1 m, $h_g = 0.8$ m, $h_e = 3.0$ m. As Dataset 5 has a steep slope, the voxel size was set at 0.05 m, $h_g = 0.7$ m, and $h_e = 3.0$ m. Figure 4.1 presents the non-ground removal result of Sample 7. Figure 4.1 (a) shows the raw point cloud of Sample 7. Figure 4.1 (b) presents the non-ground point removal results. Table 4.1 lists the non-ground removal results of the five test datasets and the proportion of non-ground points, which

proves that the computing efficiency of the proposed method can be considerably enhanced by removing non-ground points from MLS data.

Table 4.1 Results of the non-ground removal

| Dataset | Raw point cloud | Non-ground points | Percentage of non-ground points |
|---|---|---|---|
| 1 | 1,920,753 | 576,226 | 30% |
| 2 | 341,891 | 57,583 | 17% |
| 3 | 6,758,030 | 770,065 | 11% |
| 4 | 9,058,578 | 1,108,085 | 12% |
| 5 | 8,381,952 | 1,061,303 | 13% |



(a) The raw point clouds (b) The extracted ground points

4.1 Non-ground removal results

### 4.1.2 Road Edge Detection and Refinement

In this study, road surfaces were extracted by detecting curb points as road boundaries. Three important parameters were pre-defined according to priori knowledge about road design standards, namely, minimum height difference $G_{min}$ = 7 cm, maximum height difference $G_{max}$ = 30 cm, and slope $S_{slope}$ = 70°. Figure 4.2 shows curb detection results of Dataset 1, Dataset 2, and Dataset 3.

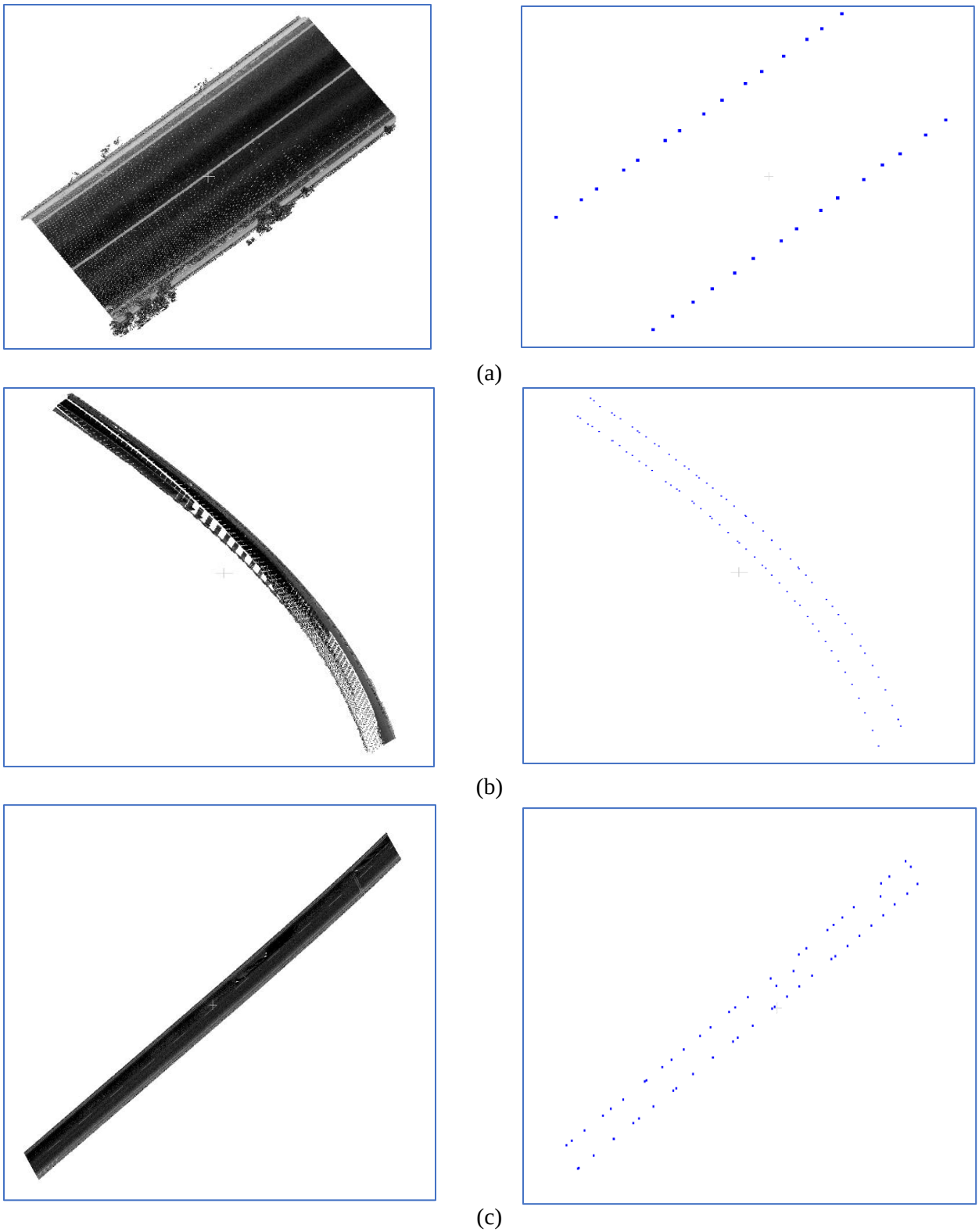
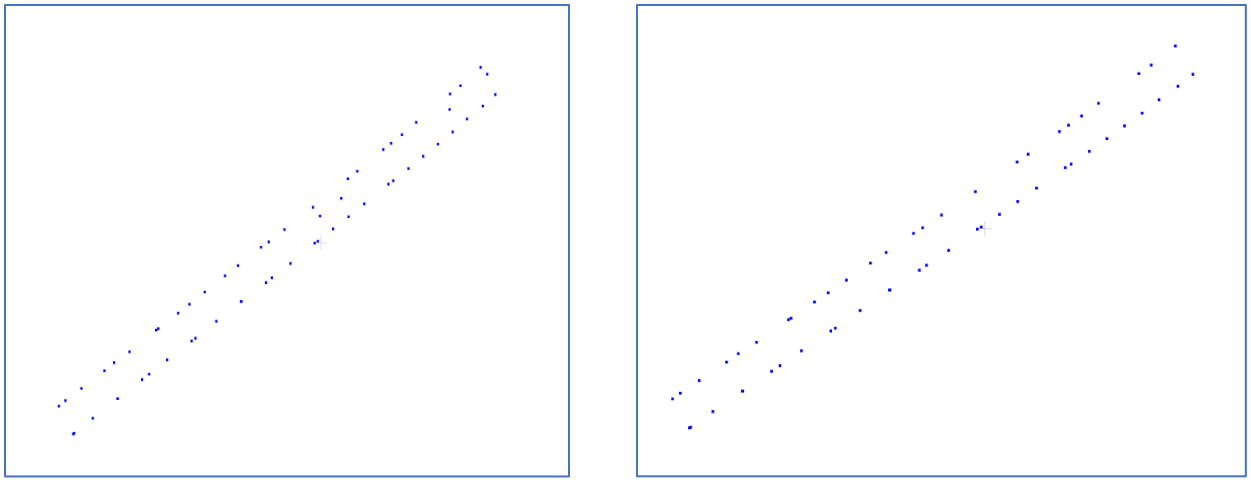


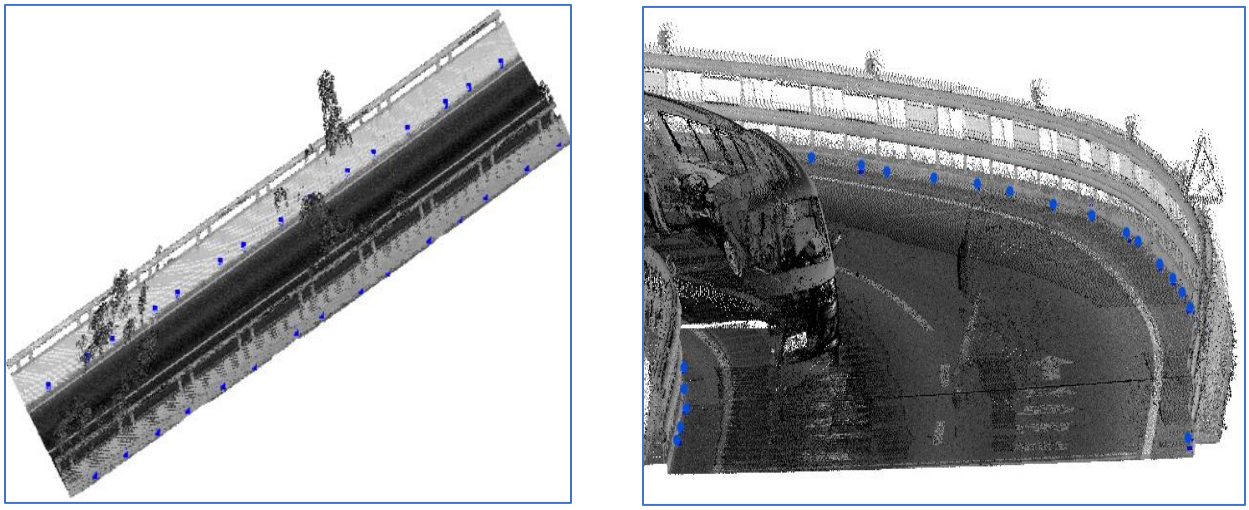
Figure 4.2 Curb detection results (a) Dataset 1; (b) Dataset 2; (c) Dataset 3: road intensity (left) and detected curb points (right)

As shown in Figure 4.2 (c), outliers inevitably exist in the curb detection results due to misclassification. Therefore, road edge refinement based on RANSAC is conducted to remove outliers and interpose points. The road edge refinement result of Dataset 3 is presented in Figure 4.3. Figure 4.4 shows the position of the extracted curb points in 3D scenes.



(a) (b)

Figure 4.3 Curb refinement results: (a) the extracted curb points; (b) the refined curb points



(a) (b)

Figure 4.4 Extracted curb points shown in blue from (a) Dataset 1; and (b) Dataset 5

### 4.1.3 Road Surface Extraction

Based on the position of the extracted road curb points, points between left and right road edges are regarded as road surface and extracted. It is noteworthy that a road edge is defined as the straight line that passes the two nearest curb points. As shown in Figure 4.5, the blue area is the extracted road surface in Dataset 1.

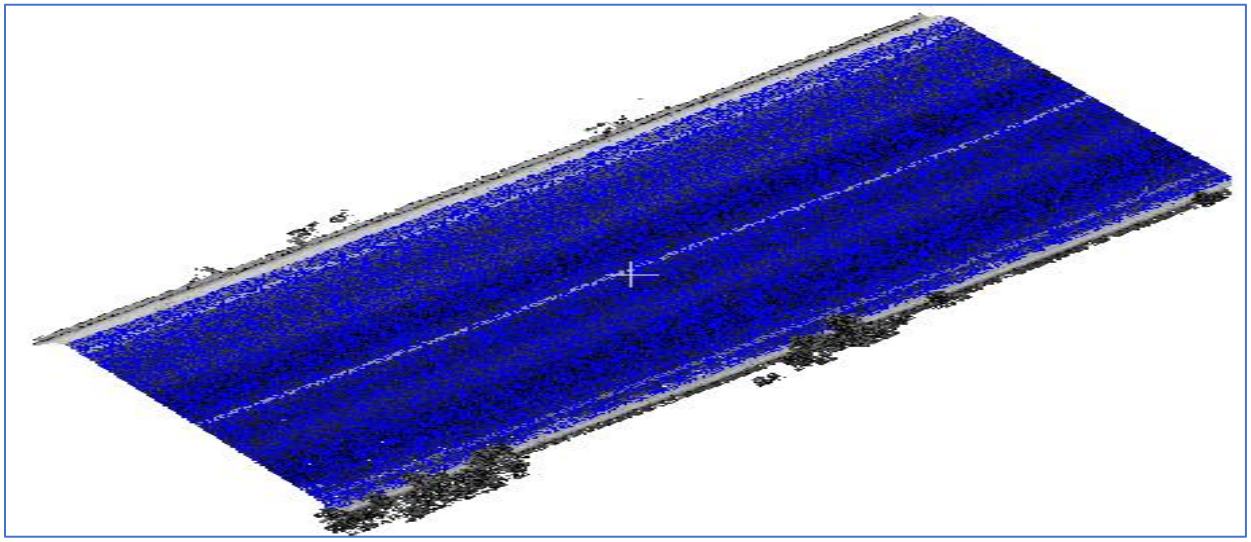
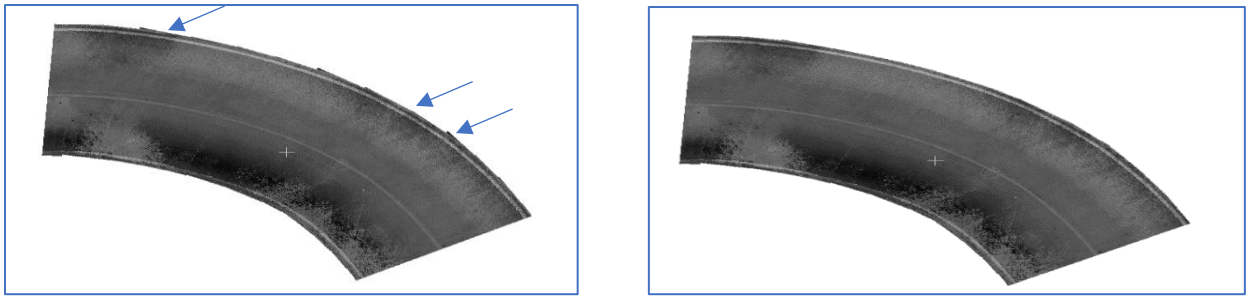


Figure 4.5 Road surface segmentation results

As mentioned in Section 3.6, partitioning data into smaller pieces can improve the performance of the proposed algorithm. This strategy is particularly useful for curve roads, as the algorithm extracts road surfaces in each local block with a pair of local road edges as reference. If the area of each block is not small enough, the estimated road edge could not fit the true road edge. As shown in Figure 4.6 (a), the more smoothed road edges can be obtained based on the 300-block segmentation (b) compared with that based on the 50-block segmentation (a). This indicates that the more blocks are used, the more accurate road edges can be extracted.



(a) (b)

Figure 4.6 Curved road surface sections segmented based on (a) 50 blocks and (b) 300 blocks.

### 4.1.4 Directional and Textual Road Marking Extraction

The proposed road marking extraction algorithm was tested on the five datasets, which contain different types of road markings including arrow, symbol, word, lane, and hatch markings. To achieve the best performance of the proposed method, local optimal thresholds were estimated based on the characteristics of different datasets. Generally, the minimum intensity threshold $I_{min}$ was set as 28000, and the maximum intensity threshold $I_{max}$ was set at 40000. Figure 4.7 presents the extracted road markings of a Dataset 5 section.

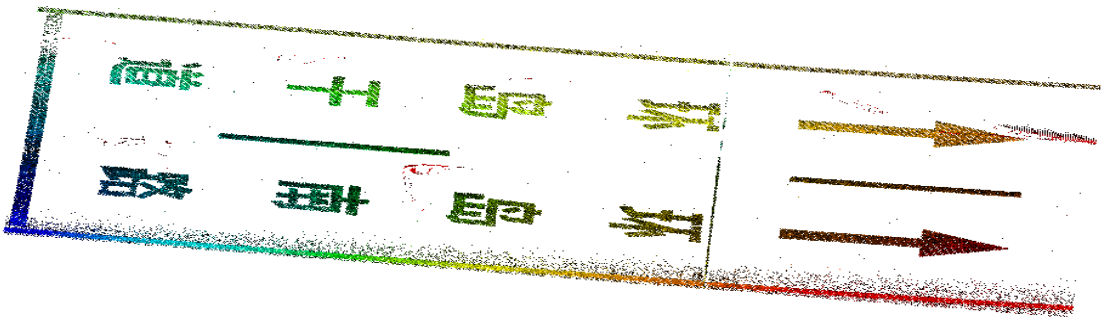


Figure 4.7 Road marking extraction results

An outlier removal function based on statistical analysis was performed to remove noise from road marking extraction results. The number of the nearest neighbours k should be set at

different values for different datasets according to their point density. Specifically, k was set at 5, 7, 7, and 8 for Datasets 1, 3, 4, and 5, respectively.

After outlier removal, the remained points still have no topological relationships. Thus, to organize discrete points into semantic groups, conditional Euclidean clustering is conducted. With a priori knowledge about road markings, the clusters of lane markings were removed from the results based on width analysis. Finally, only arrows, symbols, and words were remained in the 3D scene. As the targeted directional and textual markings are always wider than 15 cm, $w_c$ was set to 17 cm, and $d_c$ = 0.1 m, where $w_c$ is a pre-defined width threshold, and $d_c$ is a Euclidean distance threshold. Figure 4.8 shows the road markings of Dataset 5 after outlier removal. Figure 4.9 presents the extracted directional and textual markings.

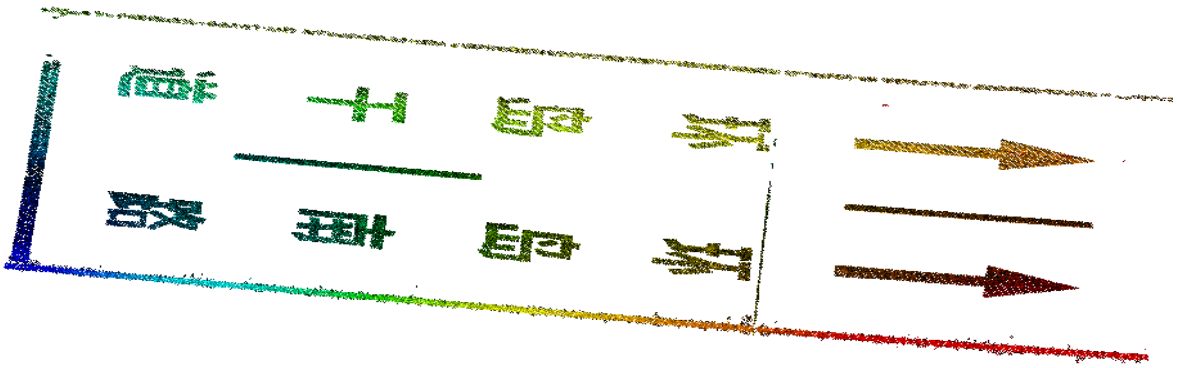


Figure 4.8 Noise-removed road markings of Dataset 5

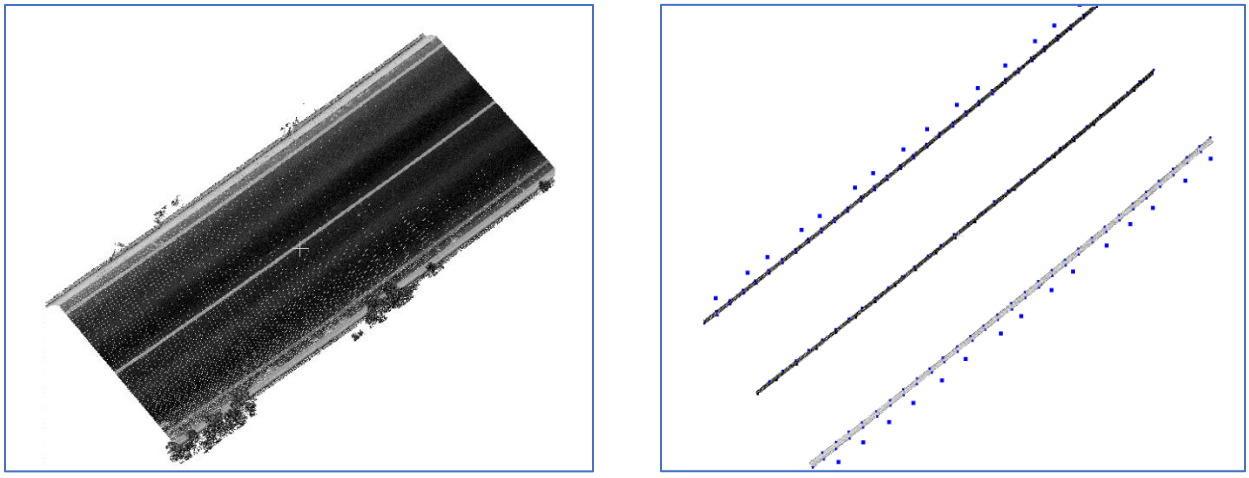
Figure 4.9 Extracted arrows and words of Dataset 5

### 4.1.5 Lane Marking Extraction

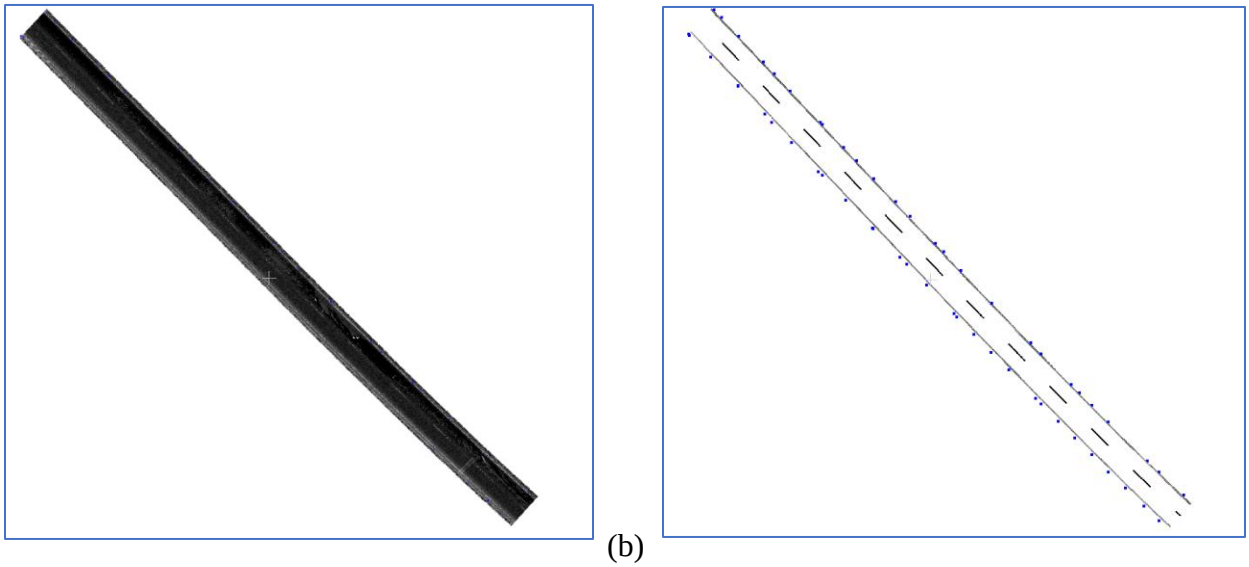
Lane markings are extracted using a multi-thresholding method, which detects lane markings both relying on intensity information and the distance to road edges. As mentioned in Section 3.6, the distance thresholds are determined by road design standards. According to *the Code for design of urban road* and *the Code for Layout of Urban Road Traffic Signs and Markings*, $d_{min}$ was set to 0.5 m, and $d_{max}$ was set to 0.9 m for roads without shoulders. For roads equipped with shoulders, $d_{min}$ = 1.5 m, $d_{max}$ = 2.75 m for the right side, and $d_{min}$ = 0.5 m, $d_{max}$ = 0.9 m for the left side. Apart from distance thresholding, intensity information is also used to detect lane marking points. Specifically, $I_{min}$ was set at 23000, and $I_{max}$ = 40000. The results obtained by the lane marking extraction algorithm is influenced by the following parameters:

- $d_{min}$: a pre-defined minimum distance threshold,
- $d_{max}$: a pre-defined maximum distance threshold,
- $I_{min}$: a pre-defined minimum intensity threshold, and
- $I_{max}$: a pre-defined maximum intensity threshold.

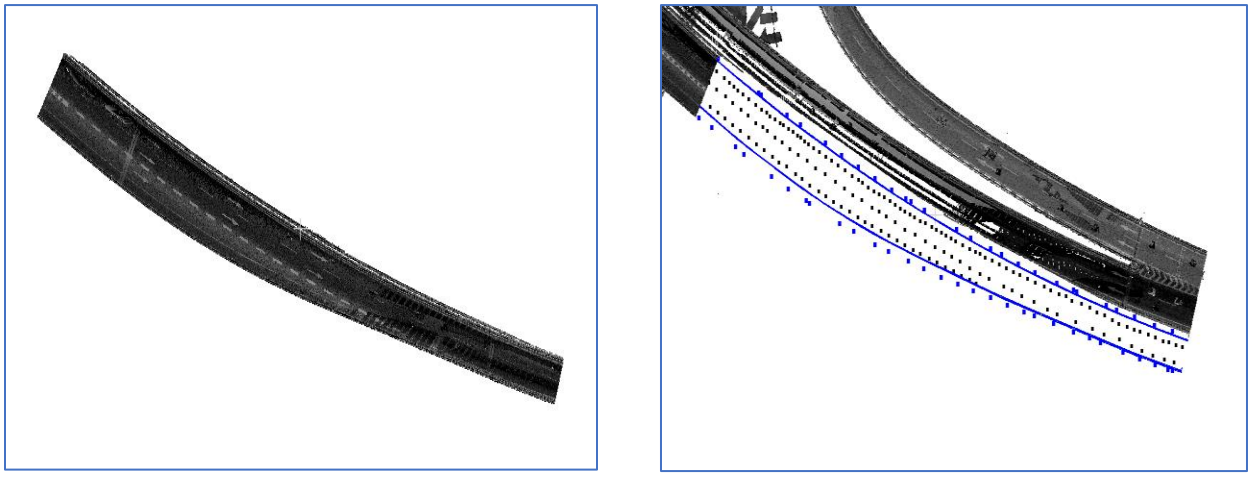
As mentioned in Section 3.1, approximately 50% of road surface data is missing in Dataset 2. Dataset 2 is not suitable for testing the proposed lane marking extraction algorithm. Figures 4.10 (a) to (d) show the lane marking extraction results and the extracted key points of Datasets 1, 3, 4, and 5. Figure 4.11 presents the detailed sections of the extracted lane markings and key points (curb points, lane marking edge points) of Datasets 1 and 3.



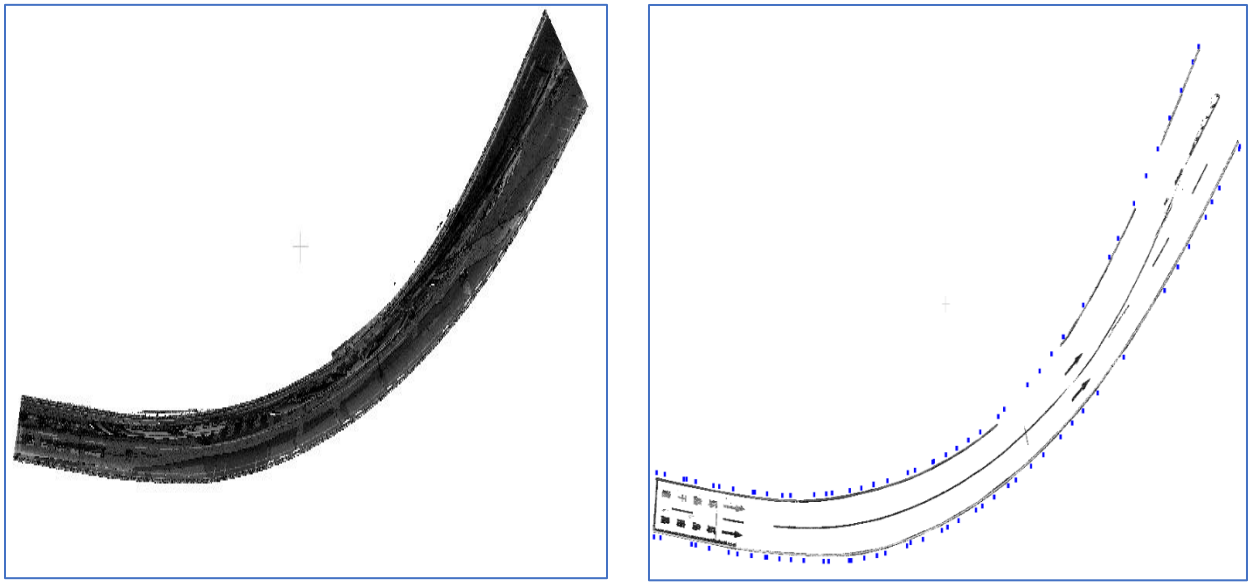
(a)



(b)

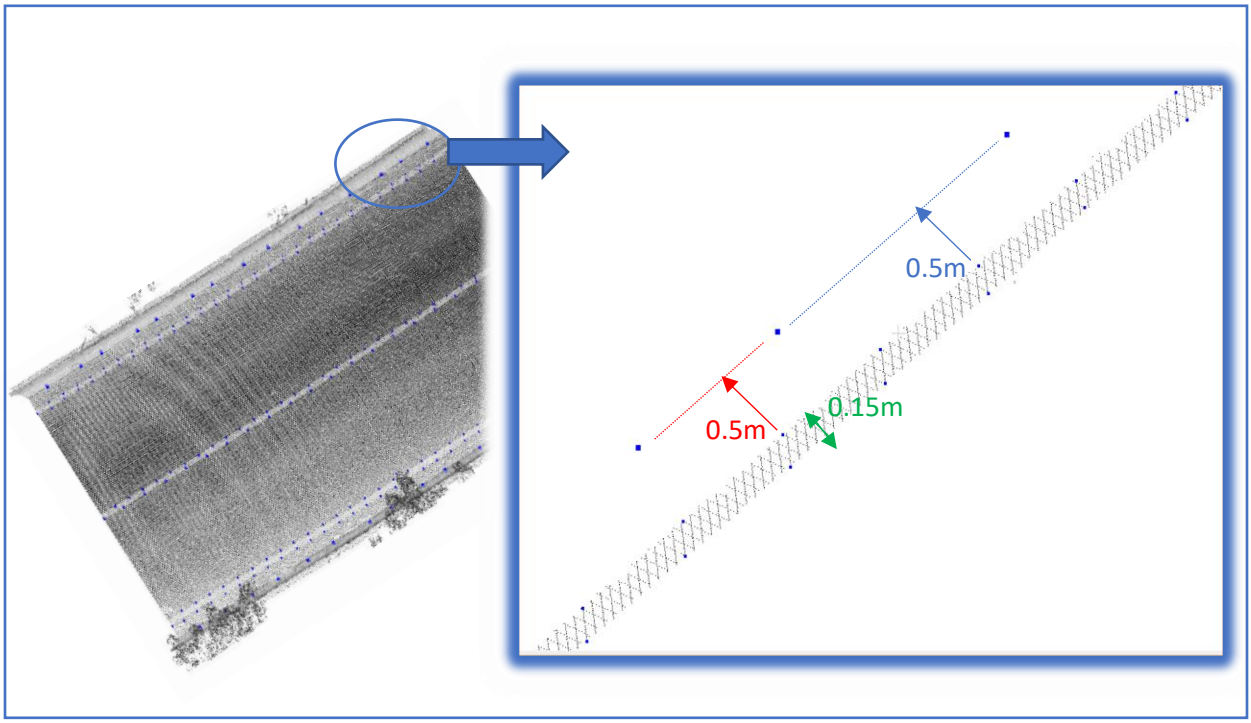


(c)

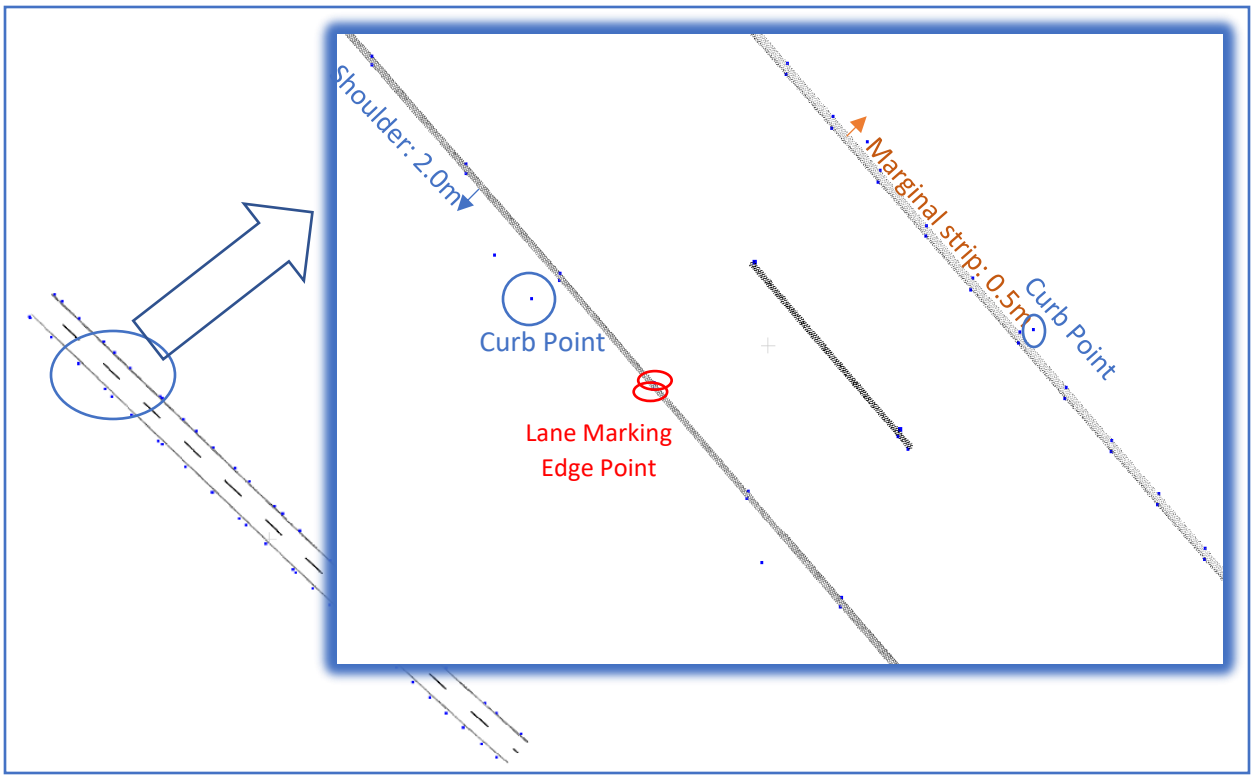


(d)

Figure 4.10 Extracted lane markings and key points (a) Dataset 1; (b) Dataset 3; (c) Dataset 4; (d) Dataset 5: road intensity (left) and key points (right)




(a)




(b)

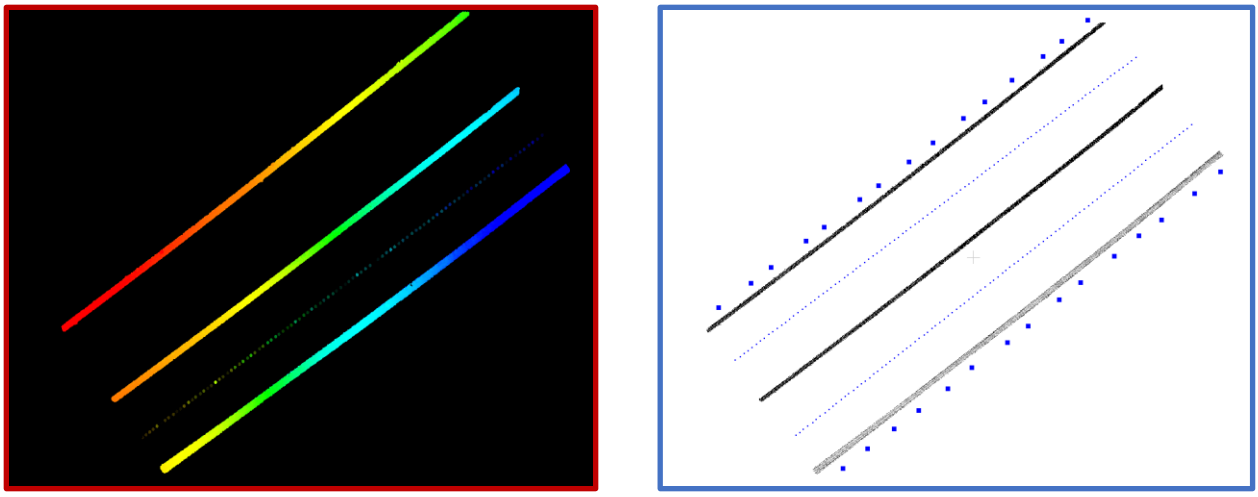
Figure 4.11 Detailed sections of the extracted lane markings and key points

(a) A section of Dataset 1; (b) a section of Dataset 3

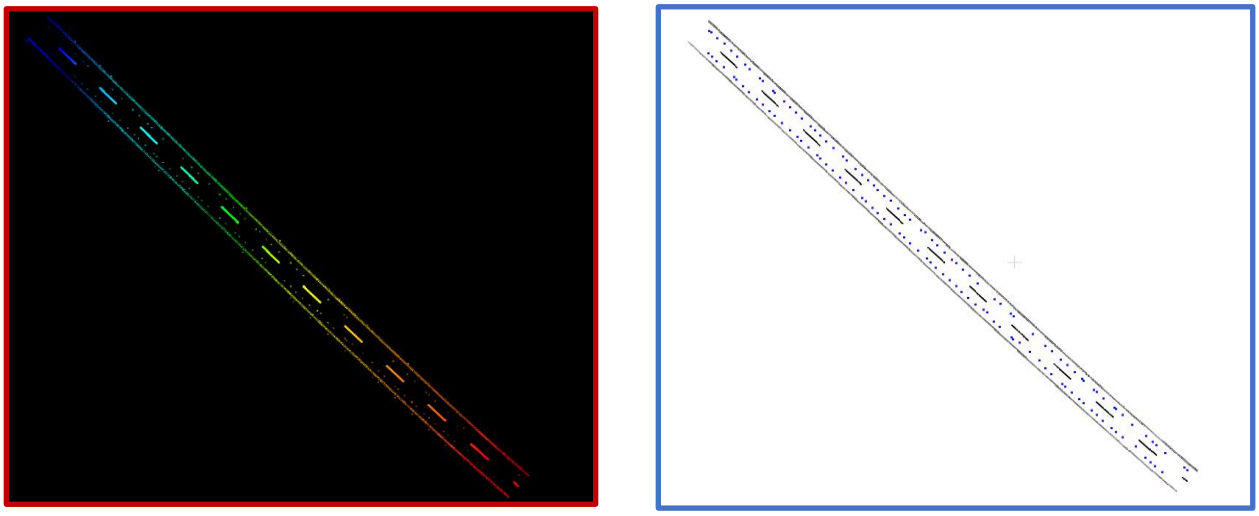
As shown in Figures 4.10 and 4.11, most lane markings were successfully extracted from raw MLS data. However, some of the road markings in Figure 4.10 (d) are incomplete. The missing parts were primarily caused by the occlusions of large obstacles, as laser beams cannot reach the shadows of the obstacles. Another probable reason is that some road markings have been painted on the road surfaces for too many years. Thus, some parts of the road markings have been worn by vehicles and pedestrians. In summary, the proposed lane marking extraction method produced promising results.

### 4.1.6 Lane Centerline Estimation

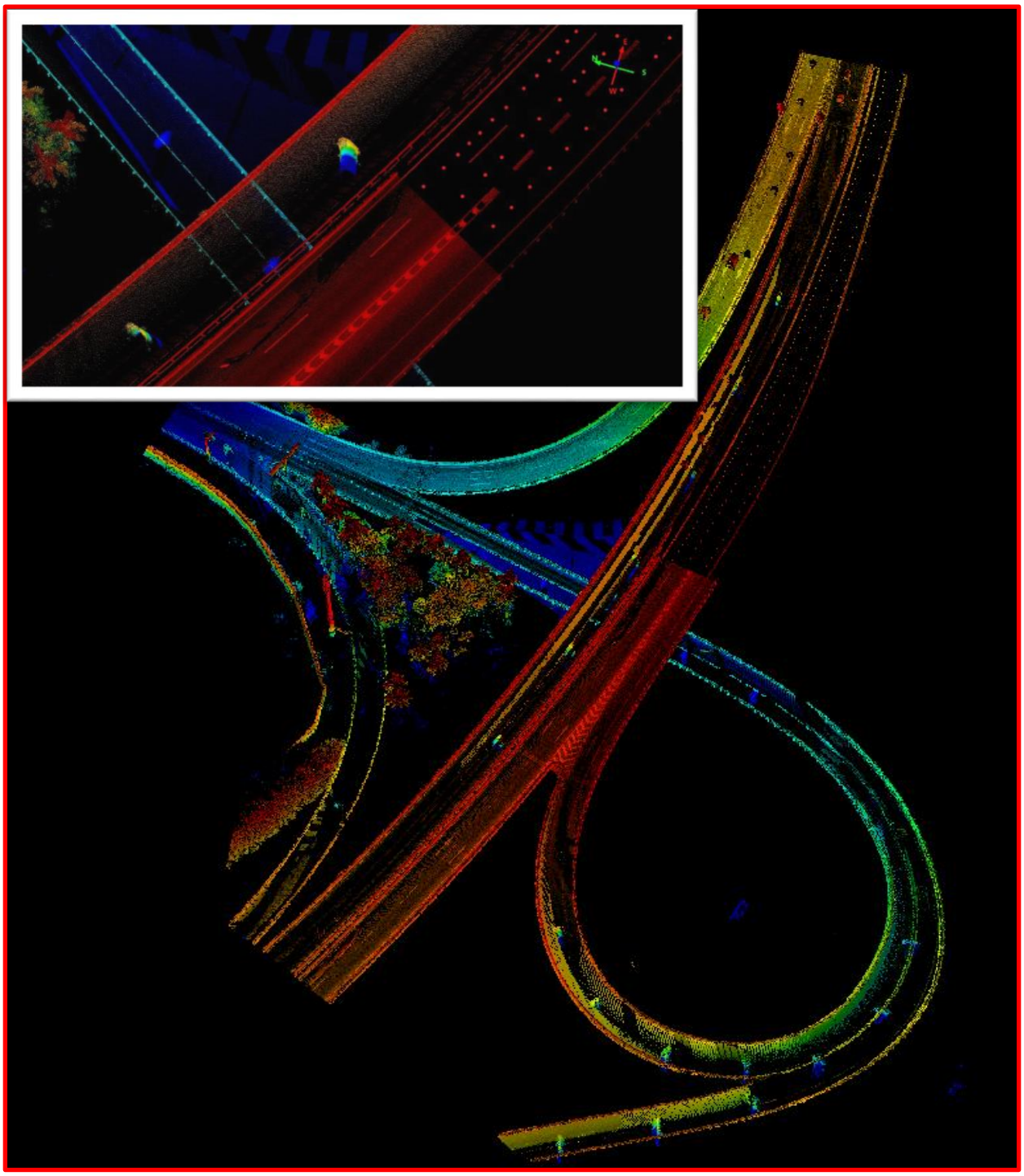
The coordinates of lane centerline points were estimated according to the coordinates of lane marking points on both left and right sides. In this study, $W_z$ was set to 2.0 m for Datasets 1, 3, 4, and $W_z$ = 0.5 m for Dataset 5. $W_z$ is a pre-defined width threshold that denotes the width of each block. For example, with a 0.5 m local block width, Dataset 5 was partitioned into over 350 blocks to estimate lane centerline points. Figures 4.12 (a) to (d) present the estimated lane centerline points of Datasets 1, 3, 4, and 5, respectively.



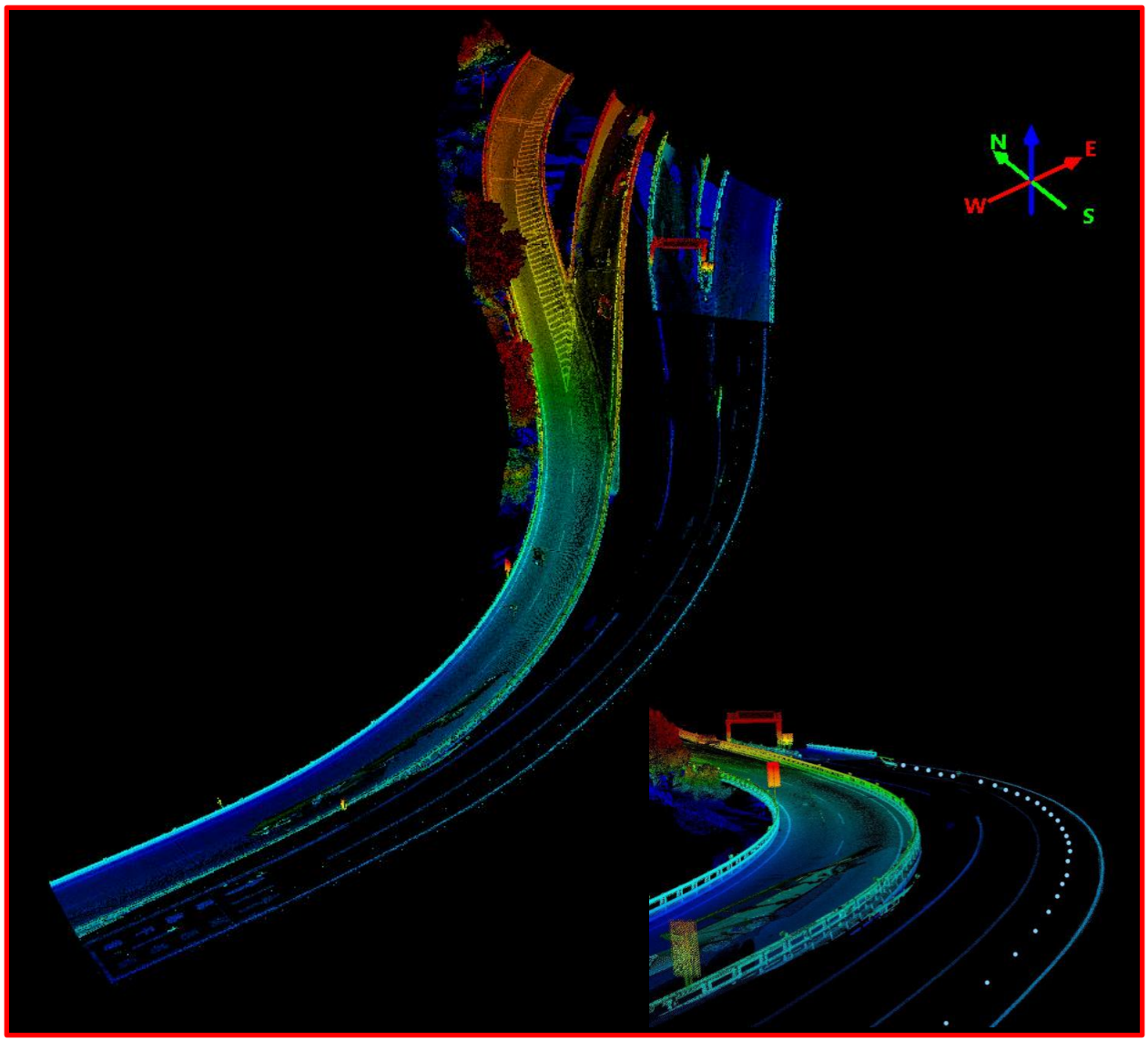
(a)



(b)



(c)




(d)

Figure 4.12 The estimated lane centerlines of (a) Dataset 1; (b) Dataset 3; (c) Dataset 4; (d) Dataset 5

### 4.1.7 3D High-definition Road Map Prototype

With the extracted road markings, road edges, and the estimated lane centerline points, a prototype of 3D high-definition road map is created. The generated road map is a robust tool for autonomous navigation. By matching or associating map features and features detected by interior

sensors and calculating the offset, the navigation system can promptly make decisions in terms of steering, brake, throttle, lane changing, etc. If the offset is lower than a given threshold, meaning the 3D scene captured by sensors matches the 3D environment stored in the priori map, the designated lane centerline would be safe for the autonomous vehicle to drive along. Figure 4.13 presents the road map of the whole viaduct area.

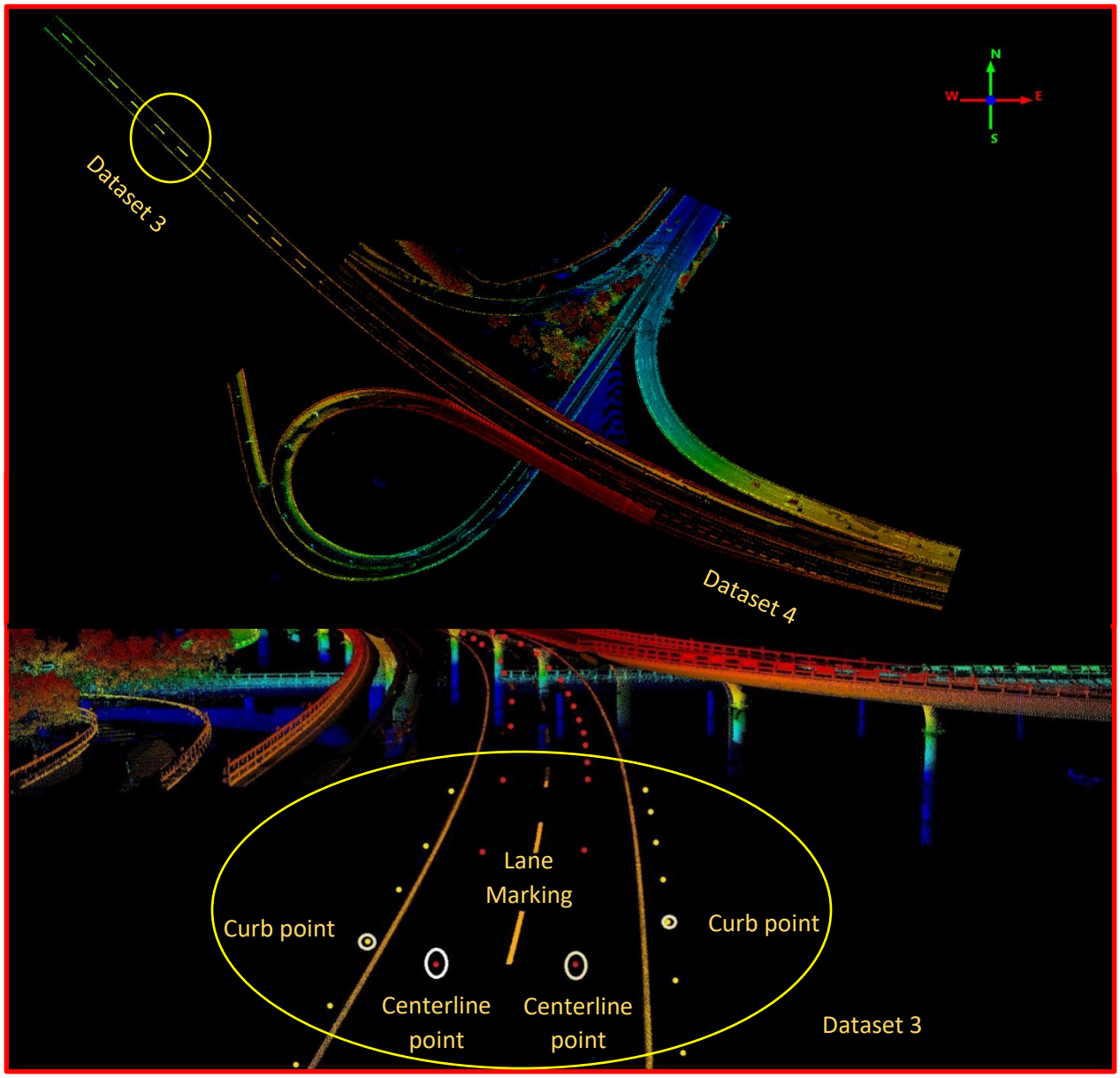



Figure 4.13 3D high-definition map prototype of Dataset 3-4

After extracting all useful road information, the generated LAS file was converted to SHP file and further processed in ArcGIS. Separate lane centerline points were linked using a XY-Line tool in ArcGIS to create smooth trajectories. In addition, the map can be converted to other types of user-defined format. Finally, the map was ready for the navigation task of autonomous vehicles. The linked lane centerlines for Datasets 3 and 4 are shown in Figure 4.14.

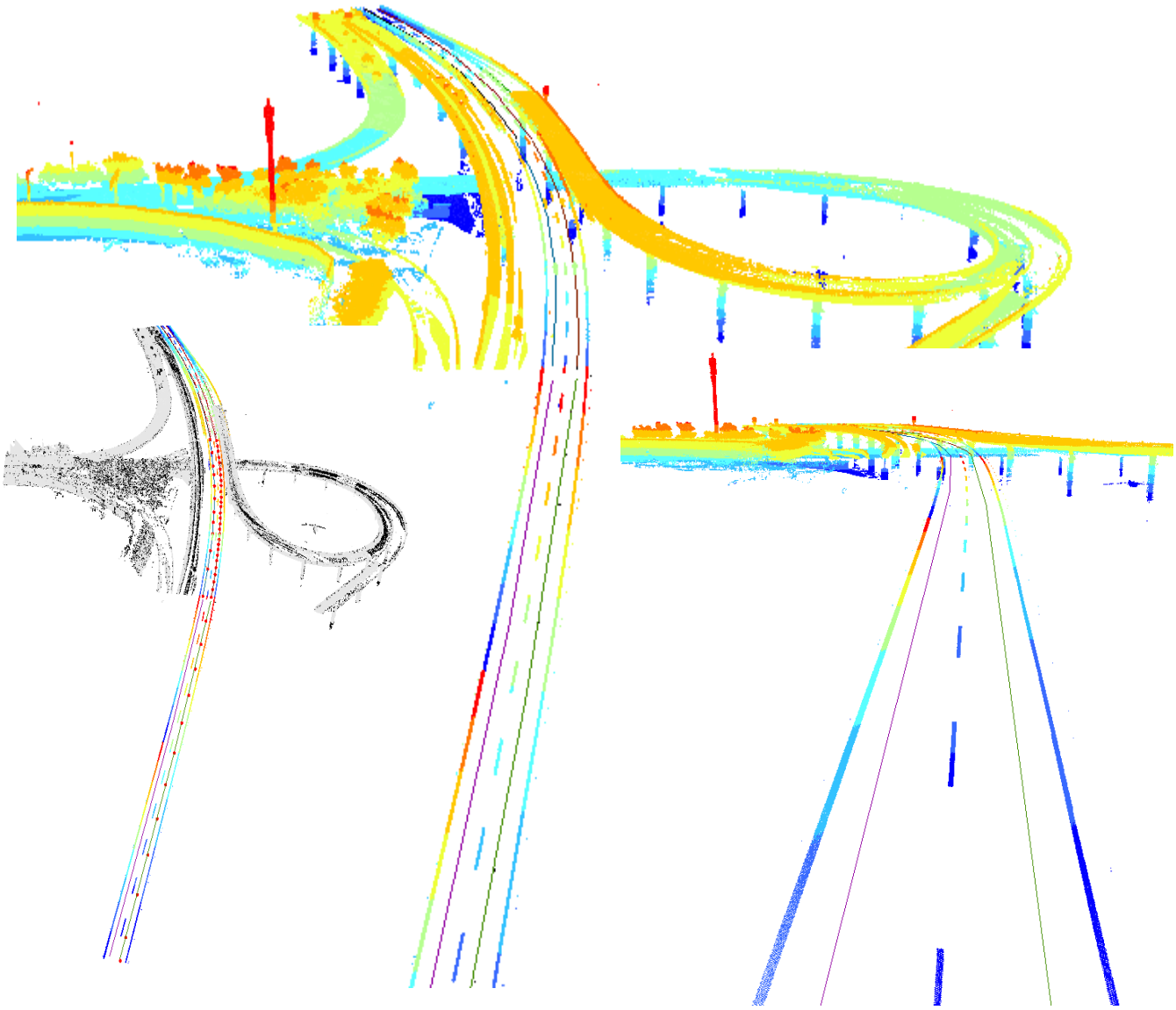


Figure 4.14 Generated trajectories in ArcGIS

## 4.2 Accuracy Assessment

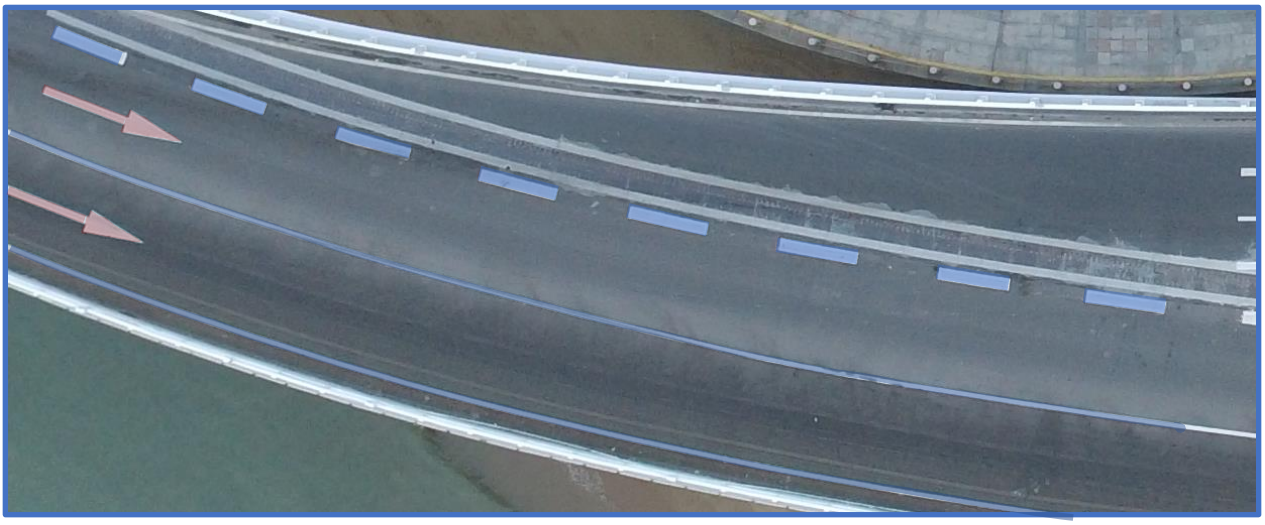
### 4.2.1 Road Marking Extraction

As mentioned in Chapter 3, the performance of road marking extraction was evaluated by manually labeled reference pixels in UAV orthoimages. The manually interpreted orthoimage and the overlapped orthoimage are set in the same resolution. Figure 4.15 shows manually interpreted road marking polygons in UAV orthoimages and the orthoimages overlapped by the extracted road markings. Figures 4.15 (a) and (b) present the validation results of Dataset 4. Figures 4.15 (c) and (d) present the validation results of Dataset 5. The precision, recall, and $F_1$-score of the five datasets are listed in Table 4.2.

(a)

(b)

(c)



(d)

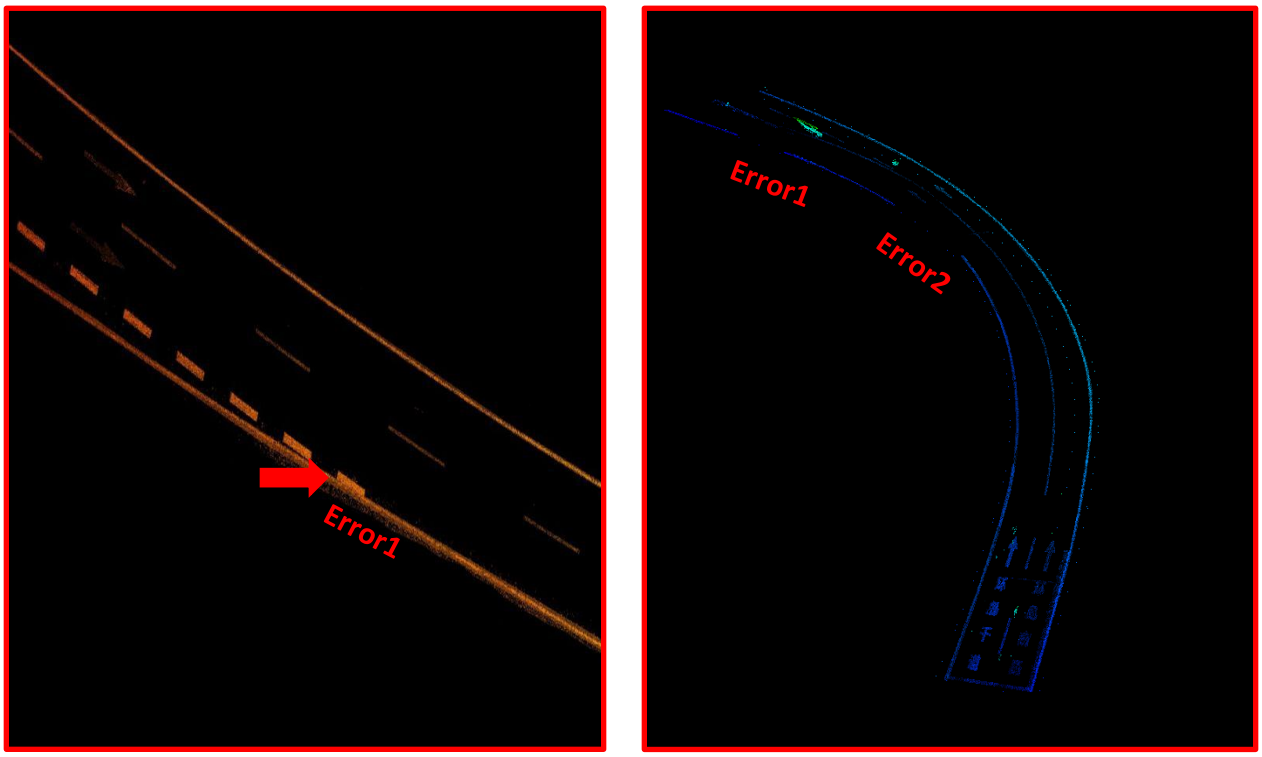
Figure 4.15 Validation results (a), (c) manual interpreted polygons in the orthoimage of Dataset 4, Dataset 5; (b), (d) orthoimage overlapped by extracted road marking pixels of Dataset 4, Dataset 5;

Table 4.2 Quantitative assessment of road marking extraction

| Dataset | Precision (%) | Recall (%) | $F_1$-score |
|---|---|---|---|
| 1 | 97.78 | 96.57 | 0.9717 |
| 2* | N/A | N/A | N/A |
| 3 | 95.65 | 95.54 | 0.9559 |
| 4 | 96.83 | 89.72 | 0.9314 |
| 5 | 85.21 | 93.22 | 0.8904 |

*Note: As most road surface points are missing, Dataset 2 is not suitable for testing road marking extraction algorithms.

As shown in Table 4.2, the proposed road marking extraction method can achieve 97.78 % in precision, 96.57 in % recall, and 0.9717 in $F_1$-Score. Both precision and recall of Dataset 1 are very high, as Dataset 1 is a short and straight road with no obstacles. Similarly, Dataset 3 is a straight section of Yanwu viaduct, but much longer than Dataset 1. The recall of Dataset 4 is relatively lower, as a lane reduction area exists in Dataset 4. The number of lanes is reduced from three lanes to two lanes in same direction. As the effective width of the road is reduced, it is more difficult to estimate local optimal thresholds for lane reduction areas. As a result, the recall for Dataset 4 is lower due to some misclassification. As to Dataset 5, two moving vehicles block some road markings on the left side of the road, thus contributing to a low precision. The occlusion of shadows and obstacles is a common shortcoming of MLS system, as laser beams cannot penetrate most objects. Figure 4.16 shows some misclassification and incomplete extraction in Datasets 4 and 5.



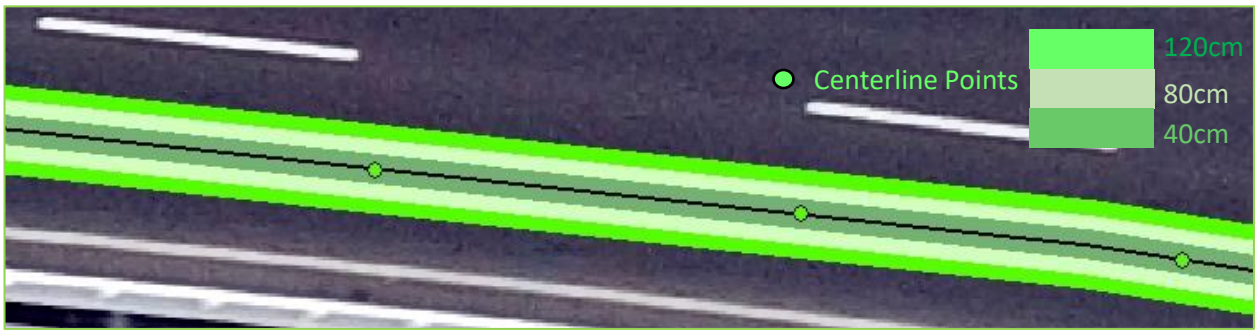

(a) (b)

Figure 4.16 Errors in road marking extraction (a) Dataset 4; (b) Dataset 5

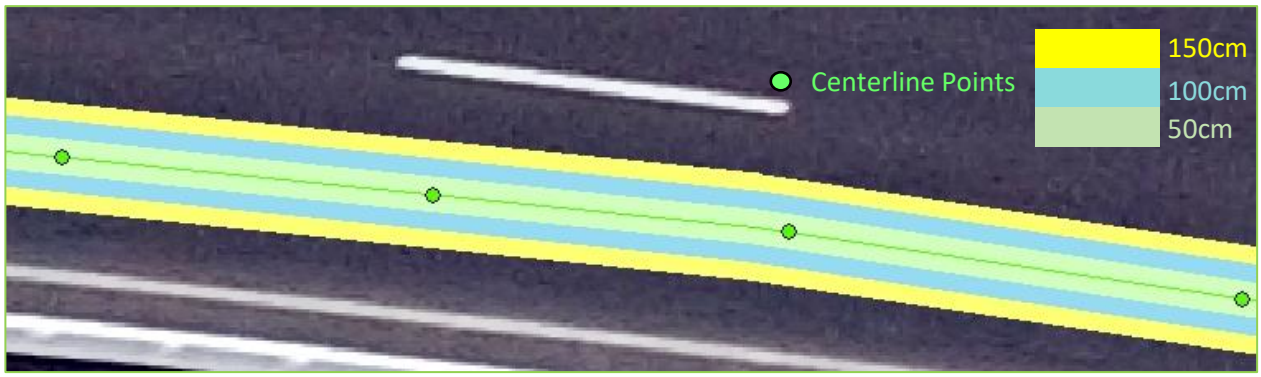
### 4.2.2 Lane Centerline Estimation

As described in Section 3.7.2, UAV orthoimages were used to validate the simulated lane centerlines. The orthoimages were collected by an UAV on May 5th, 2017. Reference lane centerline pixels were manually interpreted in these UAV orthoimages using ArcGIS v10.2.2. Afterwards, buffer zones with different widths were created and overlapped with the simulated lane centerlines. Figures 4.17 (a) and (b) show the lane centerline validation results of Datasets 4 and 5, respectively. Figures 4.17 (c) and (d) present buffers overlapped by the simulated lane centerlines in detailed road sections.

(a) (b)




(c)




(d)

Figure 4.17 Lane centerline validation results (a) Dataset 4; (b) Dataset 5; (c) (d) Detailed sections of the extracted lane centerlines

All of the simulated lane centerlines were evaluated through manual inspection. The estimated lane centerlines of Datasets 1, 3, and 4 can perfectly match the real centerlines. All of these lane centerlines were located in 10-cm-wide buffer zones. The simulated lane centerline of Dataset 5 had a small deviation due to some misclassfication and incomplete extraction of road marking. The maximal distance error is approximately 15 cm. Thus, a 30-cm-wide buffer zone was drawn to cover the whole lane centerline of Dataset 5.

In practice, the common width of cars is usually less than 2 m, and the width of a single traffic lane in China should be no less than 3.5 m. Thus, a 40-cm lateral clearance can ensure driving safety. According to the 2004 *Enhanced digital mapping project final report* by United States Department of Transportation, the accuracy requirement of lane centerline estimation is 0.3 m. Therefore, all of the above simulated lane centerlines meet this safety requirement.

### 4.3 Computing Complexity Analysis

The proposed 3D high-definition map creation method was implemented using C++. The computer used in this thesis is a desktop with an 8 GB RAM and an AMD FX-6300(TM) CPU. The computing time of each step shows the high efficiency of the proposed algorithms. Taking Dataset 3 as an example, the proposed algorithm took 76.9 s for pre-processing, 262.3 s for road marking extraction, 121.5s for lane marking extraction and lane centerlines estimation. Table 4.3 shows the computing time of each procedure in the proposed method.

Table 4.3 Efficiency of the proposed method

| Dataset | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Size | 64 MB | 23MB | 226 MB | 303 MB | 281MB |
| Number of Points | 1,920,753 | 696,891 | 6,758,030 | 9,058,578 | 8,381,952 |
| Coordinates Transformation | 1.0s | 0.8s | 9.0s | 9.8s | 9.7s |
| Ground Removal | 1.4s | 1.0s | 13.1s | 15.7s | 16.0s |
| Curb Detection | 9.5s | 9.0s | 30.3s | 40.3s | 51.7s |
| Road Edge Refinement | 1.5s | 1.2s | 2.2s | 2.5s | 2.7s |
| Road Surface Segmentation | 8.0s | 4.7s | 22.3s | 25.7s | 40.3s |
| Intensity Analysis | 7.2s | N/A | 15.7s | 17.6s | 16.7s |
| Nosie Removal | 59.2s | N/A | 201.1s | 255.7s | 230.0s |
| Clustering | 39.7s | N/A | 45.5s | 60.5s | 56.7s |
| Lane Marking Extraction | 10.7s | N/A | 99.8s | 144.8s | 179.3s |
| Lane Centerline Estimation | 10.0s | N/A | 21.7s | 28.5s | 26.6s |
| Total Time | 158.2s | N/A | 460.7s | 601.1s | 629.7s |

It is noted that Dataset 5 requires more computing time than other datasets in the same procedures. The primary reason of this phenomenon is the high data complexity of Dataset 5, as it involves an expressway entrance and a horizontal curve. To improve the proposed method performance, Dataset 5 was segmented into a considerable number of pieces, thus costing more computing time.

In addition, the volume of data used in this study is generally larger than that of many previous studies, as they usually manually selected a small region to test their algorithms. The results show the capacity of the proposed method to efficiently process big data.

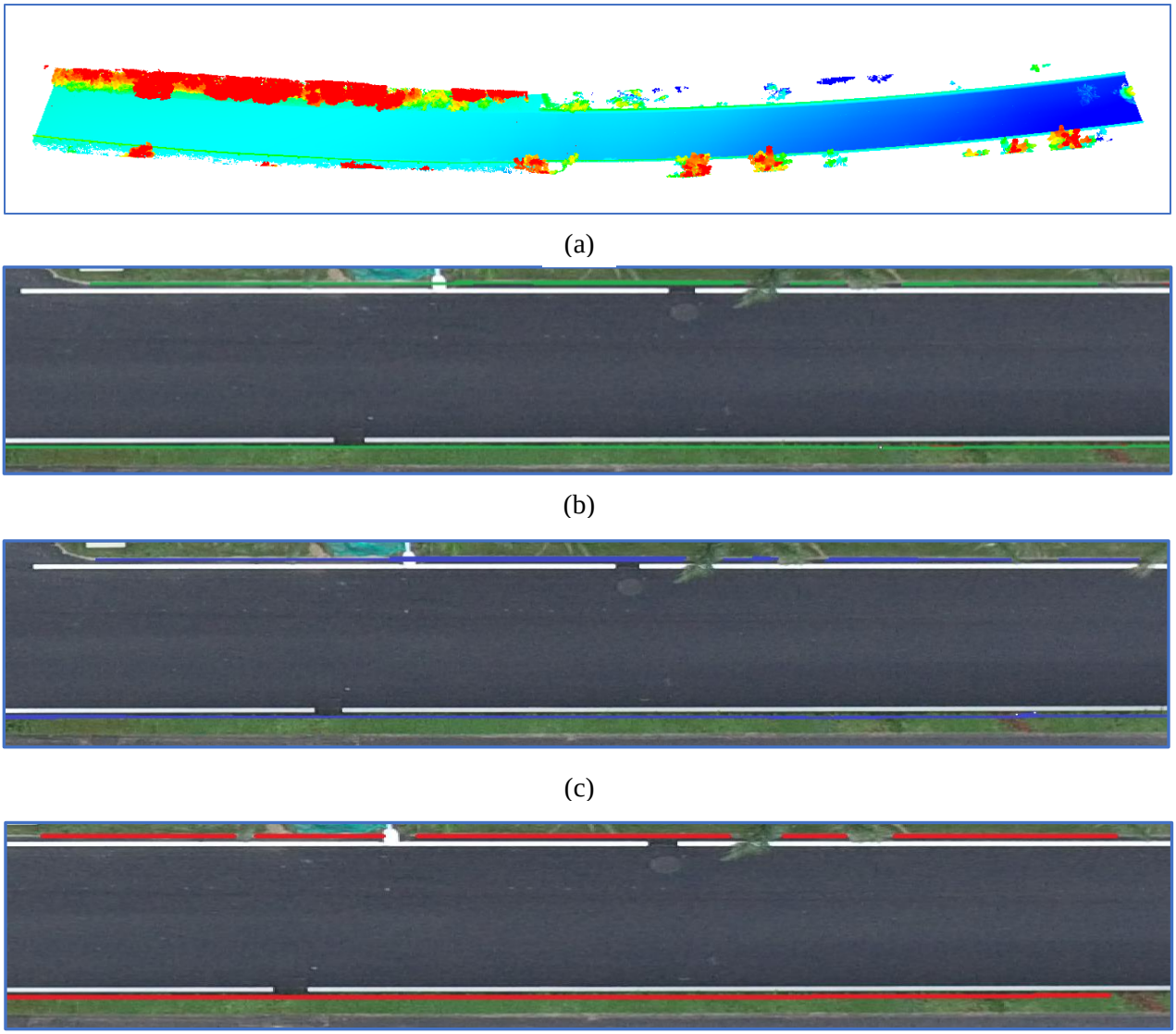
## 4.4 Comparative Study

To objectively assess the performance of the proposed method, this thesis also undertook two comparative studies. Section 4.4.1 describes the comparison between the proposed road edge detection method and Yu's method (Yu et al., 2015). Section 4.4.2 shows the comparison between the proposed road marking extraction method and several existing road marking extraction methods.

### 4.4.1 Comparative Study of Road Edge Detection

The proposed road edge detection method is an improved version of Yu et al. (2015). Thus, the proposed method has some similarities to Yu et al. (2015). For example, both Yu et al. (2015) and the proposed method are segmentation-based methods, which partition the road into profiles and detect curb points in local blocks. However, Yu et al. (2015) partitions the road into a fixed number of pieces according to the position of trajectory points, while the proposed method can partition the road into any number of segments. Furthermore, Yu et al. (2015) detects curb points by generating pseudo scan lines and selecting the highest point within the lowest layer, while the proposed method directly detects the lowest curb candidates in every local block, and a road edge refinement algorithm is applied after road edge detection. To evaluate the performance of these two methods, the extracted road edges were overlapped with UAV orthoimages and checked against manually labeled reference pixels. Both MLS point clouds and UAV orthoimages were in

the same coordinate system (UTM). Figures 4.18 and 4.19 present the evaluation results of these two methods.



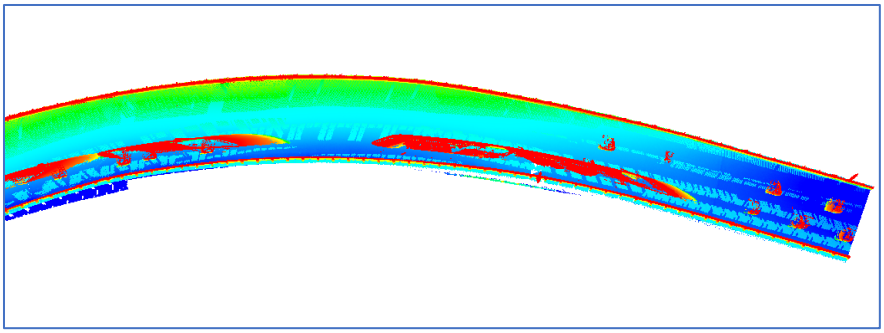
(a)

(b)

(c)

(d)

Figure 4.18 Extracted road edges from a Dataset 1 section: (a) raw point clouds; (b) Yu's result; (c) the proposed method's result; (d) manually labeled reference data



(a)

(b)

(c)

(d)

Figure 4.19 Extracted road edges from a Dataset 5 section: (a) raw point clouds; (b) Yu's result; (c) the proposed method's result; (d) manually labeled reference data

The performance of Yu's method and the proposed method was quantitatively evaluated using precision, recall, and $F_1$-score. Table 4.4 presents the quantitative evaluation results of these two methods.

Table 4.4 Accuracy assessment results of different road edge detection methods

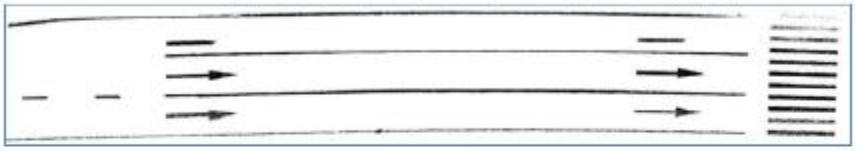
| Sample | Dataset 1 | | Dataset 5 | |
|---|---|---|---|---|
| Method | Yu et al. (2015) | The proposed | Yu et al. (2015) | The proposed |
| Precision (%) | 96.54 | 96.32 | 85.21 | 95.06 |
| Recall (%) | 95.20 | 95.61 | 83.43 | 92.91 |
| $F_1$-score | 0.9586 | 0.9596 | 0.8431 | 0.9397 |

Interestingly, Yu's method and the proposed method achieved similar precision, recall, and $F_1$-score for Dataset 1, but for Dataset 5, the proposed method is obviously superior to Yu's method. The comparison results show that Yu's method is not capable of handling scenes involving horizontal curves and obstacles.

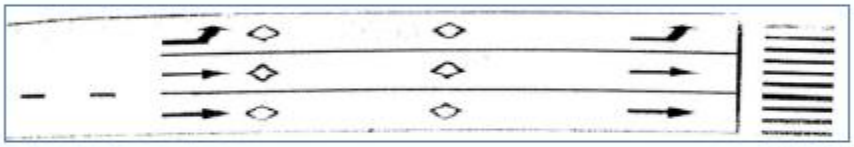
### 4.4.2 Comparative Study of Road Marking Extraction

The proposed road marking extraction method was compared with some existing methods, including Chen et al. (2009), Guan et al. (2014), Yu et al. (2015), and Cheng et al. (2017). In Guan et al. (2014) and Cheng et al. (2017), MLS point clouds were converted to 2D intensity images first. Then they extracted road markings from those 2D images with the help of software including ENVI and ArcGIS. On the other hand, Chen et al. (2009), Yu et al. (2015), and the proposed method directly extracted road markings from 3D MLS point clouds. In this section, the experimental results of these studies will be compared. Figures 4.20 and 4.21 present the evaluation results of these studies.

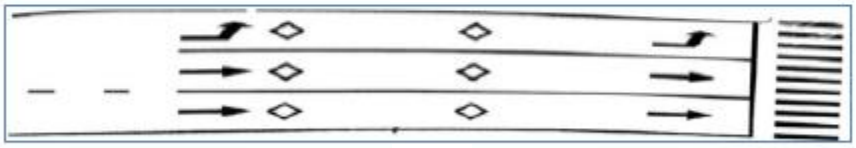
(a)



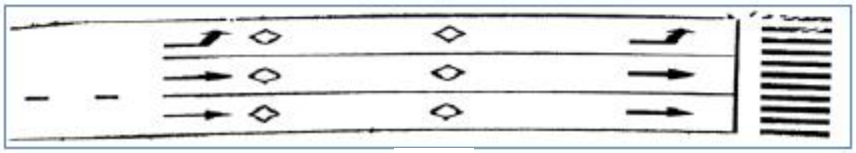
(b)



(c)

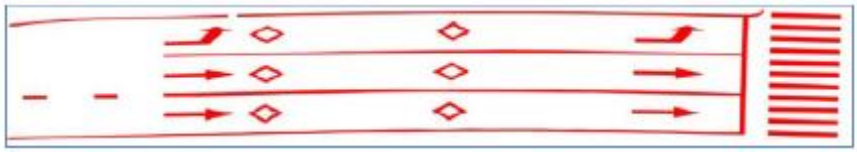


(d)



(e)

(f)



(g)

Figure 4.20 Extracted road markings from Sample 01: (a) road surface points; (b) Chen's result; (c) Guan's result; (d) Yu's result; (e) Cheng's result; (f) the proposed method's result; (f) manually labeled reference data (Yu et al., 2015)

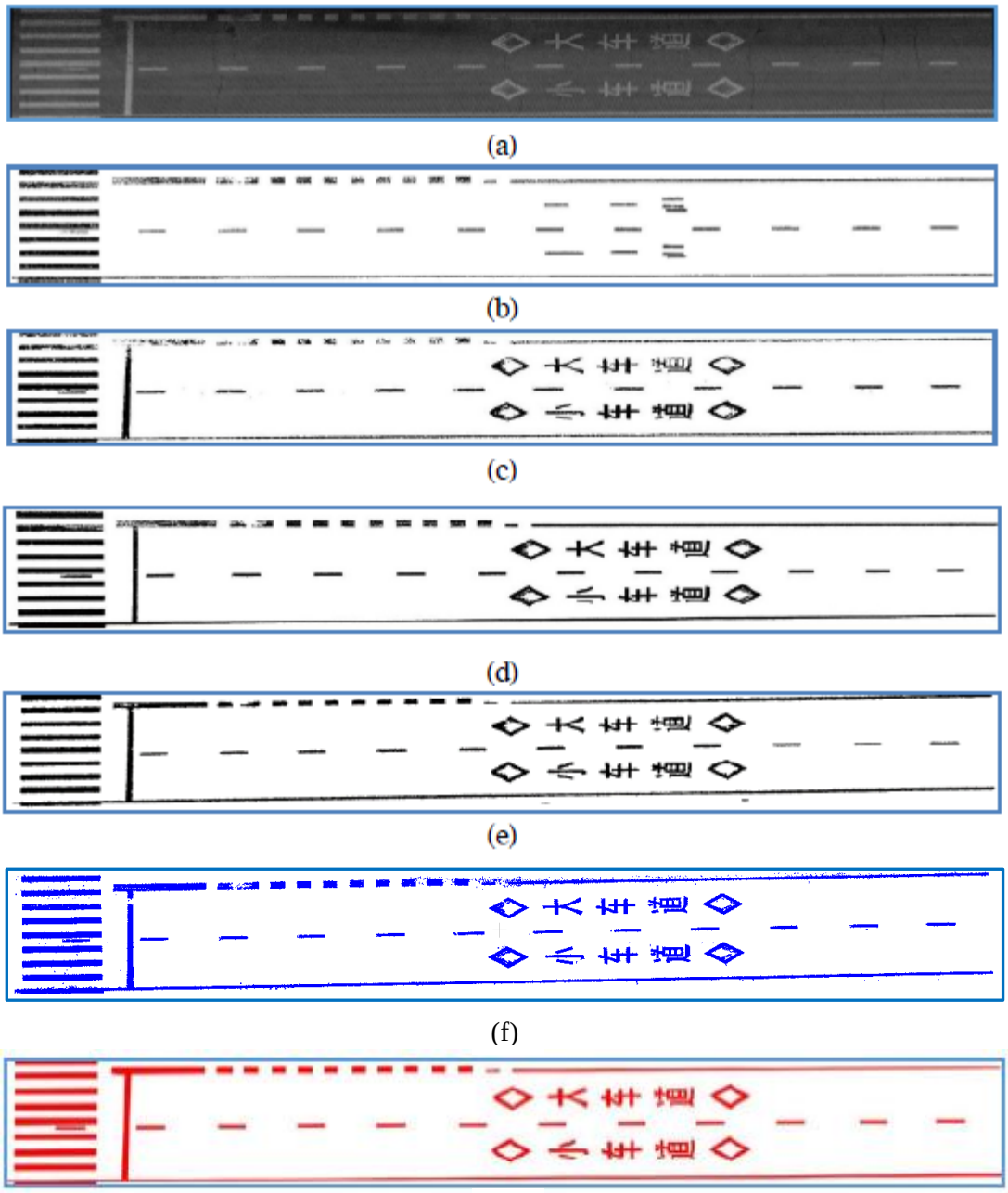


Figure 4.21 Extracted road markings from Sample 05: (a) road surface points; (b) Chen's result; (c) Guan's result; (d) Yu's result; (e) Cheng's result; (f) the proposed method's result; (f) manually labeled reference data (Yu et al., 2015)

Table 4.5 Accuracy assessment of different road marking extraction methods (Sample 01)

| Sample | Sample 01 | | | | |
|---|---|---|---|---|---|
| Method | Chen et al. (2009) | Guan et al. (2014) | Yu et al. (2015) | Cheng et al. (2017) | The proposed |
| Precision (%) | 75 | 86 | 93 | 89 | 95 |
| Recall (%) | 91 | 90 | 92 | 96 | 92 |
| $F_1$- score | 0.82 | 0.88 | 0.93 | 0.92 | 0.94 |

Table 4.6 Accuracy assessment of different road marking extraction methods (Sample 05)

| Sample | Sample 05 | | | | |
|---|---|---|---|---|---|
| Method | Chen et al. (2009) | Guan et al. (2014) | Yu et al. (2015) | Cheng et al (2017) | The proposed |
| Precision (%) | 71 | 89 | 93 | 91 | 95 |
| Recall (%) | 92 | 91 | 91 | 98 | 90 |
| $F_1$- score | 0.80 | 0.90 | 0.92 | 0.94 | 0.93 |

Tables 4.5 and 4.6 presents the quantitative evaluation results of the mentioned methods. It is noted that the proposed method achieves the highest precision among the five methods. However, our approach is inferior to Cheng et al. (2017) in terms of recall. As shown in Figure 4.21, some outliers still exist in the experimental results, especially around the right-hand lane markings. Consequently, the outlier removal algorithm and clustering algorithm still need improvement in this thesis.

The proposed approach outmatches Chen et al. (2009) and Yu et al. (2015) both in precision and recall. Chen et al. (2009) only focuses on lane markings, while Yu et al. (2015) and the proposed method can extract all types of road markings.

Compared with Guan et al. (2015) and Cheng et al. (2017), the proposed method never suffers from inconstant intensity and blurring data, thus achieving higher precision. As shown in Figures 4.20 and 4.21, road markings near the road boundary could not be successfully extracted using Guan's and Cheng's methods. Moreover, their methods greatly rely on third-party software and require extensive manual operation, which can be time-consuming.

## 4.5 Chapter Summary

This chapter presents the experimental results of the proposed method. The accuracy assessment demonstrates that the proposed method can successfully distinguish and extract road features with high accuracy. The estimated lane centerlines are suitable for autonomous navigation. However, the accuracy of the proposed methods is influenced by data complexity. The presence of sharp curves, shadows, and large obstacles could have negative effects on the proposed algorithms.

This chapter also describes the computing complexity of the proposed method. The computing time of every steps are listed and compared, which indicates the high efficiency of the proposed method. However, the computational time should be further reduced, especially for road marking extraction and noise removal.

Finally, this chapter introduces two comparative studies, which show the robustness and feasibility of the proposed method.

# Chapter 5

# Conclusions and Recommendations

This chapter presents the conclusions of this thesis, discusses its limitations, and puts forward some suggestions for future research.

## 5.1 Conclusions

The autonomous vehicle, also known as the driverless vehicle, is a vehicle that can perceive the environment and navigate itself without human input. Fully autonomous vehicles are expected to be launched in the next ten years. Among all autonomous techniques, map-aided localization plays a significant role in autonomous navigation, as priori maps can provide precise road information for a navigation system and enable it to make reasonable decisions. However, the map-aided localization technique remains elusive, as there are no robust solutions for creating 3D high-definition road maps. In addition, the unique data structure and volume of LiDAR point clouds also increase the complexity of this important task.

This study provides a semi-automated method for extracting meaningful road features and create 3D high-definition road maps for autonomous vehicles. The proposed method first removes non-ground points through a voxel-based upward growing algorithm. Then a road edge detection algorithm is performed to detect road curbs and segment road surfaces. Next, road markings, including arrows, symbols, and words, are preliminarily detected using a global intensity filter. Semantic clusters of these textual and directional road markings are extracted by the statistical analysis outlier removal algorithm and conditional Euclidean clustering algorithm. On the other hand, lane markings are extracted by local intensity analysis and distance-to-road-edge

thresholding based on road design standards. Finally, centerline points in each single lane are estimated based on the position of the extracted lane markings. Accordingly, 3D high-definition road maps with precise road boundaries, road markings, and the estimated lane centerlines have been created.

Four datasets are tested to prove the robustness of the proposed road marking extraction method. The precision, recall, and $F_1$-score obtained using the four test datasets (1, 3, 4 and 5) are (97.78%, 96.57%, 0.972), (95.65%, 95.54%, 0.956), (96.83%, 89.72%, 0.931) and (85.21%, 93.22%, 0.890), respectively. The accuracy assessment results demonstrate the feasibility of the proposed method, which can successfully and accurately extract most targeted road features from the test datasets. However, the accuracy of the proposed method is influenced by data complexity. Curves, shadows, and different types of obstacles are significant factors that could affect the accuracy of the experimental results.

In conclusion, this thesis presents a stepwise workflow to extract road edges, road surfaces, road markings, and lane centerlines from MLS point clouds. It also provides a robust solution to create 3D high-definition road maps for autonomous navigation systems. As map-aided localization is still an emerging concept and there is no regulation or standards for creating 3D priori maps, the proposed idea has the potential to meet the demands of the market, especially for automobile manufacturers and designers like Google, Tesla, and Baidu, who are endeavoring to have commercial models of their driverless cars.

## 5.2 Contributions

The proposed methods and ideas contribute to the study of MLS point cloud processing, road feature extraction, and autonomous vehicles navigation. The contributions of this thesis are described as follows:

(1) **A revised curb detection and road surface segmentation algorithm**. As discussed in Chapter 3, a segmentation-based algorithm is used to detect curb points. Different from several previous segmentation-based curb detection methods, a voxel-based upward growing algorithm is conducted first to remove non-ground points before curb detection. This strategy could largely reduce the data complexity and the computing cost of further processes. Furthermore, according to the characteristics of test datasets, the lowest curb candidate on each side of a local block is regarded as the bottom of curbs. As the study areas are viaducts and urban roads with well-defined and highly organized road features, the algorithms and thresholding strategies are designed based on the characteristics of these road types. Moreover, the majority of segmentation-based methods mainly focus on simple straight roads, while the proposed workflow can deliver accurate results from all datasets, including complicated scenes with curves and obstacles. The experimental results prove the applicability and robustness of the method put forward in this thesis.

(2) **A unique lane marking extraction method based on road design standards**. It is noted that one of the biggest challenges to development 3D high-definition road maps is to extract lane-level road features from massive MLS point clouds. Traditional road marking extraction methods are usually based on global intensity analysis. This thesis presents a new algorithm that detect lane markings based on both intensity analysis and distance-to-road-edge thresholding, overcoming the uncertainty of distance-to-

trajectory dependant thresholding methods. The thresholds used in this algorithm are determined by regional specific road design standards, which drastically reduce the time consumption of optimal threshold estimation. Finally, high-accuracy lane marking extraction results are produced from all test datasets, proving the robustness and wide applicability of the proposed lane marking extraction algorithm.

(3) **A guideline for creating 3D high-definition road maps.** Since traditional road maps cannot meet the demand of autonomous driving, 3D high-definition road map has become a necessary element of map-aided localization. To accomplish autonomous driving, a diversity of road maps has been developed in recent years, including 2D infrared reflectivity map, road topology network map, and different types of aerial images. However, most of these road maps are 2D maps focusing on a certain road feature. This study extracts different road features and estimates lane centerlines for autonomous vehicles, offering an example of a 3D high-definition road map. It may also provide useful suggestions for researchers and companies in terms of creating priori maps in the future.

### 5.3 Limitations and Future Work

As reported in Chapter 4, some limitations still exist in this approach. The detailed limitations and recommendations for future research are described in this section.

(1) The proposed method is mainly tested on urban roads and viaducts. Therefore, the proposed method might not be suitable for other road types, such as rural or coastal roads. Necessary changes should be made based on the characteristics of data to produce more accurate results for other road types.

(2) As discussed in Chapter 4, the accuracy of the experimental results is relatively lower in complicated scenes, such as curved roads. Also, the incomplete road feature extraction results show that the proposed method is affected by the occlusion of obstacles, such as moving vehicles. Therefore, the algorithms still need improvement to handle high-complexity scenes.

(3) The efficiency of the proposed method needs further improvement for higher computing efficiency, especially when dealing with curved roads. In order to improve the road edge detection performance, the road surfaces are usually segmented into smaller pieces, which means a considerable number of blocks are generated to process data locally. As a result, more computing time will be spent on the segmentation process. In the future, without reducing accuracy, more efficient algorithms that can process data globally should be developed.

(4) The proposed method relies on a priori knowledge and experience. A small part of human input is still required, such as estimating and setting parameters. Moreover, the parameters used in this study can have great influence on the performance of the algorithms. Thus, a parameter-free method is needed to decrease the sensitivity of parameters. Fully-automated algorithms are expected to be developed in future research.

(5) The true colour images acquired by the four digital cameras of the mobile laser scanning system should be taken into consideration. By integration of such images with the point clouds, the detection of road curbs and road markings may also be improved. Therefore, it is suggested that the digital images and image-based feature recognition methods could be used to improve the performance of the proposed method.

# References


ArcGIS, 2016. *ArcGIS Help 10.2, 10.2.1, and 10.2.2*. Retrieved from ArcGIS Resources.

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

BCG, 2017. *Autonomous Vehicle Adoption Study*. Retrieved from The Boston Consulting Group: https://www.bcg.com/expertise/industries/automotive/autonomous-vehicle-adoption-study.aspx

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

Chen, X., Kohlmeyer, B., Stroila, M., Alwar, N., Wang, R., & Bach, J., 2009. Next generation map making: geo-referenced ground-level LIDAR point clouds for automatic retro-reflective road feature extraction. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 488-491.

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.

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

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

Dickmanns, E. D., & Zapp, A., 1987. Autonomous high speed road vehicle guidance by computer vision. *IFAC Proceedings Volumes*, 20(5), 221-226.

Dickmanns, E. D., 2007. *Dynamic Vision for Perception and Control of Motion*. Springer Science & Business Media.

Fischler, M. A., & Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381-395.

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

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

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

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

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

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

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

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

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

Haala, N., Peter, M., Kremer, J., & Hunter, G., 2008. Mobile LiDAR mapping for 3D point cloud collection in urban areas—a performance test. *ISPRS Archives*, 37, 1119-1127.

Hata, A. Y., Osorio, F. S., & Wolf, D. F., 2014. Robust curb detection and vehicle localization in urban environments. In *Proceedings of 2014 IEEE Intelligent Vehicles Symposium*. 1257-1262.

Here, 2016. *A sample of 3D high-definition road map*. Retrieved from HERE introduction: https://here.com/en/products-services/products/here-hd-live-map

Huang, X., Gruen, A., Du, R. Q., & Fang, W., 2013. Integration of mobile laser scanning data with UAV imagery for very high resolution 3D city modeling. The *7th International Symposium on Mobile Mapping Technology*, Tainan, Taiwan, 1-3.

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

Ieng, S. S., Tarel, J. P., & Labayrade, R. 2003. On the design of a single lane-markings detectors regardless the on-board camera's position. In *2003 Intelligent Vehicles Symposium*, 564-569.

Kammel, S., & Pitzer, B., 2008. LiDAR-based lane marker detection and mapping. In *2008 IEEE Intelligent Vehicles Symposium*, 1137-1142.

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

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

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

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

Levinson, J., & Thrun, S, 2010. Robust vehicle localization in urban environments using probabilistic maps. In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, 4372-4378.

Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., & Sokolsky, M., 2011, June. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium*, 163-168.

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.

Lindner, P., Richter, E., Wanielik, G., Takagi, K., & Isogai, A., 2009. Multi-channel LiDAR processing for lane detection and estimation. In *12th International IEEE Conference on Intelligent Transportation Systems*, 1-6.

Loose, H., & Franke, U., 2010. B-spline-based road model for 3D lane recognition. In *2010 13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 91-98.

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

Manz, M., Luettel, T., von Hundelshausen, F., & Wuensche, H. J., 2011. Monocular model-based 3D vehicle tracking for autonomous vehicles in unstructured environment. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2465-2471.

Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S. & Johnston, D., 2008. Junior: The Stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9), 569-597.

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

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

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

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

Otsu, N., 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62-66.

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

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

Pink, O., Moosmann, F., & Bachmann, A., 2009. Visual features for vehicle localization and ego-motion estimation. In *2009 IEEE Intelligent Vehicles Symposium*, 254-260.

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

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

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

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

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

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

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

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

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

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

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.

Sprickerhof, J., Nüchter, A., Lingemann, K., & Hertzberg, J. (2011). A Heuristic Loop Closing Technique for Large-Scale 6D SLAM. Automatika: *Journal for Control, Measurement, Electronics, Computing & Communications*, 52(3), 199-222.

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

Tao, Z., Bonnifait, P., Fremont, V., & Ibanez-Guzman, J., 2013. Mapping and localization using GPS, lane markings and proprioceptive sensors. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 406-412.

Tao, Z., Bonnifait, P., Fremont, V., & Ibanez-Guzman, J., 2013. Lane marking aided vehicle localization. *IEEE International Conference on Intelligent Transportation Systems*, 1509-1515.

Thorpe, C., Herbert, M., Kanade, T., & Shafer, S., 1991. Toward autonomous driving: the CMU Navlab. i. perception. *IEEE Expert*, 6(4), 31-42.

Thrun, S., 2003. Learning occupancy grid maps with forward sensor models. *Autonomous Robots*, 15(2), 111-127.

Thuy, M., & León, F., 2010. Lane detection and tracking based on LiDAR data. *Metrology and Measurement Systems*, 17(3), 311-321.

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M. N., ... & Gittleman, M. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8), 425-466.

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

Von Hundelshausen, F., Himmelsbach, M., Hecker, F., Mueller, A., & Wuensche, H. J., 2008. Driving with tentacles: Integral structures for sensing and motion. *Journal of Field Robotics*, 25(9), 640-673.

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

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

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

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

Yen, K. S., Ravani, B. & Lasky, T. A., 2011. *LiDAR for Data Efficiency*. WSDOT Research Report, No. WA-RD 778.1, Washington State Department of Transportation.

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

Ziegler, J., & Werling, M., 2008. Navigating car-like robots in unstructured environments using an obstacle sensitive cost function. In *2008 IEEE Intelligent Vehicles Symposium*, 787-791.

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

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

Zhou, M., 2016. *Semi-automated extraction of 3D building windows from MLS data*, MSc Thesis, Department of Geography and Environmental Management, University of Waterloo.

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.