Precise 3D Modeling and Feature Extraction based on Mobile Mapping Data in Road Environment

（モービルマッピングデータに基づく道路の高精度3次元モデリングと特徴抽出）

Submitted in partial fulfillment of the requirements

of the degree of

Doctor of Philosophy

by

蘇　景昕

Department of Information Engineering

HIROSHIMA UNIVERSITY

(March 2020)

Dedicated to

All who have helped on the path.

Declaration

I declare that this dissertation represents my original work except where states otherwise by reference or acknowledgement, and that this work has not been submitted for any previous degree.

I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any source in my submission. I understand that any violation of the above will be cause for disciplinary action by the University. Parts of the presented work have been published in:

- Journal: International Journal of Automation Technology; Accepted; 2018.
- Journal: Sensors Volume 19 Issue 24; Accepted; 2019.
- Conference: The International Workshop on Frontiers of Computer Vision; Korea; Feb 2019.

Su Jingxin

December 2019

Acknowledgements

I would like to express my sincere appreciation to my supervisor Prof.Kaneda, for his guidance, encouragement and continuous support through my studies, and to my co-supervisor, Associate Professor Tamaki, for his useful suggestions and comments. I also would like to thank Prof. Harada for his advice, not only academically, but also personally.

I must express the deepest appreciation to Dr.Miyazaki for his advice, help and time.

I thank all the members in Information Engineering Department, including the professors, staffs and students. It was a great experience to study with people from different countries, that we have different language, culture, religion.

Finally, I dedicate this thesis to my family and my friends, for their love and support through my life.

The mobile mapping datasets used in this work are provided by Sanei co. Japan.

Abstract

In recent years, autonomous driving has become an increasingly important topic in the academia and industry. While various techniques have been developed to handle different types of issues, it is recognized that stability and reliability are both key consideration in the assessment of systems. However, to help the self-driving car development and evaluation, a high accuracy and high precision three-dimensional model of the road surface is necessary and very valuable. The aim of the present study is to outline a workflow for 3D modeling road surface based on mobile mapping data.

In Chapter 1, we introduce the definition of road characteristics and environmental factors and motivation of our study.

In Chapter 2, we propose an approach for precise 3D modeling of lane marks by combining both image and point cloud features. Instead of applying only 2D image processing techniques for lane markings detection, we introduce a line-based point cloud region growing method and an image-based scan-line method for the extraction. The line-based point cloud region growing is used to identify boundary points, which guarantees a precise road surface region segmentation and boundary points extraction. The image-based scan-line algorithm is designed specifically for the environments where it is difficult to clearly identify the road surface region. The road surface points are projected onto color image to find precise lane mark region. Then, we perform an inverse projection to recover the 3D coordinates of the detected 2D lane mark points. Quantitative evaluation is conducted with comparison of the proposed method and a region of interest method.

In Chapter 3, we use the obtained 3D lane marking model for the road trajectory estimation directly. A morphological closing operation is used to refine the lane marks. Next, using the length and angle information of points, we produce a two-dimensional representation of the 3D points. This allows us to apply mature two-dimensional algorithm to obtain the missing points. Points are represented in a length-angle space in order to estimate the points between broken white lines. Considering that the road trajectories are the sequences consists of centerline points. Finally, we generate a 3D point sequence to represent the trajectory points. Since there is no ground truth data and design curvature of the road trajectory, we manually build a reference dataset for the comparison and evaluation. The result shows that a continuous-curvature trajectory was obtained by the proposed method.

In Chapter 4, we make use of road trajectory points the reference line and use a curved regular grid (CRG) model to get a complete view of road surface that contains geographic information of road surface and use elevation information to show the shape of road surface. We firstly apply a robust and effective method that can divide road into three road segment categories: straight line, circular arc and clothoid curve. To create CRG model file, a regular elevation grid which provides elevation values is needed. To effectively and accurately interpolate the elevation values, the regular grid is created using a non-regular gird. In the next step, we estimate the regular grid from the non-regular grid by applying the bilinear interpolation method. Experimental results on real world road scenarios is evaluated based on the root-mean-square error between the proposed method and an existing nearest-neighbour search method. The numerical analysis shows that the proposed approaches are capable of accurately modeling the road surface.

Finally, we build a precise road model that contains geographic information of road surface features and use elevation information to show the shape of road surface in high-resolution. The next step in future development, we may further build the road network database. Some conclusions and final remarks are provided in Chapter 5.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation

As mobile mapping system become a mature technology, there are many applications for the process of the measured data, such as advanced driver-assistance systems (ADAS), driving simulator and autonomous driving. Since all the vehicle related techniques have very high demand on high precision, a high accuracy and high resolution road model with detailed information about the surrounding environment can play an important role in this research field.

However, two significant issues stands in way of 3D road modeling, one is the precision, another is the efficiency. In order to meet the requirement of precise road surface measurements, we use point clouds as input data in this work. Point cloud data consists of a set of measured points in three-dimensional space. The point cloud is one of the most widely used data types in three-dimensional image processing.

The mobile mapping system (MMS) produce large-scale 3D point clouds and very high precision geometric measurements. A comprehensive overview of recent research in MMSs and surveying technologies is given in [1]. Some examples of the new technologies and systems are presented in [2-5]. These systems produce large-scale 3D point clouds and high-precision geometric measurements. The produced point clouds are used for many road related research tasks, including for missing road point regions detection [6,7], for

road damage information detection [8-10], for road segmentation and recognition [11-17], etc.

There have been a lot of efforts to create and manipulate the road model in the community. Many of them are still at the proof-of-concept stage, those proposed methods can be improved on both aspects significantly. From another perspective, many of the efforts are focused on specific type of data. This makes it difficult to achieve that maintain both precision and efficiency in high performance. Considering the variety information collected by mobile mapping system, a method employ multiple data sources is meaningful and is possible to overcome the limitations.

Consequently, the intention of this thesis is to address the issue of modeling 3D road surface model with high-precision and high-accuracy. We propose several methods based on combinations of different data sources. We also build models stepwisely which guarantees the adequacy and accuracy.

1.1.1 3D Modeling of Lane Marks

When we drive a car, the white lines on the road show us where the lanes are. The lane marks are act as a reference for where to steer the vehicle. Naturally, in the developing of ADAS and autonomous driving, Lane Keeping Assist System (LKAS) and Lane Departure Warning (LDW) play important roles in preventing accidents, therefore, lane mark detection has become a critical issue.

Many papers have presented their solutions of lane mark extraction [2,14,18-21]. Mainly, there are two types of methods have been proposed: 2D image processing based approaches [18-20] and methods using 3D point cloud data [13,14,21]. Only a few researchers have combined images and point clouds for the lane marking detection [2,15]. For instance, a framework is presented in [15] that includes road point selection and lane mark extraction. First, they selected the road points by using a thresholding algorithm. Next, the lane marks were extracted using a template matching method with the help of an intensity image of the point cloud data. The image-based methods are effective when the road surface region is accurately defined, but because the color information is highly sensitive to environmental conditions, it is often difficult to extract accurate road surface region using only the color information of the image. The 3D model based approaches can remove the environmental noises easily, but most of the above mentioned methods require human intervention (i.e., manual selection of regions of interest (ROI) and other information, such as the intensity values of points collected by specific types of points or colored point cloud scanners). In addition, point intensity values and colored point clouds are highly sensitive to the distance between the measurement point and the laser scanner. Therefore, in order to obtain high-accuracy 3D lane mark model, we have proposed a method for 3D

modeling of lane marks by combining both image and point cloud features.

1.1.2 3D Modeling of Road Trajectory

In theory, the ADASs have the potential to significantly reduce traffic accidents by lane detection techniques, but one thing we have noticed about the ADAS is that it is not capable of fully replacing human drivers. In recent years, many self-driving car involved accidents are reported, and it is the type of accident that autonomous vehicles are supposed to avoid, this is due to the complex road environment.

To help the ADAS development and evaluation, an accurate estimation of the trajectory of the road is needed. To allow the correction and analysis of ADAS, the most commonly used solutions are the global navigation satellite system (GNSS) and iterative closest point (ICP) algorithm. In a complex urban environment, the loss of GNSS signals should be considered. And point clouds can never align perfectly using ICP algorithm. [22] used a strategy based on the trajectory loop information and ICP registration. This motivates us to generate a 3D point sequence to represent the road trajectory. Thus, to overcome the aforementioned limitations, we proposed a stepwise method, which generates the 3D road trajectory step by step.

1.1.3 Road Modeling and Representation in Curved Regular Grid Model

Since the precise road trajectory is successfully extracted, we can use the trajectory as the reference line in road surface modeling. There are many ways to build detailed representations of road models, and they may produce different models for various purposes. For example, in [23,24], the authors presented approaches for grid-based road model estimation for ADASs. Their measurements from sensors are transformed into a grid-based road model and a geometrical description is extracted out of this model by the use of a path-planning based method. For other uses, some authors proposed methods using the elevation information to building road models [25-28]. It has been proven that the elevation-based methods are suitable for detection techniques.

In [29,30], the authors demonstrate a workflow for representing point cloud data in a curved regular grid (CRG) model. The input laser-scanned point clouds and geometric description of the road both need strong manual intervention in the preparation phase. The selected pilot road has a near perpendicular segment, and the elevation values are calculated by a fixed-radius nearest-neighbour search algorithm from the input point cloud directly. The mean elevation values of points inside the circle are stored in the curved regular grid cells. This also causes the problem that the resolution of the generated CRG model is limited by the density of the input point cloud. However, the performance

and accuracy of the two key steps—road segmentation and elevation estimation—can be further enhanced. Therefore, with the help of OpenCRG [31-33], we presented a method for representing three-dimensional road data in CRG models, which can generate high-accuracy and high-resolution road model.

1.2 Structure of the Thesis

In this thesis, we address the issue of modeling 3D road surface model with high-precision and high-accuracy. A complete workflow is presented to build the road model from scratch. This thesis consists of 5 chapters and is organized as follows:

In Chapter 1, we introduced the objective of our research.

In Chapter 2, we firstly give an overview of the input dataset. A line-based region growing method based on three-dimensional point cloud data information is introduced for defining a precise road surface. After this, to improve the efficiency, we propose a image-based scan-line algorithm to extract the road surface region. The evaluation is based on a comparison of our method and a region of interest method.

Chapter 3 follows the study on 3D modeling of road features. In this chapter, we propose a method that extracts the trajectory points of the road. For the missing points between broken white lines, we produce a two-dimensional representation of the 3D points by a length-angle space. The main focus here is the continuous-curvature profile of the trajectory points. We discuss the importance of the continuous-curvature by the observation of a comparison reference data.

Chapter 4 extends the usage of road trajectory points to build a road surface model with elevation information. A short overview of OpenCRG project shows the recent research efforts in the industry and academia. The requirement of creating CRG model is introduced here. In the following section, we discuss the difficulty of find the regular elevation grid from input point cloud directly. We then introduce a non-regular grid to estimate the regular grid. Comparing with the existing method, the increase of both accuracy and efficiency is discussed by a numerical analysis.

Chapter 5 presents the conclusions of the research described in the thesis, and future directions are discussed.

CHAPTER 2

3D Modeling of Lane Marks using a Combination of Images and Mobile Mapping Data

2.1 Introduction

There has been a large increase in the use of point cloud datasets in three-dimensional (3D) image processing techniques. This is due to the advance of 3D scanning technologies advance. Today, many affordable and accurate laser scanning systems have become available for research or practical use. Mobile mapping systems are one of the most popular surveying devices for capturing large-scale point clouds and digital images in urban environments, especially for road condition surveying. It typically has two or more laser scanners and Charge-coupled device (CCD) cameras that are easy to mount on a variety of vehicles. Global positioning system (GPS) and Inertial measurement unit (IMU) are also combined in the system that is used to collect angle and acceleration data. This system can quickly capture the geometry of the road and its surroundings. The generated point cloud data representing roads and their surroundings can often contain millions of 3D vertices. A review of a recently available MMS and surveying technologies can be found in [1]. Some of the newly developed and presented systems are [2-5]. The systems produce large-scale 3D point clouds and high precision geometric measurements. These data are very helpful for road inspection tasks. Recently, there have been active research efforts to perform road related processing, such as road segmentation, surface reconstruction, and classification of valuable items. In [8], road damage information is extracted

from image data. In [11,12], laser scanning data are used to detect vertical pole-like objects beside the road. In [34], a LiDAR scanner was used to extract roads in a large-scale urban environment. However, it has been recognized that MMS is useful in describing the characteristics of complex urban road environments.

Although there are many interesting topics related to MMS, intelligent vehicles oriented topics have attracted wide attention. In particular, lane mark detection is a crucial topic in this research field. A number of papers have addressed the issue of lane mark extraction from [2,14,18-21]. There are two main types of approaches that have been proposed: (i) approaches based on image processing [18-20], and (ii) methods using 3D point cloud data [13,14,21]. Only a few researchers have combined both images and point clouds for lane mark detection [2,15]. The authors of [2] designed a MMS to acquire accurate camera and laser-scanner positions. The road line locations were then calculated by using both the image and the 3D road surface model information collected by MMS. A framework is presented in [15] that includes road point selection and lane mark extraction. First, they selected the road points by using a thresholding algorithm. Next, the lane marks were extracted using a template matching method with the help of an intensity image of the point cloud data. The image-based methods are efficient when the road surface region is accurately-defined, but it is often difficult to extract precise road regions using only color information of an image because color information is highly sensitive to environmental conditions. The 3D model-based approaches can deal with the environmental noises, but most of the aforementioned approaches require a manually selected region of interest (ROI) and additional information such as the intensity value of points or a colored point cloud collected by a specific type of scanner. Furthermore, the intensity value of points and the colored point cloud are highly sensitive to the distance between the measured points and laser scanner. Therefore, it is difficult to precisely extract lane marks with existing methods.

In order to obtain a 3D trajectory or orbit information of a vehicle, high accuracy lane mark detection is demanded for advanced driver-assistance systems and autonomous driving. Three-dimensional lane mark models can be used as a reference to create such trajectory or orbit information. They can not only aid steering to stay within the lane, but can also control the vehicle to follow the road accurately in self-driving mode.

Therefore, we have performed this research. The objective of this work is to derive an approach for precise 3D modeling of lane marks by combining both image and point cloud features. In this work, we develop a fast, precise and effective method to create 3D point cloud models of lane marks. The main idea is to extract a precise road surface region from the image using the combination of the color image and point cloud data. According to the features of point cloud data, three-dimensional information is used for defining a

precise road surface region, and we adopt a line-based region growing method to extract the region [17], which provides accurate boundary points. Such an information is used as the constraints for representing the boundary edge in the process of road surface region extraction from the image. Then, to extract the road surface region from the image, a specialized scan-line algorithm is used. After that, a binarization method is applied for lane mark extraction. Finally, the extracted lane marks are converted into 3D models.

In summary, the main contributions of this work are as follows. First of all, we propose a novel road surface region extraction method from the image that combines information from the color image and point cloud data. Second, the specialized scan-line algorithm reduces the execution time of the whole procedure significantly. It can also improve the accuracy of the road surface region edge estimation slightly. Finally, we perform experiments on a complex urban road dataset. Precise 3D lane mark models are created. With the guidance of these models, we can educe the overall perspective of road track.

The rest of this chapter is organized as follows. Section 2.2 presents the proposed approach in detail. The experimental results of extractions are provided in Section 2.3, and Section 2.4 concludes this work.

2.2 Method

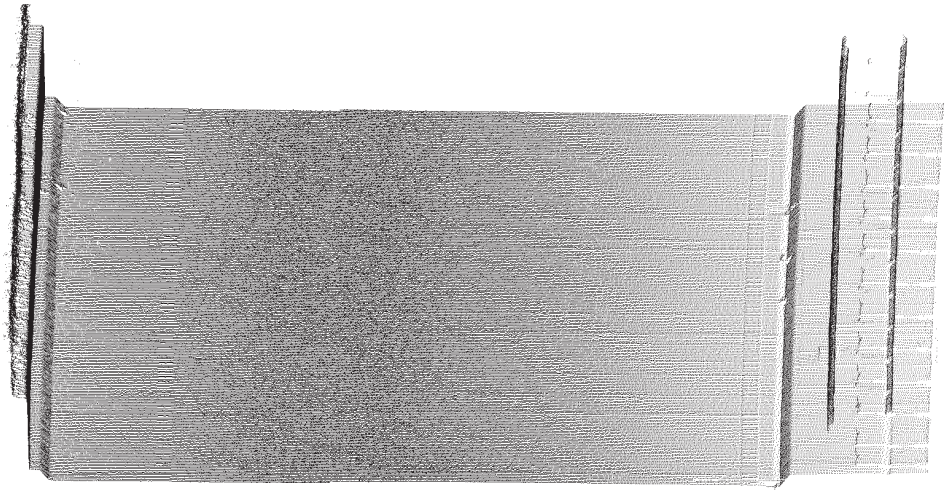
2.2.1 Datasets

The point cloud data and images used here are captured using a 3D laser scanner and CCD camera. The benefit of a 3D laser scanner is that the vehicle position is known and can be used for information on the road's location and orientation.

The whole dataset comprised of approximately 91.25 million points. The length of the road is approximately 2,900 m. To cope with the large size of raw point data, the collected data is automatically divided into volumes. Each volume has around 300,000 points that collectively represent about 8 m of the road and its surroundings.

Each point not only has a 3D coordinate but also has a laser irradiation angle and GPS time. This information can be used to structurize the points. From this information, we use the laser irradiation angle in order to separate the point cloud into scanlines. Moreover, we order points in a scanline and find the neighborhood elements by laser irradiation angle information. The point cloud data and color image used in our research are illustrated in Figure 2.1.

However, the interval of measured points along the direction the MMS travels depends on the rotating speed of the laser irradiation part and the speed at which the MMS travels. The rotation period of the laser irradiation part is much longer than the laser irradiation



(a) Point cloud data collected by laser scanner

(b) Color image collected by camera

Fig. 2.1 Data obtained by MMS

period. The measurement interval along the direction the MMS travels is often a few hundreds of millimeters. Thus, the density of point distribution is greatly unbalanced with the direction Figure 2.2.

This uneven distribution causes a problem when calculating geometric information using neighboring points because it is necessary to define a neighborhood range that is too wide in the sparse direction. For example, if the range of the neighborhood is defined too widely in estimating the normal vector, we cannot estimate the vector precisely at the point on the planar region around the boundary of the structures. Figure 2.3 shows an example of such a case. The estimated normal vector is tilting because the neighborhood region includes points of a vertical plane even though the target point is on the horizontal plane. We solve this problem by employing a local best-fit plane, which is described in

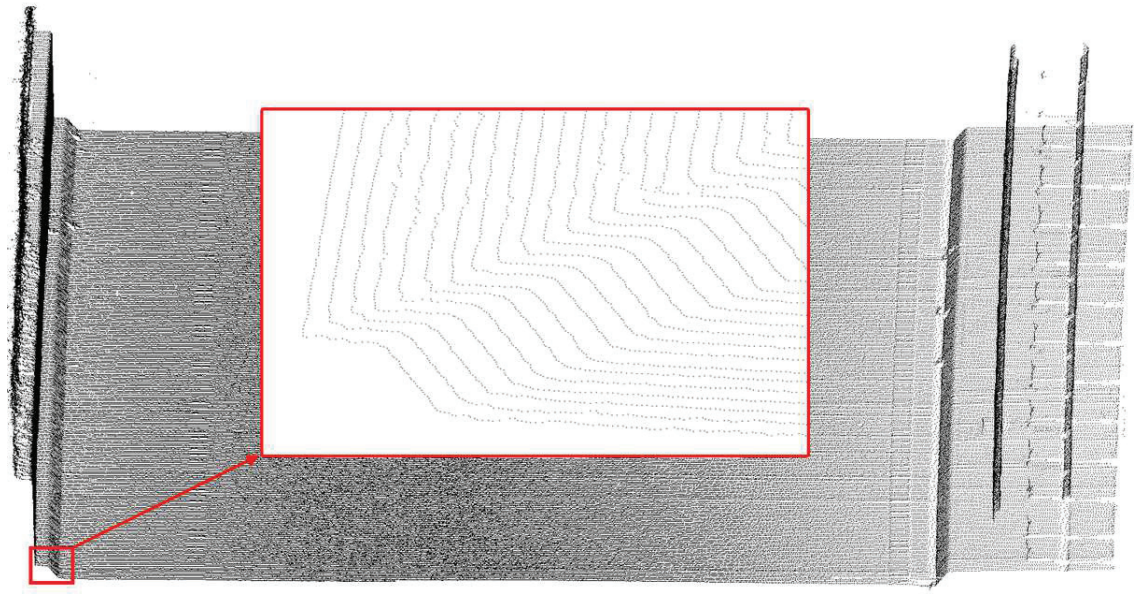


Fig. 2.2 The difference in point density according to the direction.

Section 2.3.

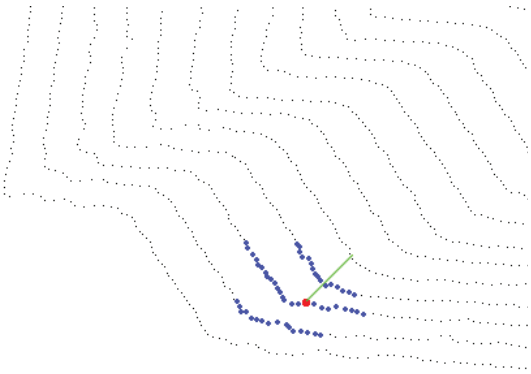


Fig. 2.3 The estimated normal vector at the point around the boundary. The green line is the normal vector at the point colored red. Small blue dots are the neighborhood points used for estimating the normal vector.

2.2.2 Overview of method

Our method takes a sequence of point clouds and images as input. The GPS time stamp is used to find an image and its corresponding point clouds that were captured during a certain time interval. For each volume of raw point cloud data, we find an image to which all 3D points can be projected to this image.

We first apply a line-based region growing approach for road surface region extraction from point cloud data. The end points of each scanline from the extracted point cloud explicitly represents the edge points between the road and curb.

We then convert the 3D end-point coordinates into two-dimensional (2D) space. We can also project the end points from point cloud onto image.

The converted end points are considered as the input of our specialized scan-line algorithm. For each side of road boundary, we initialize an edge list that holds end points and compute the intersections for each edge segment with scan-line. The road surface region is then extracted with the scan-line algorithm.

Because the extracted road surface region images used here are taken along the road, we can assume that the lane marks have the highest intensity pixel value. A thresholding binarization method is used to detect the lane marks from the extracted road surface region image. Finally, we extract the 3D points corresponding to the detected lane marks.

The pipeline of processing steps is depicted in Figure 2.4.

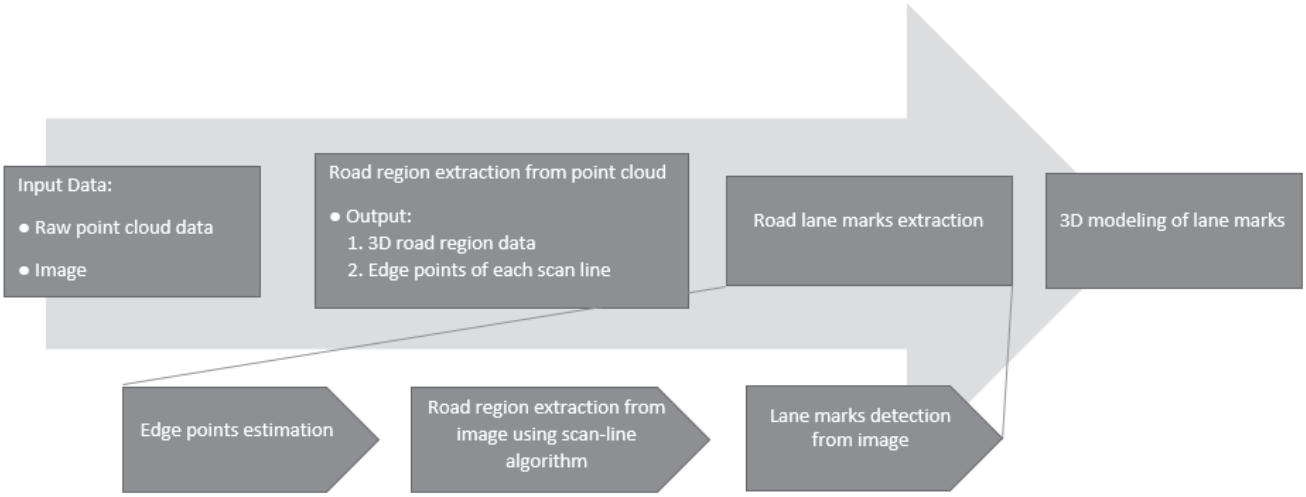


Fig. 2.4 Processing pipeline.

2.2.3 Line-based region growing

The point clouds used here are collected by the Z+F IMAGER 5010 laser scanning system. The system outputs approximately one million points per second and the rotation speed of the laser head is 50 rps. Each point has a 3D coordinate and some additional information, including laser irradiation, the GPS time, and distance between the source point of the laser and a sampled point. We assume a geometric discontinuity at the boundary of the road. In that case, the region growing process can stop exactly at a boundary point.

To find the location of the boundary points precisely, we use a line-based region growing method [35, 36] to extract the road surface region from the point cloud. Miyazaki et al. proposed the method mentioned in Section 2.1 to tackle such problems. Following this

method, the input to our algorithm is a set of line segments. We first create line segments from a point sequence using the angle of laser irradiation. We then use the line segments as processing elements for the road surface region extraction.

For searching of neighborhood line segments, we use the laser irradiation angle associated directly with sampled points. If two points on two consecutive scanning lines have a similar laser irradiation angle, these points are considered to be located near each other.

In the region growing approach, normal vector estimation is a crucial step. The difference between the angles of the normal vectors is used to determine whether a neighborhood should be added to the region. Least-squares fitting for neighbors is often used for normal vector estimation. However, such a method is unable to derive a precise estimation of a normal vector for our purposes. Thus, we adopt the local best-fit plane of the neighboring line segments to estimate the normal vector of a line segment. The local best-fit plane is defined as the plane that passes through the seed line segment and includes the most neighboring line segments [35, 36]. The normal vector in the local best-fit plane is used as the normal vector of the line segment. Figure 2.5 shows an example of the line segments and normal vector estimation.

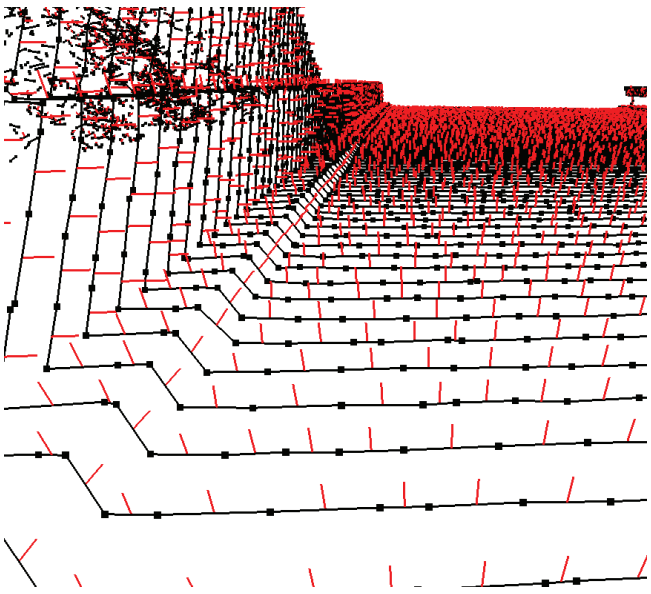


Fig. 2.5 Example of the line segments and normal vector estimation.

After estimating the normal vector for the line segments by using the local best-fit plane, we calculate the difference between the angles of the normal vectors to determine if a neighboring line segment should be added to the region.

The common region growing approach starts with a selected seed. In our case, we select the seed segment with the largest degree of fitting from the input line segment that has not yet been assigned to any region.

2.2.4 Scan-line algorithm

Scan-line algorithm, also known as scan-line rendering or scan-line fill algorithm, is a widely used shaded region determination algorithm. This algorithm works by intersecting the scan-line with polygon edges and filling the polygon between pairs of intersections. Most scan-line algorithms are designed for 3D and 2D image rendering.

In our case, each pair of the road surface region boundary points can be seen as two end points of a segment of the polygon edge. Therefore, we consider those segments as the edges of the road surface region. The flowchart of an algorithm based on this idea is shown in Figure 2.6.

All scan-lines in this algorithm are horizontal. Because of the setting of MMS, the line segments in the point cloud obtained by MMS and the scan-lines are not parallel. Hence, we consider the projected road surface region to be a trapezoid shape, and then apply a scan-line based approach for precise road surface region extraction from the projected 2D image. Figure 2.7 illustrates the idea of the scan-line algorithm with example data.

In the case of a typical scan-line algorithm, to cope with multiple intersections, an active edge list was maintained for edges that cross the current scan-line. However, for a trapezoid-shaped road surface region, there are only two intersections with a scan-line. Therefore, the edges can be held by two edge lists. We can contain the active edges using pointers to the edge lists. Such a data structure can reduce the execution time and the amount of memory used.

Boundary points of the projected road surface region represent precise end points of the road surface region. As a preprocessing step, this algorithm creates edges between each pairs of adjacent boundary points and two edge lists needs to be created.

We hold the edges formed by the left and right end points in two edge lists. The data structure of the edge lists is shown in Figure 2.8(a). To handle the parallel sides of road surface region, preprocessing is needed. The parallel side edges in Figure 2.7 are decreasing, i.e., the slope is negative. Then, first element of the right edge list will be inserted to the top of the left edge list and the last element of the left edge list will be added to the bottom of the right edge list. Figure 2.8(b) illustrates the special handling.

After the edge lists were created, all edges were sorted according to their minimum y-value $miny$ and maximum y-value $maxy$. Next, the algorithm needs a range of scan-lines spanning the region $(Sybottom, Sytop)$, where $Sybottom$ and $Sytop$ are the lowest and the highest y values of the end points, respectively. We process the scan line from bottom to top in increasing y order. Instead of a traditional active-edge list, the active-edge pointers to the edge lists are needed here. The pointers contain the active edges crossed by current

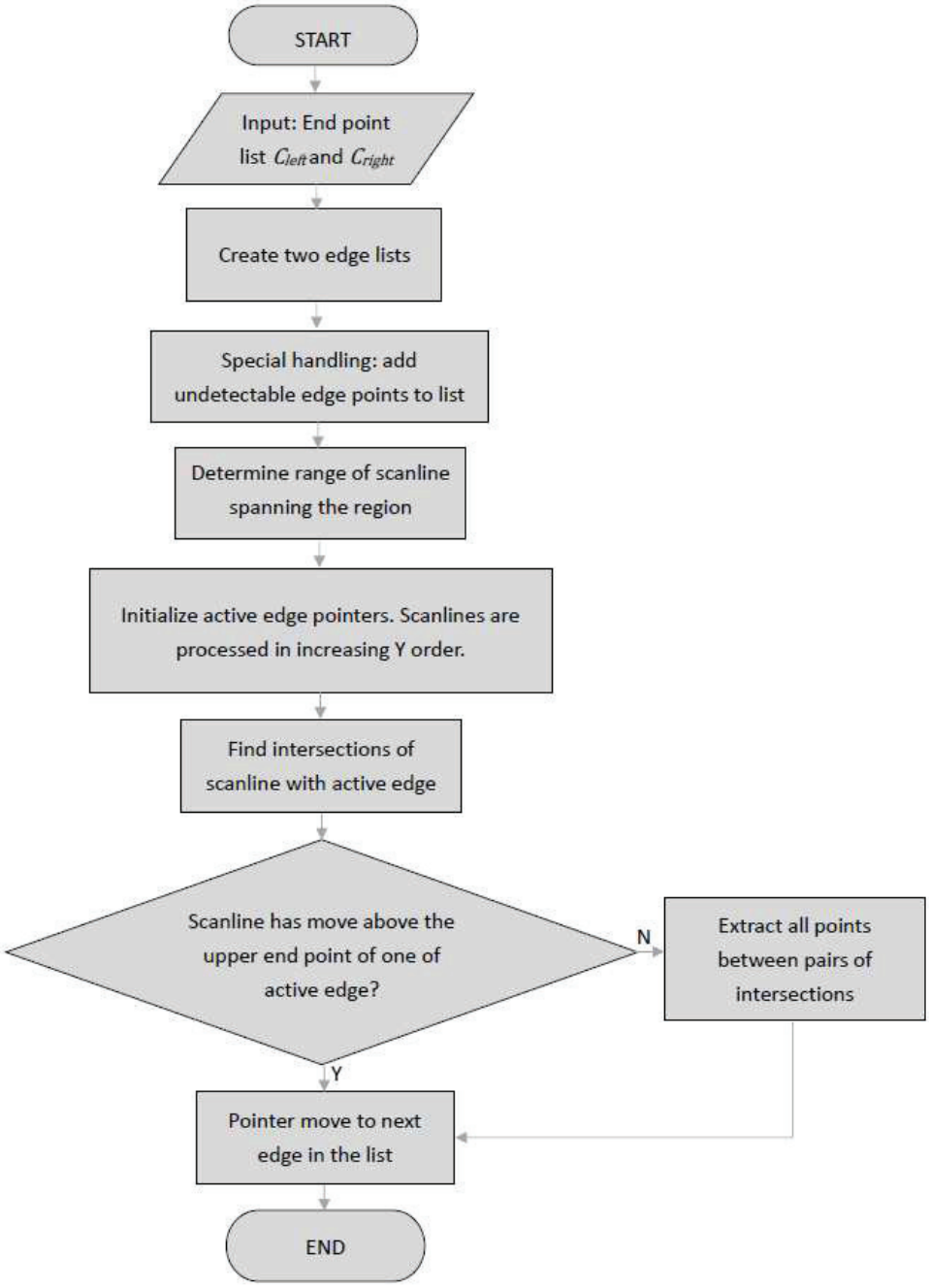


Fig. 2.6 Flowchart of the scan-line algorithm.

scan line. The algorithm first initializes two points to the edge lists according to *miny*. Second, the algorithm finds the intersections of the scan-line with each active edge, and then extracts all the points between the pairs of intersections. When the current scan line moves above the upper endpoint of an active edge, then, it becomes inactive. The pointer moves to the next edge in the list. Finally, the scan-line reaches *Sytop* and obtains the road surface region from the image. Pseudocode for this scan-line algorithm is shown in Figure 2.9.

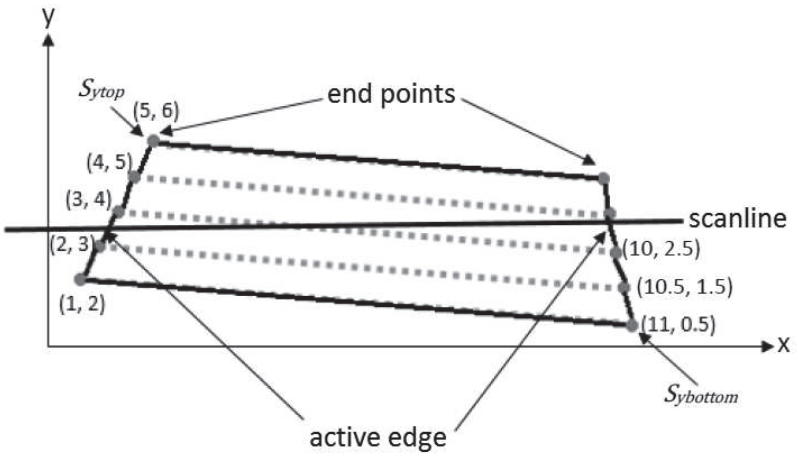
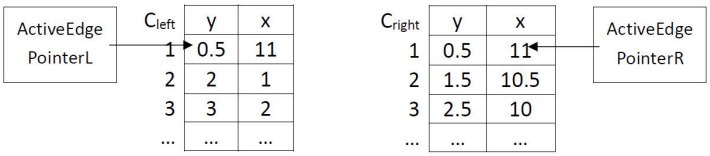


Fig. 2.7 Idea of the scanline algorithm with example data.

C_{left}	y	x
1	2	1
2	3	2
3	4	3
...	...	...

C_{right}	y	x
1	0.5	11
2	1.5	10.5
3	2.5	10
...	...	...

(a) Data structure of edge lists



(b) Parallel sides handling

Fig. 2.8 Data structure and special handling of the scan-line algorithm

2.2.5 Binarization and lane marks detection

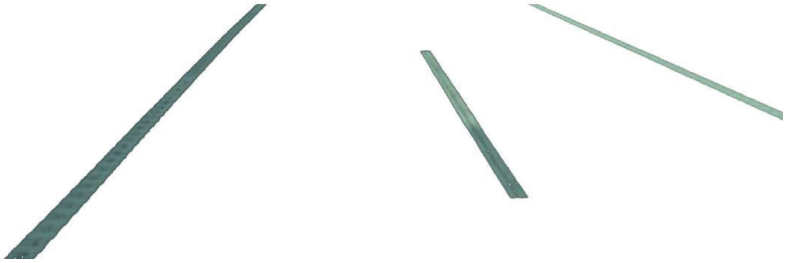
Because a white lane line is a bright object against a dark background, a binarization method is carried out to obtain the brighter regions. We generate a binary image of the extracted road surface region to find the lane mark locations. The thresholds are manually selected because the brightness of the roads are influenced by environmental lighting conditions. An example of a binary image of an extracted road surface region is shown in Figure 2.10(a), where each lane mark is indicated by a different color. Figure 2.10(b) shows the result of the extracted lane marks.

Algorithm 1 Pseudo-code for Scan-line algorithm

```
1: add boundary points to endpoint lists;
2: create two edge lists;
3: handle parallel side edges;
4: initialize active edge pointers activeL and activeR;
5: for (each scan-line S) do
6:     find intersections of S with active edges;
7:     extract points between pairs of intersections;
8:     move pointer activeL or activeR to next edge;
9: end for
```

Fig. 2.9 Pseudocode for the scan-line algorithm.

(a) Binarization result of extracted road surface region. Each line is indicated by a different color



(b) Extracted lane marks

Fig. 2.10 Result of lane mark extraction

2.2.6 3D lane marks points extraction

After lane mark extraction from the image, 3D lane mark models in our approach are represented as 3D point clouds. A 3D lane mark model is a set of 3D points that is lies in a 2D lane mark region when projected onto the image. Here, we perform an inverse projection to recover the 3D coordinates of the detected 2D lane mark points. The inverse projection is based on the information of projecting a 3D point onto an image and outputs 3D lane mark point cloud models.

2.3 Experiment Results and Discussion

In this paper, as mentioned in Section 2.3, the input point cloud is measured by an MMS equipped with the Z+F IMAGER 5010 laser scanning system. Figure 2.11 shows the input point cloud used by this experiment. This point cloud consists of approximately 2.1 million points. There are 21 color images used in the experiment. The color images were taken by the same MMS during data collection.

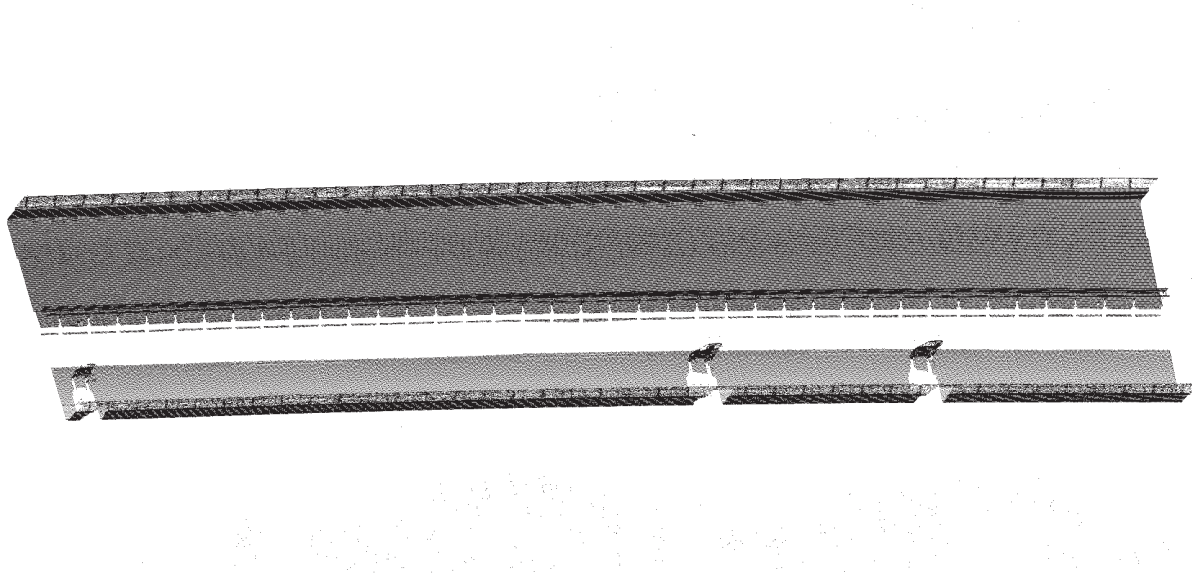


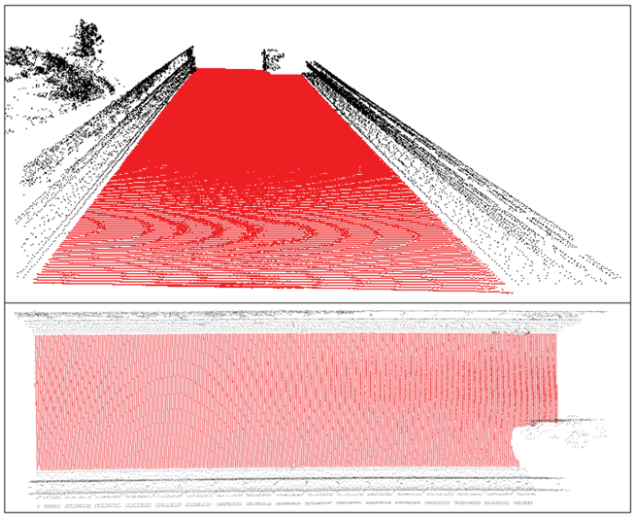
Fig. 2.11 The input point cloud.

Figure 2.12(a) is the extracted road surface region by line-based region growing. We paint the road surface points and non-road points in red and black, respectively. There are a total of 598,101 road surface points. We can see that the line-based region growing method successfully extracts the road surface region.

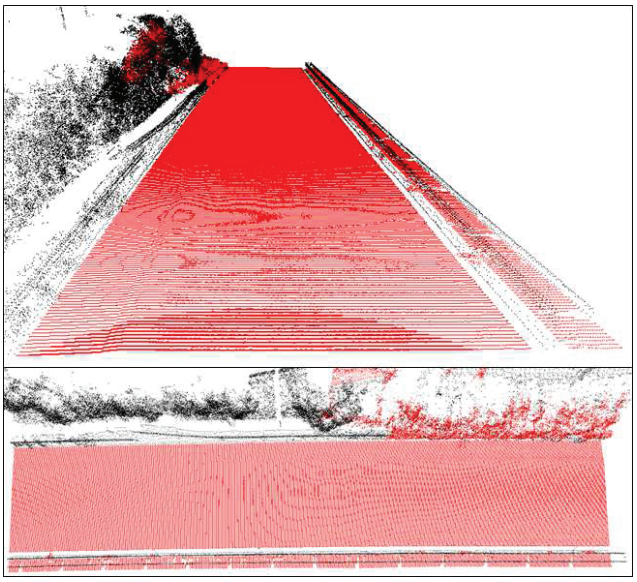
Because our region growing method uses points located at the boundary between road surface and curb, the limitation of our approach is that geometrical discontinuity is a necessary boundary of the road, as shown in Figure 2.12(b). In situations like this, the region may exceed the appropriate boundary.

2.3.1 Road surface region extraction

For evaluating the effect of the scan-line algorithm, we have extracted the road surface region from the image either with or without the scan-line algorithm. As a preprocessing step, we only need to project the end points to the image by using the scan-line algorithm. In contrast, without scan-line algorithm, we have to project all extracted road surface region points to the image during the preprocessing, and then extract the projected area from the image. The execution time of the experiment is illustrated in Table 2.1. In the



(a) The result of properly extracted region (Upper: perspective view; bottom: top view)



(b) The region exceeds the appropriate boundary (Upper: perspective view; bottom: top view)

Fig. 2.12 The results of road surface region extraction from point cloud

preprocessing step of the extraction method without the scan-line algorithm, the time cost of reading the point cloud data from raw data and projecting the 3D points to 2D image are included. For the scan-line algorithm, the end point extraction is also considered as a preprocessing step. It can be seen from Table 2.1 that 3D point projection is time consuming and the execution time for extraction without scan-line algorithm is longer than 15 s. Although many mature image processing methods can be used for the road surface region detection, the number of points that need to be projected onto the image is the most important issue in choosing the methodology. We can see that the scan-line algorithm reduces the execution time substantially. Figure 2.13 shows a sample result from the scan-line algorithm process on a road segment.

Table 2.1 Comparison of execution time.

	Preprocessing	Road surface region extraction
Extraction without scan-line algorithm	18.480 sec	0.106 sec
Extraction with scan-line algorithm	0.792 sec	0.098 sec

Fig. 2.13 The result of road surface region extraction from image.

2.3.2 Lane mark extraction

In [15,34,35], a manually selected ROI is used to define the road surface region in the image. Defining an ROI can help to reduce the number of false-positives in the extraction result. However, such a region cannot guarantee accurate extraction of lane marks under complex urban environments. We predefined the binarization threshold as 101 to detect the lane marks. To compare our method with other methods, the lane marks were also extracted using the ROI method. Figure 2.14(a) shows the result of lane mark detection with a precise road surface region. Figure 2.14(b) shows the result of lane mark extraction using a mask as an ROI filter. We specify a mask with the same height, width and position as our result. As shown in the figure, the guard rail and pavement are falsely included in the region.

In this experiment, we do not have any ground truth extraction results to compare with the experimental results. Thus, the reference data were manually created. To evaluate

(a) The result of lane mark detection with precise road surface region

(b) The result of lane mark detection using ROI

Fig. 2.14 Comparison of lane mark detection

the performance, we use precision, recall and F-measure scores as evaluation metrics. Quantitative evaluation was conducted and the results are illustrated in Table 2.2. It can be seen from Table 2.2 that the precision, recall, and F-measure of our lane mark extraction method from the image are 0.965, 0.963, and 0.964, respectively, and the precision, recall, and F-measure of our lane mark extraction method from the point cloud are 0.981, 0.974, and 0.977, respectively. The performance of the proposed method is quite acceptable. In fact, from Table 2.2 and Figure 2.14(a), it can be seen that the pixels were incorrectly extracted only because of faded lane marks.

Table 2.2 Quantitative evaluation results.

	Precision	Recall	F-measure
Our result in **Fig. 2.14(a)**	0.965	0.963	0.964
ROI result in **Fig. 2.14(b)**	0.368	0.963	0.533
Our result in **Fig. 2.15(a)**	0.981	0.974	0.977
ROI result in **Fig. 2.15(b)**	0.690	0.974	0.808

2.3.3 Result of 3D modeling of lane marks

Figure 2.15 provides a comparison of 3D lane mark modeling results with a longer road area. The length of the road is approximately 60 m. Figure 2.15(a) presents the result of 3D modeling the lane marks by the proposed method, in which points are grouped into 6 lane marks and each lane mark is indicated by a different color. This result indicates

(a) The result of 3D lane mark modeling by the proposed method

(b) The result of 3D lane mark modeling by the ROI method

Fig. 2.15 Comparison of 3D lane mark modeling

that the proposed method can guarantee a precise lane mark location extraction. Figure 2.15(b) shows the result of 3D lane mark modeling using the ROI method. The input images used are obtained through the same ROI filter as in Figure 2.14(b). Objects near the white line are also extracted as false-positive points. For instance, the guard rail points are extracted and colored brown.

2.4 Conclusions

We propose a novel approach to create a 3D model of lane marks that combines information from color images and point cloud data. In line-based region growing algorithm, the precise boundary points are extracted to define the road surface region. We specialized the scan-line algorithm to the input dataset. The experiment results show that the proposed method can obtain a precise 3D model of the lane marks. Our approach can generate better results if curbstones are found on both sides of a road, but in some cases,

the curbstone does not exist.

CHAPTER 3

Precise Road Trajectory Estimation from Mobile Mapping Data

3.1 Introduction

The fast-growing market of advanced driver-assistance systems has created huge interest for its further improvement, particularly their reliability and safety. Various evaluation approaches will be necessary to analyze the ADAS performance, to ensure the reliability and safety. If we are able to create a highly precise road trajectory, the analysis on reliability can be more convictive. To produce precise road trajectory, all important clues such as mobile mapping data, road surface region extraction, lane mark detection and trajectory estimation should be utilized.

Nowadays, many mobile mapping systems were developed to provide dense point cloud data, images and road geometric information. MMS can quickly and accurately acquire geometry data of the road and its surroundings. A review of a recently available MMS and surveying technologies can be found in [1].

Some papers have present road surface region extraction and lane mark detection from point cloud with the help of intensity value of points. A framework is presented in [15] that includes road point selection and lane mark extraction. First, they selected the road points by using a thresholding algorithm. Next, the lane marks were extracted using a template matching method with the help of an intensity image of the point cloud data. The image-based methods are efficient when the road surface region is accurately-defined.

But it is difficult to extract precise 3D lane mark points using a thresholding filter on the intensity value. Because intensity information is highly sensitive to the distance between measured point and laser scanner.

In addition to the correction and analysis of ADAS, the most commonly used solutions are the global navigation satellite system (GNSS) and iterative closest point (ICP) algorithm. In a complex urban environment, the loss of GNSS signals should be considered. And point clouds can never align perfectly using ICP algorithm. [22] used a strategy based on the trajectory loop information and ICP registration.

Thus, to overcome the aforementioned limitations, in this research, we propose a novel stepwise method, which generates the 3D road trajectory step by step. We have adopted the line-based region growing method described in [35] to extract the road surface region, which can detect boundary points precisely. To improve computational efficiency, we use a scan-line algorithm to minimize the cost of lane marks extraction [6]. A morphological closing operation is used to refine the lane marks. Next, using the length and angle information of points, we produce a two-dimensional (2D) representation of the 3D points. Points are represented in a length-angle space in order to estimate the points between broken white lines. Finally, we calculate the 3D road trajectory points by the point sequences which consist of midpoints of white lines.

In sum, the main contribution of this paper is as follows:

1. We propose a novel stepwise approach which performs precise road trajectory estimation by incorporating image and mobile mapping data.
2. We now have a 2D representation of 3D points. Reducing the dimensions of data to 2D allow us to have a easy 2D solution for missing point estimation.

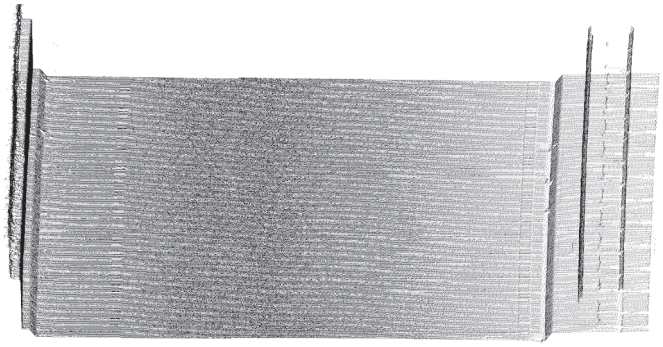
The remainder of this chapter is organized as follows. Section 3.2 presents the proposed approach in detail. Experiments are given in Section 3.3 and Section 3.4 concludes this study.

3.2 Method

3.2.1 Overview

The point cloud data and images used here are captured using a 3D laser scanner and CCD camera. The benefit of a 3D laser scanner is that the vehicle position is known and can be used for information on the road's location and orientation.

The whole dataset comprised of approximately 91.25 million points. The length of the road is approximately 2,900 meters. To cope with the large size of raw point data, the



(a) Point cloud data collected by laser scanner

(b) Color image collected by camera

Fig. 3.1 Dataset obtained by MMS.

collected data is automatically divided into volumes. Each volume has around 300,000 points that collectively represent about 8 m of the road and its surroundings.

Each point not only has a 3D coordinate but also has a laser irradiation angle and GPS time. This information can be used to structurize the points. We use the laser irradiation angle in order to separate the point cloud into scanlines. Moreover, we order points in a scanline and find the neighborhood elements by laser irradiation angle information. The point cloud data and color image used in this research are illustrated in Figure 3.1.

Our method takes a sequence of mobile mapping data and image as input. The GPS time stamp is used to find an image and its corresponding point cloud that were captured during a certain time interval. For each volume of raw point cloud data, we find an image to which all 3D points can be projected to this image. The method mainly includes the following three steps:

1. Road surface region extraction: following the line-based region growing approach, road surface regions are extracted from point cloud data. The endpoints of each

scanline from the extracted point cloud explicitly represents the edge points between the road and curb. We then project the endpoints from point cloud onto image. The converted endpoints are considered as the input of our specialized scan-line algorithm. The road surface region is then extracted from image with the scan-line algorithm.

2. Lane mark extraction: a thresholding binarization method is used to detect the lane marks from the extracted road surface region image. In spite of the result from thresholding, the extracted lane marks still contain noise and are incomplete. We apply a morphological operation to remove noise and complete their shapes.
3. Road trajectory estimation: the detected 2D lane mark points are converted to 3D coordinates. The points between broken white lines are estimated using linear interpolation method applied to a length-angle space. The road trajectories are represented by the point sequences which consist of midpoints of white lines.

Figure 3.2 illustrates the complete procedures used in this study. To prevent misunderstandings, Figure 3.3 shows the definitions of lines on the road.

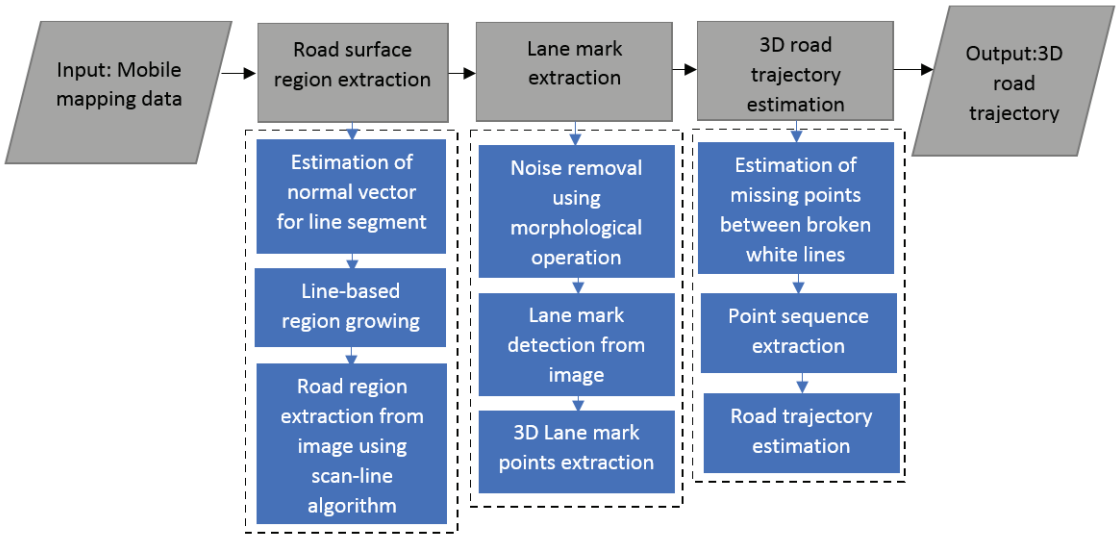


Fig. 3.2 Complete workflow of the proposed method.

3.2.2 Road Surface Region Extraction

Line-based region growing

The point clouds used here are collected by the Z+F IMAGER 5010 laser scanning system. Each point has a 3D coordinate and some additional information, including laser irradiation, the GPS time, and distance between the source point of the laser and a measured point. We assume a geometric discontinuity at the boundary of the road. In that case, the region growing process can stop exactly at a boundary point.

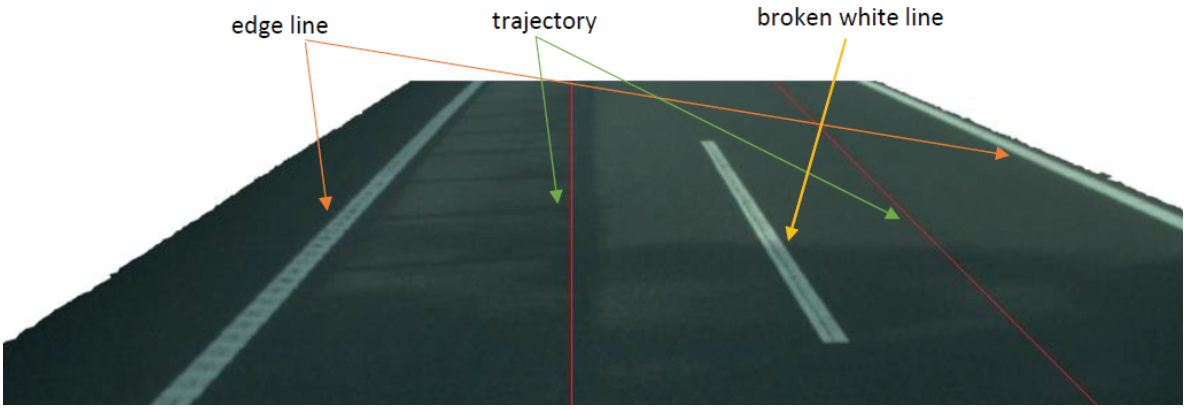


Fig. 3.3 Definitions of lines.

To find the location of the boundary points precisely, we use the line-based region growing method used in [35] to extract the road surface region from the point cloud. Following this method, the input to our algorithm is a set of line segments. We first create line segments from a point sequence using the angle of laser irradiation. We then use the line segments as processing elements for the road surface region extraction.

For searching of neighborhood line segments, we use the laser irradiation angle associated directly with sampled points. If two points on two consecutive scanning lines have a similar laser irradiation angle, these points are considered to be located near each other.

In the region growing approach, normal vector estimation is a crucial step. The difference between the angles of the normal vectors is used to determine whether a neighborhood should be added to the region. Least-squares fitting for neighbors is often used for normal vector estimation. However, such a method is unable to derive a precise estimation of a normal vector for our purposes. Thus, we adopt the local best-fit plane of the neighboring line segments to estimate the normal vector of a line segment. The local best-fit plane is defined as the plane that passes through the seed line segment and includes the most neighboring line segments. The normal vector in the local best-fit plane is used as the normal vector of the line segment. Figure 3.4 shows an example of the line segments and normal vector estimation.

After estimating the normal vector for the line segments by using the local best-fit plane, we calculate the difference between the angles of the normal vectors to determine if a neighboring line segment should be added to the region. The region growing approach starts with a selected seed. In our case, we select the seed segment with the largest degree of fitting from the input line segment that has not yet been assigned to any region.

Figure 3.5 is an example of extracted road surface region by line-based region growing. We paint the road surface points and non-road points in red and black, respectively. We

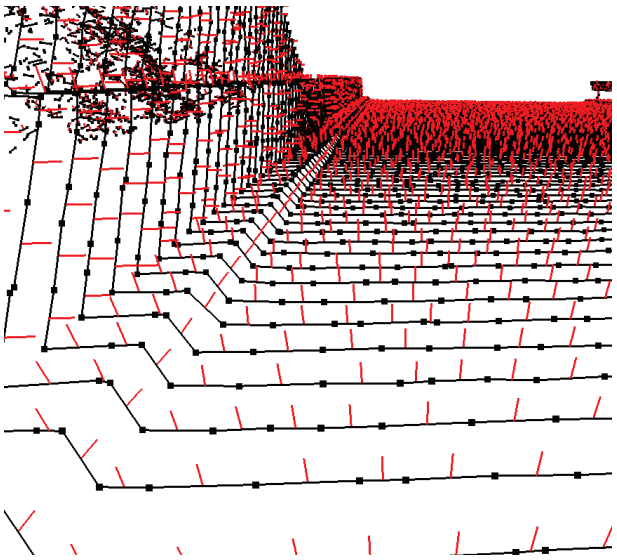


Fig. 3.4 Example of the line segments and normal vector estimation.

can see that the line-based region growing method precisely extracts the road surface region.

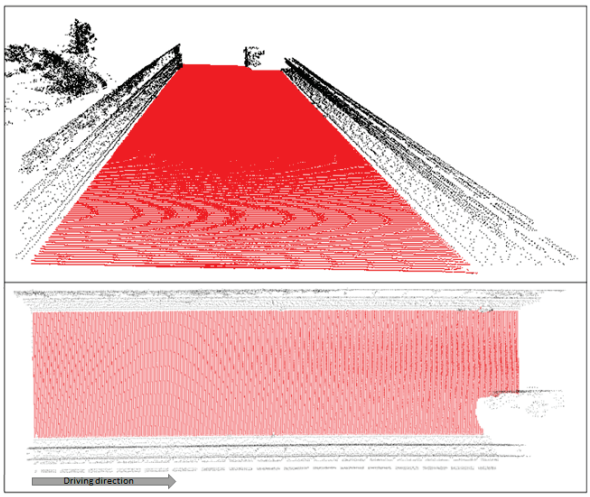


Fig. 3.5 The result of extracted road surface region (Upper: perspective view; bottom: top view).

Road surface region extraction from image

Given the extracted 3D road surface region, we need to project 3D points into 2D in order to extract the road region from image. Although many mature image processing methods can be used for the road surface region detection, the number of points that need to be projected onto the image is the most important issue in choosing the methodology.

The extracted 3D road surface region can provide accurate endpoints. Such an information is used as the constraints for representing the boundary edge in the process of road surface

region extraction from the image. Then, to extract the road region from the image, a specialized scan-line algorithm is used to reduce the execution time [6]. Because we only need to project the endpoints to the image by using the scan-line algorithm.

Scan-line algorithm works by intersecting the scan-line with polygon edges and filling the polygon between pairs of intersections. Most scan-line algorithms are designed for 3D and 2D image rendering. In our case, each pair of the road surface region boundary points can be seen as two end points of a segment of the polygon edge. Therefore, we consider those segments as the edges of the road surface region.

All scan-lines in this algorithm are horizontal. Because of the setting of MMS, the line segments in the point cloud obtained by MMS and the scan-lines are not parallel. Hence, we consider the projected road surface region to be a trapezoid shape, and then apply a scan-line based approach for precise road surface region extraction from the projected 2D image.

3.2.3 Lane Marks Detection

Binarization and Noise Removal

Once the road surface region have been extracted, we find the lane mark points by a binarization method. Because the extracted road surface region images used here are taken along the road, we can assume that the lane marks have the highest intensity pixel value. Therefore, a binarization method is carried out to obtain the brighter regions. We generate a binary image of the extracted road surface region to find the lane mark locations. The thresholds are manually selected because the brightness of the roads are influenced by environmental lighting conditions.

In some circumstances, the extracted lane marks have missing parts. For instance, the road is old with faded lane marks or the lane marks are on a damaged road area. Therefore, a noise removal method is needed here. We use a morphological closing operator [37], for fill in the missing parts on the white lines. It is defined simply as a dilation followed by an erosion. The morphological operations probe the lane marks with a shape called a structuring element. A structuring element is a small binary image, that is, a small matrix of pixels, each with a value of zero or one. The structuring element is related to the size, origin, and shape. Shape examples include square, cross, and diamond. A common practice is to use a structuring matrix of odd size to make it to be symmetric around its origin. Here, we use an elliptical structuring element with size 5 to dilate and erode the lane marks. We employ a dilation operation to remove noises and fill in the holes in the lane marks. Subsequently, an erosion operation is used to shrink the lane marks. An example of an extracted road surface region is shown in Figure 3.6(a). Figure

(a) An example of road surface region extraction from image

(b) Binarization result of extracted road surface region

(c) The lane marks refined by closing operation

Fig. 3.6 Result of lane mark extraction from image.

3.6(b) shows the result of the binary image of the extracted lane marks. As shown in Figure 3.6(c), the holes and other noises are removed after performing the morphological closing operation.

Lane Mark Points Extraction

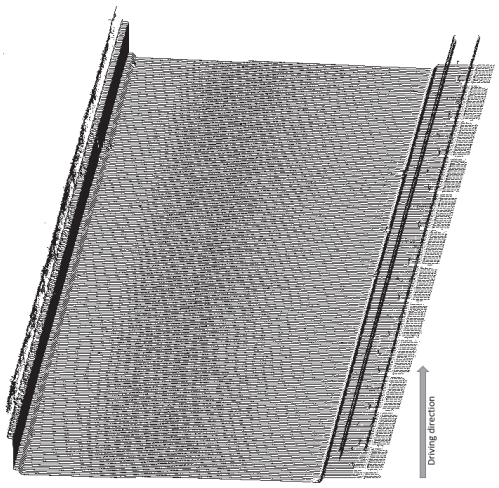
After lane mark detection from the image, 3D lane mark model in our approach are represented as 3D point clouds. Here, we consider a 3D lane mark model is a set of 3D points that lies in a 2D lane mark region when projected onto the image. The lane mark points are extracted by checking their projection's location.

3.2.4 3D Road Trajectory Estimation

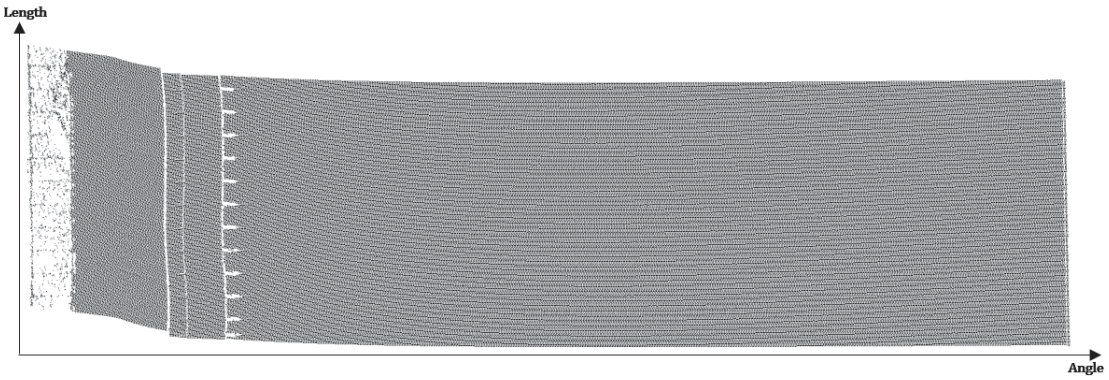
Considering that the road trajectories are the sequences consists of lane centerline points. Road trajectories are estimated in two steps. First, we extract sequences of points that indicates approximate midpoints of edge lines and broken white line. Then, to find the trajectory points, we calculate the nearest point of centroid of each scanline between the point sequences.

Point Sequence Estimation

In a complex urban road environment, we cannot guarantee that a long section of road is ideal straight. It is difficult to estimate the points between broken white lines in 3D. Since our measured points has not only three-dimensional coordinates but also the laser irradiation angles, we can convert the mobile mapping data into a 2D coordinate system. By employing angle information, a length-angle space is used to describe the data alternatively. This allows us to apply mature two-dimensional algorithm to obtain the missing points. Figure 3.7 shows an example of the length-angle space.



(a) A sample data of 3D point cloud



(b) Result of length-angle space

Fig. 3.7 An example of the length-angle space.

Hence, we calculate the midpoints of edge lines and broken white line using the distance between edge points of each scanline. The next step is to find the missing points between broken white lines. In this step, we establish the length-angle space, and then for each broken white line, we find a beginning point and an endpoint. Because the measured points can be affected by various noise such as holes or defects, therefore, in each broken white line, a least-square fitting algorithm is performed to decide if the beginning point

and endpoint are correct or false detection. For each broken white line, we initialize the least-square fitting algorithm with the midpoints, and then we compute the nearest points to the fitting line on the first and last scanline. Figure 3.8 shows the result of finding beginning point and endpoint.

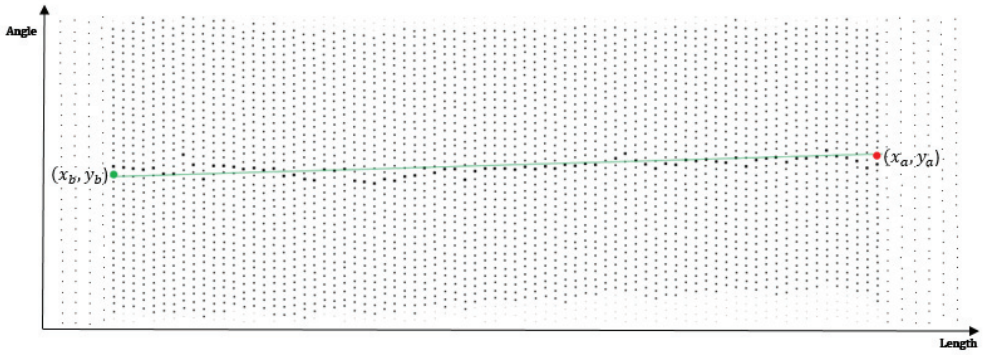


Fig. 3.8 Result of finding beginning point and endpoint. Beginning point and endpoint are painted in green and red, respectively. Green line is the fitted line.

The previous step gives us a beginning point (x_b, y_b) and an endpoint (x_a, y_a). For every two adjacent broken white line, we take a straight line between beginning point and endpoint, and run the linear interpolation on the line. We solve for y_n as follows:

$$y_n = y_a + (x_n - x_a)\frac{y_b - y_a}{x_b - x_a} \tag{3.1}$$

where (x_n, y_n) is the interpolated point. The result of missing point estimation is displayed in both 2D and 3D space in Figure 3.9.

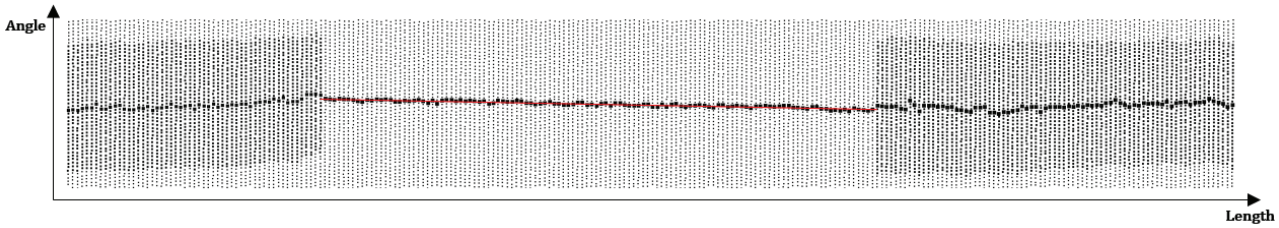
Road Trajectory Estimation

Having solved the problem of missing points estimation as explained above, we select the nearest point of centroid of each scanline between the point sequences as trajectory point.

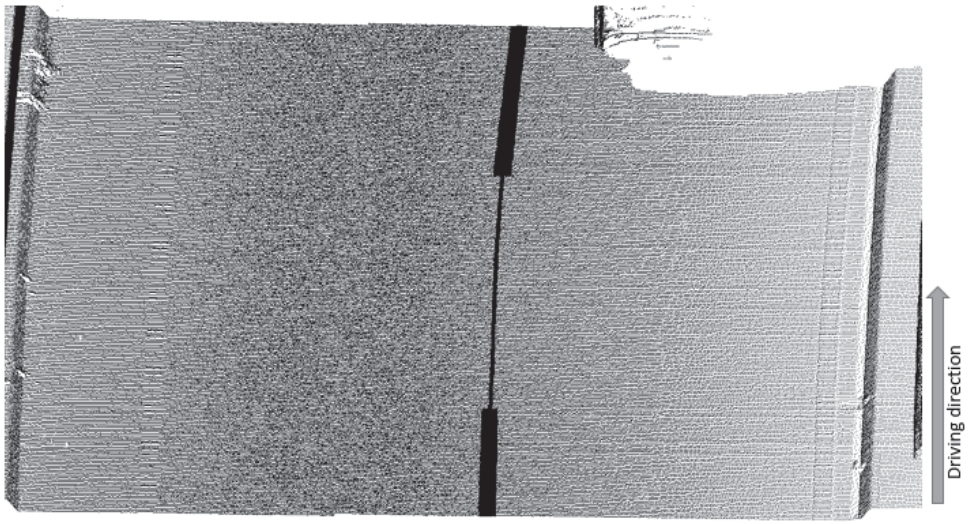
In this paper, the input point cloud used by this experiment consists of approximately 33 million points. The length of the road is approximately 1100 meters. Figure 3.10 shows the road trajectories which is created from the one kilometer long point cloud by the proposed method. Three magnified images shows close-up views of each road segment with both lane mark points and trajectory points.

3.3 Evaluation

The 3D modeling of road trajectories provides a reliable way to analyze the usefulness of ADAS. The accuracy of the trajectory point sequences estimation is the most concerned in the procedure. However, the road must have continuous curvature profile in the real world.



(a) Result of linear interpolation in length-angle space



(b) Result of linear interpolation shown in 3D

Fig. 3.9 Result of linear interpolation.

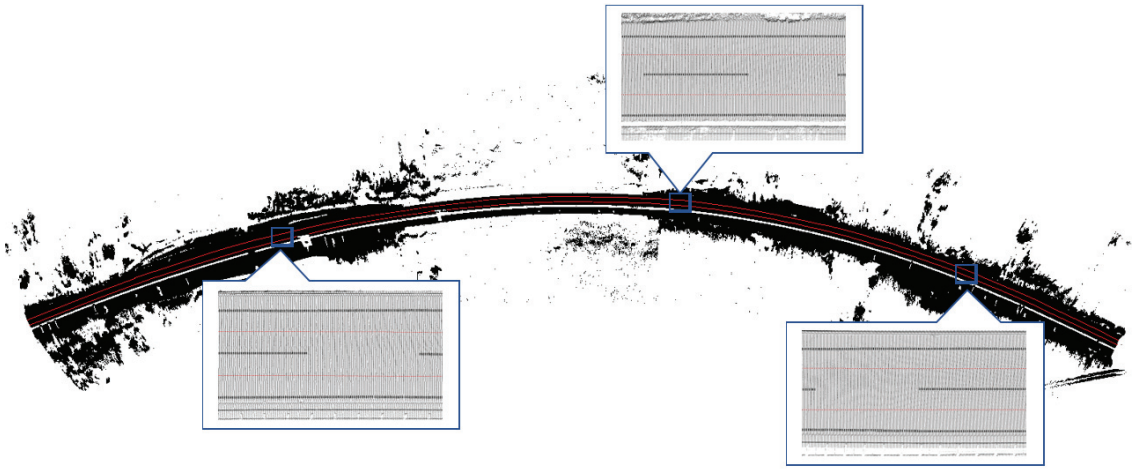


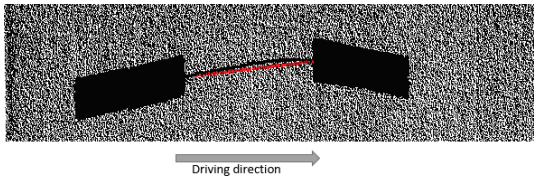
Fig. 3.10 An example of 3D road trajectory from a large-scale point cloud. Large black points are lane mark points and red points are trajectory points.

This ensures everyone on the road has a smooth and safe driving. The curvature at each point on the road trajectory are compared between our result and the reference. Here, a segment of the road which contains two broken white lines is used for the comparison. Each

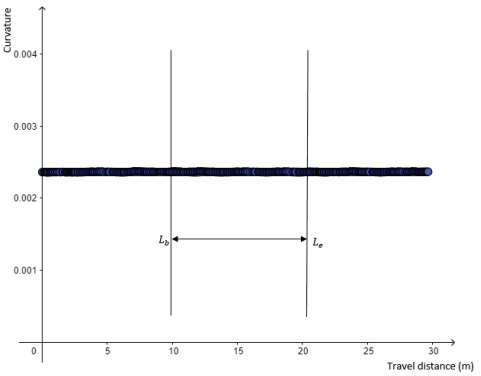
broken white line contains approximately 100 points. We calculate the curvature at each point and its 30 nearest neighbours using a least-square method. The computational time for the whole processing workflow was around 20 minutes on a PC with Intel Core i5-4690K at 3.5 GHz and 16 GB RAM. Since there is no ground truth data and design curvature of the road trajectory, we build a reference dataset for the comparison and evaluation. In the reference, the points between adjacent broken white lines are calculated by connect the beginning point and endpoint with a straight line in 3D without using the length-angle space. Trajectory points in the reference are then calculated follow the same procedure as described in Section 2.4.2. This evaluation is shown in Figure 3.11. In Figure 3.11(a), an example of input data pair which contains two broken white lines is presented. Based on the comparison, as shown in Figure 3.11(b), it is observed that a continuous-curvature point sequence was obtained by the proposed method, which is approaching to a straight line. Figure 3.11(c) shows the changes of curvature of the reference data. The curvature of points between broken white line decreases significantly. To determine the validity of our work, we applied our method for acquiring the curvature of a 1100 meter long road data. As it can be seen in Figure 3.12(a), the curvature of points obtained by our method are approaching an approximately constant value. On the contrary, Figure 3.12(b) indicates curvature changes along 1100m long reference road data. The comparison of two methods can also confirm the importance of the proposed method.

3.4 Conclusions

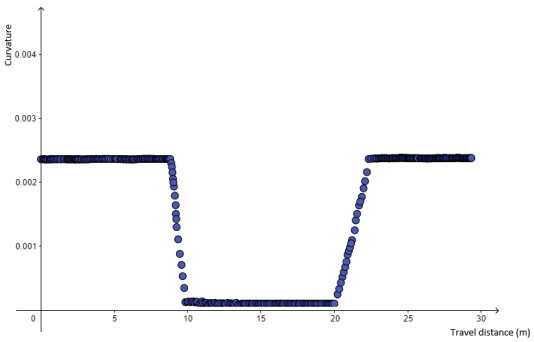
In this paper, we proposed a novel, efficient and precise approach for road trajectory estimation. The road surface region and lane mark extraction offers precise 3D model of lane marks. Due to the difficulty in finding missing points between broken white lines, we introduced a length-angle space to make 3D problem 2D. As shown, incorporating all the available data, the proposed method can produce precise result in each step. The produced road trajectories contribute to comparison with the vehicle trajectories, and thus analyze the ADAS reliability. There are still some limitations to the proposed method. According to [6,35], the road surface extraction and the lane mark extraction can be considered as preprocessing of the point cloud data, which has always been time-consuming. This means that our workflow is not supposed to be run in real-time. Furthermore, if the road surface and lane mark were extracted during data preparation phase, then the computational time can be significantly improved. In future work, it will be focused on extending our method in order to support more complex environments and improve the real-time capability.



(a) Example of two broken white lines shown in 3D. Red points are the reference points

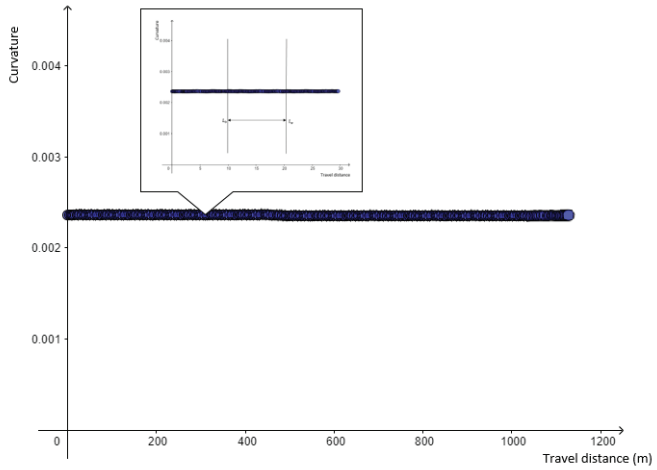


(b) The curvature of our result. (L_b and L_e are beginning point and endpoint, respectively)

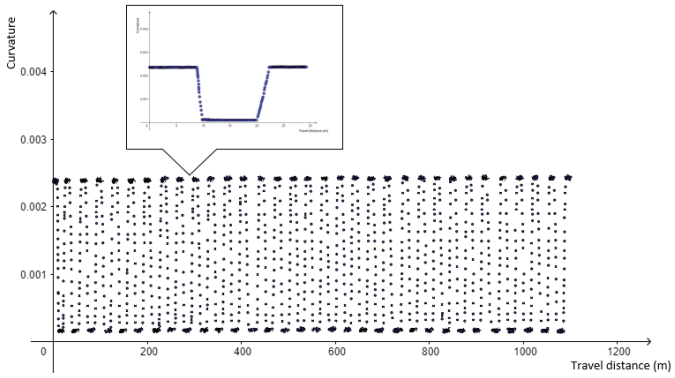


(c) The curvature of reference data

Fig. 3.11 Comparison of the results.



(a) The curvature result by applying our method



(b) The curvature result of reference data

Fig. 3.12 The curvature of a 1100m long road data.

CHAPTER 4

High-Resolution Representation for Mobile Mapping Data in Curved Regular Grid Model

4.1 Introduction

The modeling of road scenes has become an increasingly important topic in the academia and industry. While various techniques have been developed to handle different types of issues, it is recognized that efficiency and reliability are both key consideration in the assessment of systems. However, to help the vehicle development and evaluation, a high accuracy and high precision three-dimensional model of the road surface is necessary and very valuable. There are multiple methods of building detailed representations for the road model, and they can result in different models based on various purposes. For example, in [23] and [24], the authors presented approaches for grid-based road model estimation for advanced driver assistance system. Their measurements from sensors are transformed into a grid-based road model and a geometrical description is extracted out of this model by the use of a path-planning based method. In [40], modeling of the road with geometric parameter representation has been proposed, which contains three parts of road: straight line, circular arc and clothoid. The relationship between trajectory curvature and velocity is established, then, simulation has been carried out to verify the road model and proved that the model can be used to control the four-wheeled robot. For other uses, some authors proposed methods using the elevation information for building road model [25]-[38]. It has been proven that the elevation-based methods are suitable

for detection techniques.

However, precise 3D road surface measurements and efficient 3D road data representation are two essential requirements for building high precision 3D road model. To fulfill the requirement for precise road surface measurements, we use point cloud as input data. Point cloud data is a set of measured points in three-dimensional space. Point cloud is one of the most widely used data type in three-dimensional image processing. Point clouds are already having a huge impact on different industries, especially on the road-related projects, e.g., road survey, road modeling and autonomous vehicle. Mobile mapping system is one of the most widely used surveying device for capture large-scale point cloud and digital images. [1] provided a recently available MMS and surveying technologies review. Some of the newly developed and presented systems are [41]-[44]. The systems produce large-scale 3D point clouds and very high precision geometric measurements. The produced point clouds are used for many road related research tasks, including for missing road point regions detection [6,7], for road damage information detection [8-10], for road segmentation and recognition [11-17], etc.

Since the geometry relationships between road segments are normally assumed as straight line segment and curved arc segment. There are many different approaches that are available, dealing with various situations, examples can be found in [17,46-48]. In geometric road design, we demand G2 continuity, which means that not only the tangent vectors between different road segments are lying along the same direction, but also having the same curvature at the joint point. The commonly used algorithms for straight line and circular arc detection are the random sample consensus (RANSAC) and the Hough transform. A comprehensive overview of recent research in RANSAC-based estimation method is depicted in [49]. Therefore, most of the strategies focus on the transition curves. In [29] and [30], the authors demonstrate a workflow for representing the point cloud data in curved regular grid model. The input laser scanned point clouds and geometry description of the road both need strong manual intervention in the preparation phase. The selected pilot road has a near perpendicular segment, and the elevation values are calculated by a fixed-radius near neighbours search algorithm from the input point cloud directly. The mean elevation values of points inside the circle are stored in the CRG cells. This also cause the problem that the resolution of generated CRG model is limited by the density of input point cloud. However, the performance and accuracy of the two key steps, road segmentation and elevation estimation, can be further enhanced. In our previous work [6][50], we have presented a workflow which can produce high-precision three-dimensional point cloud model of road surface region and trajectory points. In this paper, we extend the method based on our previous results for representing the mobile mapping data in CRG model efficiently.

In this work, the goal is to build a road model that contains geographic information of road surface and use elevation information to show the shape of road surface. We firstly apply a robust and effective method that can divide road into three road segment categories: straight line, circular arc and clothoid curve. For the straight line and circular arc solution, we utilize the RANSAC algorithm. Then, we adopt a G2 interpolation method [51] to estimate the transitions between road segments, that is the clothoid curves. To create CRG model file, a regular grid which provides elevation values is needed. To effectively and accurately accomplish this, we use a two-step method to generate the regular grid. In the first step, for finding regular grid, a non-regular grid is initially created from the input point cloud and trajectory points. In the next step, we estimate the regular grid from the non-regular grid by applying the bilinear interpolation method. Finally, the presented process is applied to our real-world point cloud data which is collected from the Japanese highway network.

In sum, the main contributions of this work are as follows. First of all, in the data preparation phase, the whole process operates without any human intervention. To enhance the accuracy of elevation estimation, with the two-steps method, the bilinear interpolation ensures the elevation values are precisely computed. It can also provide a guarantee of high-resolution, which the grid resolution can be selected by the user. Secondly, to improve the time and space efficiency, comparing with the near neighbours search approach used in [29], the two-step method can decrease the storage and query execution time. Furthermore, our previous work results provides precise road surface region point cloud and trajectory points. Precision data can be used to create a more accurate elevation regular grid for the purpose of building the CRG model. Finally, with the help of OpenCRG, we can represent three-dimensional (3D) road data in CRG models.

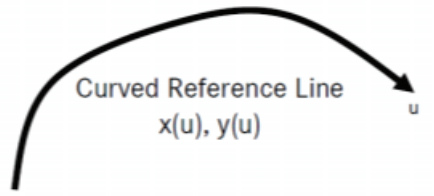
4.1.1 Overview of OpenCRG

In 2005, a project called OpenDRIVE [31] was started by a team of driving simulation experts from Daimler AG and VIRES Simulationstechnologie GmbH. This project aimed to standardize the road description to facilitate the data exchange between various driving simulators. This is the first member of OpenSolutions family. After the debut of the OpenDRIVE in 2006, other big companies joined in, e.g. BMW, Audi AG, Porsche AG and Volkswagen Group, thus, OpenDRIVE is now being managed by an international board. OpenDRIVE provides a road evaluation library which can make the data exchange between different serves and applications easier. It is also available for vehicle dynamics, traffic simulation and sensor simulation via the library.

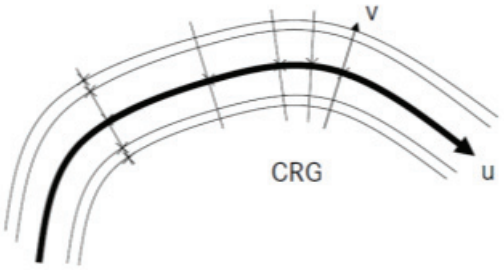
As a complementary, the OpenCRG project established in October 2008 [32]. CRG stands for curved regular grid. Its objective is to provide open file formats and tools for the

representation of high precision 3D road surfaces. The predecessor of OpenCRG is a format called CRG which has been used internally for several years by Daimler AG. OpenCRG is designed to represent road surface in very high-resolution, so that the CRG files can be used for tire simulation, vibration simulation and driving simulation.

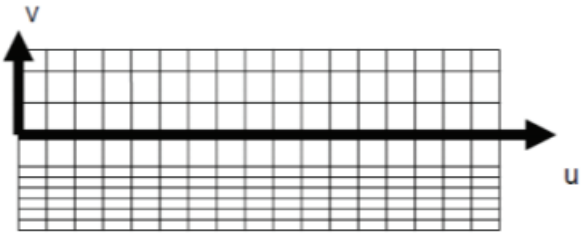
In order to present the road data in CRG model, the road parameters must be defined, e.g., start position, end position, road width, slope, heading angle and elevation. Start position and end position are actually the start point and end point of a road segment. As shown in Figure 4.1, a curved regular grid represents road elevation data close to a road center line. A CRG model consists of two major parts, reference line and regular elevation grid. The reference line is defined by a start position, an end position and consecutive heading angles. The u-axis lies on the reference line and the v-axis is orthogonal to the reference line. The regular elevation grid is a special form of regular grid which is locally orthogonal. Columns are longitudinal cuts that are parallel to the reference line, and rows are lateral cuts that are orthogonal to the reference line. The grid nodes contains elevation information of road surface. A toolbox of MATLAB and C-API had been developed for the handling, evaluation and generation of CRG data.



(a) Curved reference line.



(b) Lateral cuts are orthogonal to the reference line.



(c) Regular elevation grid.

Fig. 4.1 The basic idea of CRG.

In addition, the OpenSolutions family has expanded with a new member OpenSCENARIO [33]. It was formally presented in 2016 and is still in development stage. The purpose of

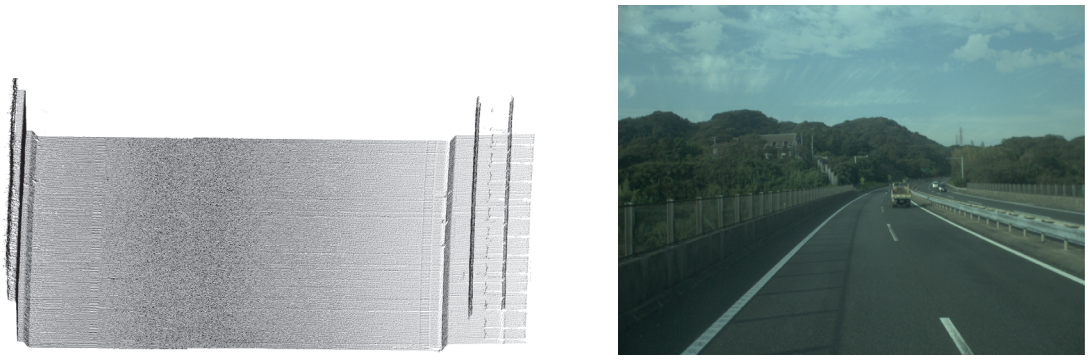
the project is to establish a standard for dynamic content in vehicle simulations, such like traffic model, driver model, infrastructure event model, etc.

The paper is structured as follows. In section 2, we give a brief review of our previous work, the proposed method is then described in detail. Experiment results are illustrated in Section 3 and Section 4 concludes this paper.

4.2 Method

4.2.1 Input data preparation

The point cloud data and images used in our work are captured by using 3D laser line scanner and CCD camera. It has the benefit of a 3D laser line scanner is that the vehicle position is known and can be used for information on the roads location and orientation. Each point not only has three dimensional coordinate but also has laser irradiation angle and GPS time. These information can be used to structurize the points. From these information, we use laser irradiation angle in order to separate point cloud into scan lines. Moreover, we order points in a scan line and find neighborhood elements by laser irradiation angle information. Point cloud data and color image which are used in our research are illustrated in Figure 4.2.



(a) Point cloud data collected by laser scanner (b) Color image collected by camera

Fig. 4.2 Dataset obtained by mobile mapping system.

However, the interval of measured points along the direction of which the MMS travels depends on the rotating speed of the laser irradiation part and the speed of which the MMS travels. The rotation period of the laser irradiation part is much longer compared to the laser irradiation period. The measurement interval along the direction the MMS travels is often about a few hundreds of millimeters. Thus, the density of point distribution is greatly unbalanced with the direction (Figure 4.3).

This situation causes the problem to the method of creating regular grids of elevation

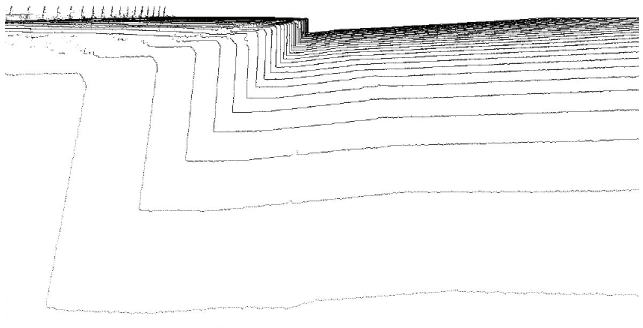
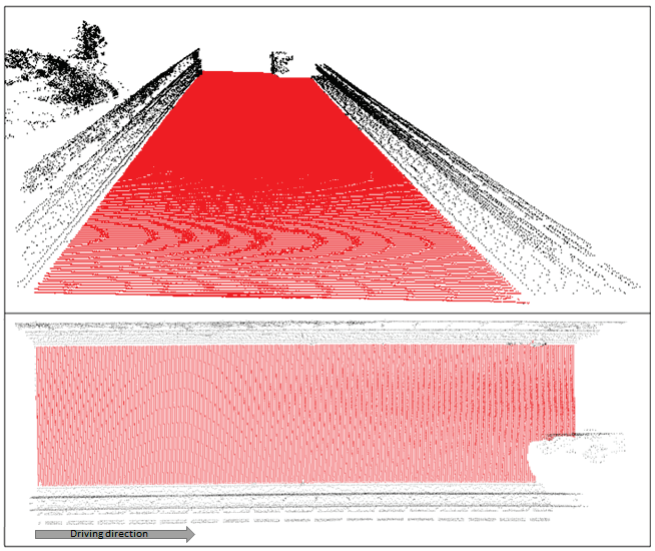


Fig. 4.3 The difference of point density according to the direction.

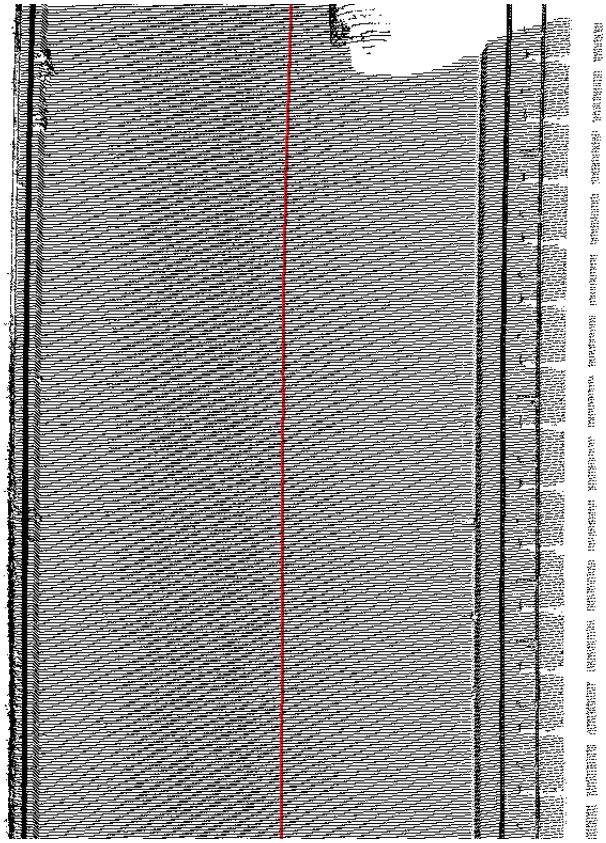
values onto raw point cloud data directly. In such a case, the manual process is always involved, besides, this kind of methods are very time-consuming. For example, in [52], the authors presented an algorithm of local gridding. The elevation values are computed based on local binning geometry. The accuracy and time complexity of such an algorithm is not suitable for creating the CRG model.

In [17] to find the location of the boundary points precisely, we use the line-based region growing method to extract the road surface region from the point cloud. Following this method, the input to our algorithm is a set of line segments. We first create line segments from a point sequence using the angle of laser irradiation. We then use the line segments as processing elements for the road surface region extraction. For searching of neighborhood line segments, we use the laser irradiation angle associated directly with sampled points. If two points on two consecutive scanning lines have a similar laser irradiation angle, these points are considered to be located near each other. Next, we extract lane marks and their midpoints [6]. The road surface points are projected onto color image to find precise lane mark region. Then, we perform an inverse projection to recover the 3D coordinates of the detected 2D lane mark points. For the missing points between broken white lines, we describe the three-dimensional points in a length-angle space to fill the gap. Considering that the road trajectories are the sequences consists of centerline points. Finally, we generate a 3D point sequence to represent the trajectory points [50], and hence we can use the trajectory of the road as the reference line in road surface modeling.

Figures 4.4(a) and 4.4(b) shows examples of the results in our previous work. Figure 4.4(a) is an example of extracted road surface region by line-based region growing. We paint the road surface points and non-road points in red and black, respectively. We can see that the line-based region growing method precisely extracts the road surface region. Figure 4.4(b) shows an example of 3D road trajectory points, which are colored in red.



(a) The result of extracted road surface region (Upper: perspective view; bottom: top view).



(b) The result of trajectory points shown in 3D (Top view)

Fig. 4.4 The results of our previous work.

4.2.2 Overview of method

In the first step, we divide the input trajectory point into three types, straight line, circular arc and clothoid. Naturally, the curvature of straight line is nearly zero, circular arc has constant curvature and the curvature of clothoid varies linearly along the arc. The straight and circular road segments are both extracted by the RANSAC algorihtm, which is not only able to give accurate detection result, but also maintain speed and stability. The straight lines are firstly extracted, then, circular road segments are extracted from the rest of the data. Points between straight line and circular segment should be fitted by a clothoid curve.

In CRG model, the left and right road width must be fixed. In the real world, we have an emergency lane on the side of road, which means the width is not ideally equal. In our case, we are analyzing a symmetric road which means that the width values are the same. Figure 4.5 illustrates definitions of left width, right width and emergency lane.

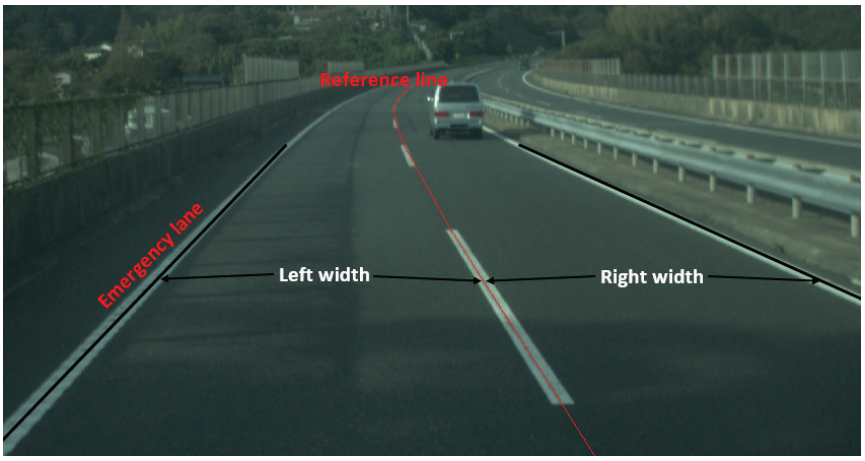


Fig. 4.5 A road image of left width, right width and emergency lane.

Elevation values are indicated by z-values of each point. The regular grid of elevation is the most important component of the CRG file. Thus, we create a non-regular grid of elevation for each road segment in order to acquire a regular grid of elevation due to varied distance between points and sparse scanlines. A bilinear interpolation method used here to compute the elevation of the regular grid point from surrounding non-regular grid points. Finally, we make the CRG file according to the format specification of OpenCRG model. The complete workflow used in this paper is illustrated in Figure 4.6.

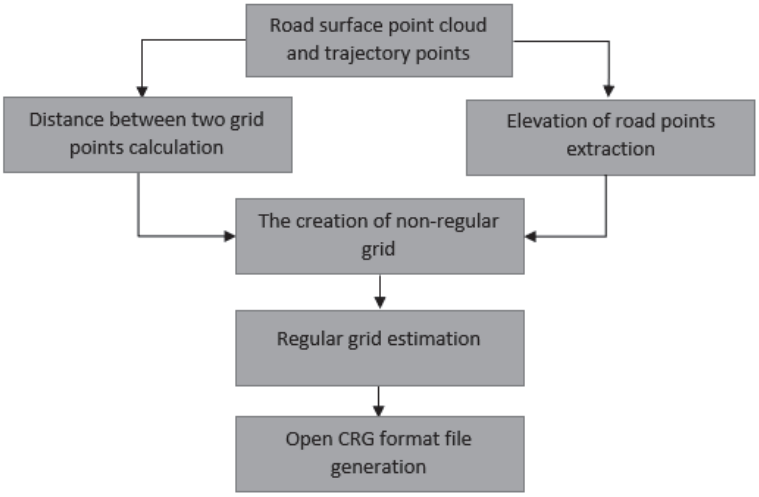


Fig. 4.6 Processing workflow.

4.2.3 Road segmentation process

To find straight lines and circular arcs, among many mature algorithms, the RANSAC algorithm is widely used because of its reliability and accuracy. Since the trajectory is a point sequence with continuous-curvature profile, we apply a common RANSAC strategy for straight line and circular arc detection. The x and y coordinates of trajectory points are used as input. A minimal subset of the trajectory points is chosen randomly and model parameters are estimated from this subset. The estimated model is then checked by the entire dataset and all data points are classified as inliers or outliers by calculating the residuals to the model. In each iteration, the algorithm perform the same actions until the best model is determined. The straight lines are first estimated and the circular arcs are detected from the remaining data points.

To dealing with the transitions between road segments, we use the G2 interpolation method in [48] to estimate the clothoid curve. Given a start point (x_0, y_0) and an end point (x_1, y_1) of a transition clothoid curve, the heading angle and curvature are also calculated. In general, it is not possible to estimate a transition curve with only one clothoid. Consider two clothoid segments that have to join with G2 continuity, an intermediate point $M = (x_M, y_M)$ that joins the two arcs with G2 hypothesis is introduced here. Assuming two segments are starting respectively from the start and the end point, matching at M. Given two points (x_0, y_0) and (x_1, y_1), two angles ν_0 and ν_1 and two curvatures κ_0 and κ_1. Let s_0 and s_1 be the lengths of the two matching arcs, and the curvature change rates are

κ_0' and κ_1', we can define two arcs by

$$s_0 = \alpha L, s_1 = (1-\alpha)L, \kappa_0' = \frac{A(\alpha, L)}{\alpha^2 L^2}, \kappa_1' = \frac{A(\alpha-1, L)}{(1-\alpha)^2 L^2} \tag{4.1}$$

where

$$\begin{aligned} A(\alpha, L) = \alpha^2 L \Delta\kappa + \alpha(2\Delta\nu - L(\kappa_0 + \kappa_1)), \\ \Delta\kappa = \kappa_1 - \kappa_0, \Delta\nu = \nu_1 - \nu_0 \end{aligned} \tag{4.2}$$

and α, L are the solution of the smaller nonlinear system:

$$\begin{aligned} \Delta x/L = \alpha X_0(A(\alpha, L), \alpha L\kappa_0, \nu_0) + \\ (1-\alpha)X_0(A(\alpha-1, L), (\alpha-1)L\kappa_1, \nu_1) \end{aligned} \tag{4.3}$$

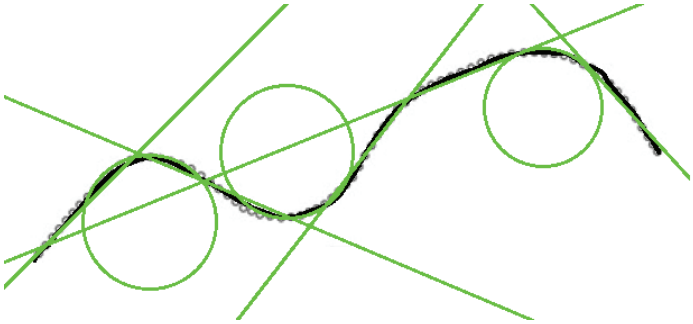
$$\begin{aligned} \Delta y/L = \alpha Y_0(A(\alpha, L), \alpha L\kappa_0, \nu_0) + \\ (1-\alpha)Y_0(A(\alpha-1, L), (\alpha-1)L\kappa_1, \nu_1) \end{aligned} \tag{4.4}$$

where $\Delta x = x_1 - x_0$ and $\Delta y = y_1 - y_0$.

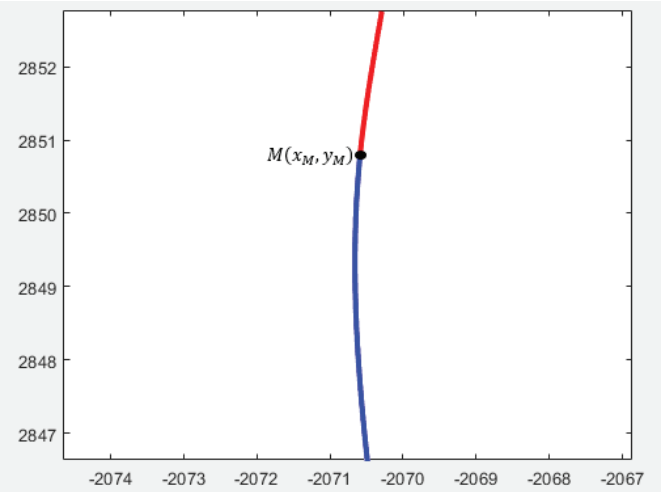
Figure 4.7 shows the examples of finding straight lines, circular arcs and clothoid curves. Figure 4.7(a) is an example of RANSAC algorithm result. The example of clothoid estimation is shown in Figure 4.7(b).

4.2.4 Creating non-regular grid

After dividing trajectory points into road segments by three categories, to create a non-regular grid for each road segment, we separate the input point cloud into scanlines using laser irradiation angle, and we also find the ordering for the points on each scanline. Then, we assign a corresponding grid point in the non-regular grid for each point in point cloud by the order of points on each scanline. Figure 4.8 shows the idea, p_a, p_b, p_c are trajectory points, horizontal and vertical distance values between measured points are adopted to non-regular grid points. For example, along both length and width axes, the distance between two grid points are calculated by two adjacent trajectory points and two neighbour points lying on the same scanline, respectively. In the other word, horizontal distance means the length along the vehicle heading direction, vertical distance means the width along the scanline. Based on the input road surface point cloud, we extract elevation by the z-value of each point.



(a) Straight line and circular arc detection



(b) Clothoid curve estimation

Fig. 4.7 Example of straight line, circular arc detection (top) and clothoid curve estimation (bottom).

4.2.5 Creating regular grid

According to the format specification of OpenCRG model [53], to create an OpenCRG file, a regular gird which contains elevation value of the points is needed. A bilinear interpolation is applied here to estimate a regular grid from the non-regular grid with a predefined resolution. In Figure 4.9 we illustrate the bilinear interpolation process.

For each interpolated point $p(x, y)$, we find four surrounding non-regular grid points. The elevation value h at an interpolated regular grid point is defined as

$$\begin{aligned} h = \frac{1}{(x_2 - x_1)(y_2 - y_1)}((x_2 - x)(y_2 - y)h_{11} + \\ (x - x_1)(y_2 - y)h_{21} + (x_2 - x)(y - y_1)h_{12} + \\ (x - x_1)(y - y_1)h_{22}) \end{aligned} \tag{4.5}$$

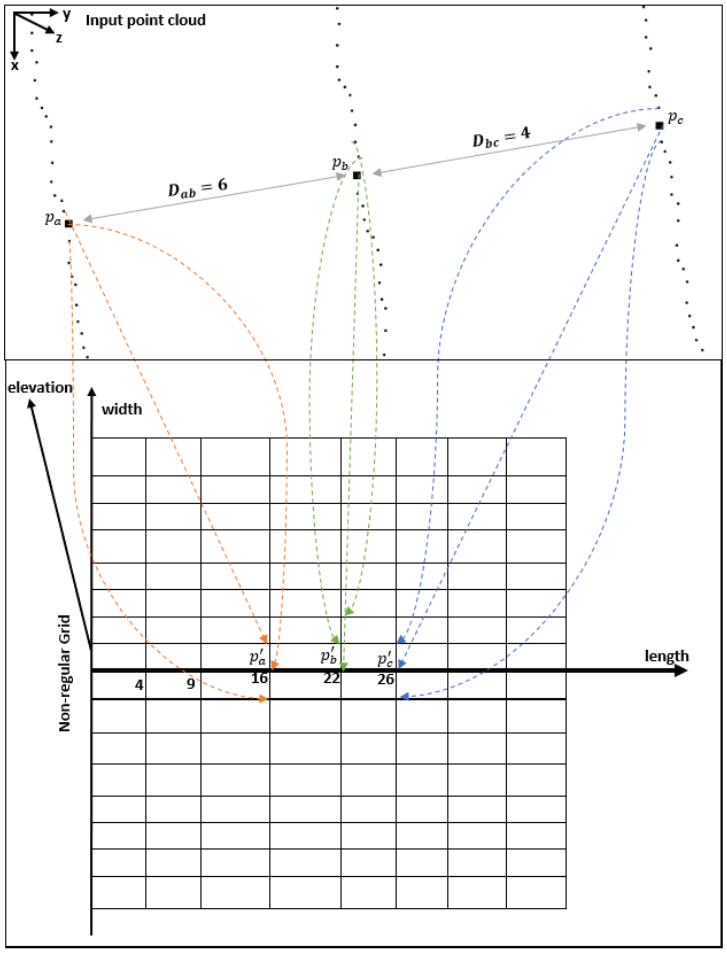


Fig. 4.8 Idea of finding corresponding non-regular grid points (distance value in the figure are the unit distances).

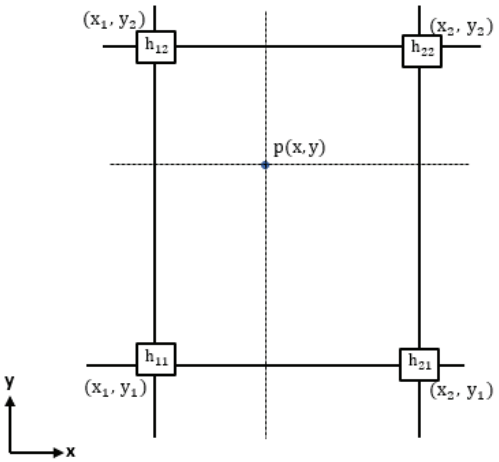


Fig. 4.9 Bilinear interpolation.

where h is the elevation at an interpolated point $p(x, y)$, $h_{11}, h_{12}, h_{21}, h_{22}$ are the elevation values at four surrounding points.

4.2.6 Building CRG model

Since the regular grid of elevation are created, we follow the specification of OpenCRG to make the file for each road segment. Start position, end position and consecutively heading angles allows us to build CRG model for the road data. Increment on reference line and spacing on left and right of reference line are both fixed by the user defined resolution.

4.3 Results

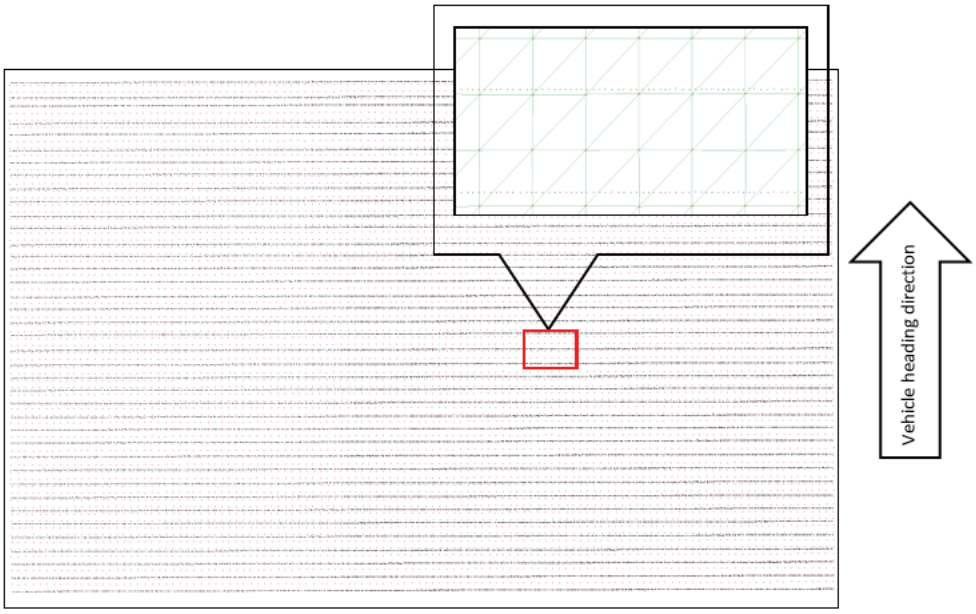
In this paper, the input point cloud measured by the MMS equipped with the Z+F IMAGER 5010 laser scanning system. The color images were taken by the same MMS during the data collection. The datasets are collected on the toll road in Yokohama, Japan. Considering the density of input point cloud and [54], in order to find the straight line and circular arcs, the minimum and maximum length of road segment are set to 20 m and 1000 m, respectively. The number of iterations of RANSAC is set to 2000 to ensure a reliable result.

4.3.1 Comparison of non-regular grid and regular grid

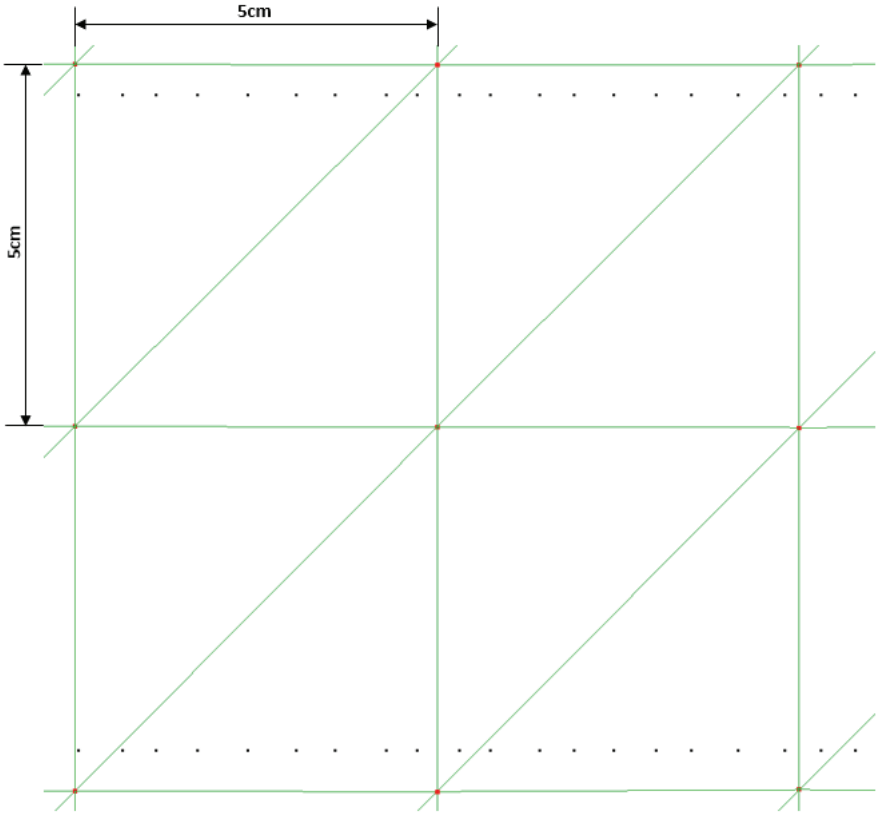
The proposed method create the non-regular grid first in order to find the regular grid. Considering that the expressway road is in good condition, a regular grid with a longitudinal and lateral resolution of 5 cm $\times$ 5 cm is estimated by our method. A magnified view of the sample of generated non-regular grid and regular grid are shown in Figure 4.10. The black points represents the non-regular grid points in the road segment. The red points represents the estimated regular grid points. As can be seen, the spacing between non-regular grid points is uneven, and regular grid points are accordingly estimated from surrounding points. The elevation values of each grid are estimated from 4 optimal surrounding measured points by bilinear interpolation. To better observe the result, we create a mesh for the regular grid points. In Figure 4.10(b) is a close-up view of the grid points, shows the relationship of two kinds of grid points. Consequently, the created regular grid can generate a uniform mesh, which means each grid cell has the same structure.

4.3.2 Result of CRG model

The CRG model are visualized by the OpenCRG MATLAB tools. The result consists of a reference line XY overview map, a road XYZ map and a road UVZ map. The CRG road XYZ map represents the road segment in curved XY grid with Z axis as elevation. The road UVZ map shows the road in uncurved UV grid with Z axis as elevation. A 5 cm resolution CRG model was built in this experiment. Figure 4.11 provides an example of



(a) Part of generated grid points.



(b) A close-up view of the grid points.

Fig. 4.10 Comparison of non-regular grid and regular grid. Black dots are the non-regular grid points; red points are the estimated regular grid points.

clothoid road segment. The length of this clothoid segment was 2.5 m and 2.94 m width on both side of road center line symmetrically. As we can observe clearly, the road has a slight height difference. The minimum height of the road is -4.04 m and the maximum height is -3.99 m. Here, the term height refers to the distance above (or below) mean sea level. The curvature changes is too small to see, this is due to the fact that the input data is obtained from a highway environment. Figure 4.12 illustrates a circular road segment CRG model. The length and width of sample circular point cloud data are 173.45 m and 5.88 m, respectively. The height value changes are shown intuitively. The minimum and maximum height are 4.12 m and 11.08 m, respectively.

In our case, the generated regular elevation grid is more dense than the input point cloud, it is difficult to evaluate the accuracy of resultant model by numerical analysis, because there is no ground truth data. Since the method finds optimal measured points for estimating elevation of each grid by bilinear interpolation, the elevation values are estimated with reasonable accuracy. We assume that the estimated elevation values by proposed method are the ground truth data. One way to evaluate the effectiveness and accuracy is calculate the root-mean-square error (RMSE) to measure the differences of elevation values between proposed method and a nearest-neighbour search (NNS) method. A subsampled grid is created from the non-regular grid by the distance between grid points. In our experiment, two subsets of points are obtained by assigning the distances of 0.05 m and 0.10 m. For each regular grid point P_i, we use the obtained elevation Z_i as the reference elevation. The elevation Z_{bi} of the point P_i is calculated by the proposed method, and the closest grid point is used to obtain the elevation value Z_{nn} by the NNS method. Then, we compare the elevation Z_{bi} and Z_{nn} with reference elevation Z_i and calculate the RMSE. The RMSE of residuals is defined by the follow formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Z_{ti} - Z_i)^2}{n}} \tag{4.6}$$

where n represents the grid point number of the generated regular grid and Z_{ti} is the elevation determined by proposed method and NNS method. The quantitative evaluation result is shown in Table 4.1.

Table 4.1 RMSE of residuals on proposed method and nearest-neighbour search method.

Subsampling spacing(m)	RMSE of our method(m)	RMSE of NNS method(m)
0.05	0.001062611	0.005046622
0.10	0.001433533	0.01131894

Table 4.1 shows that the RMSE of the 0.05 m spacing dataset for our method and NNS method are 0.001062611 m and 0.005046622 m, respectively. And the RMSE value for 0.10 m spacing dataset are 0.001433533 m and 0.01131894 m, respectively. We can observe that our method achieved better accuracy on both datasets. In addition, it can be seen that the accuracy of the proposed method can maintain a high-precision even when a sparse dataset is used as input.

The execution time for two example CRG models are 4.75 seconds and 38.91 seconds on a PC with Intel Core i5-9600k at 3.70 GHz and 16 GM RAM. The preprocessing time cost for extracting reference line by our previous presented method is less than 20 minutes. For comparison, as described in [29], a 259.55 m pilot site is used in their experiment. Their computational time for building the CRG model was 3 hours. Since there is no detailed explanation on the composition of computational time, we assume that the running time during the data preprocessing phase is also counted, e.g., manually finding road surface region, manually finding reference line, etc.

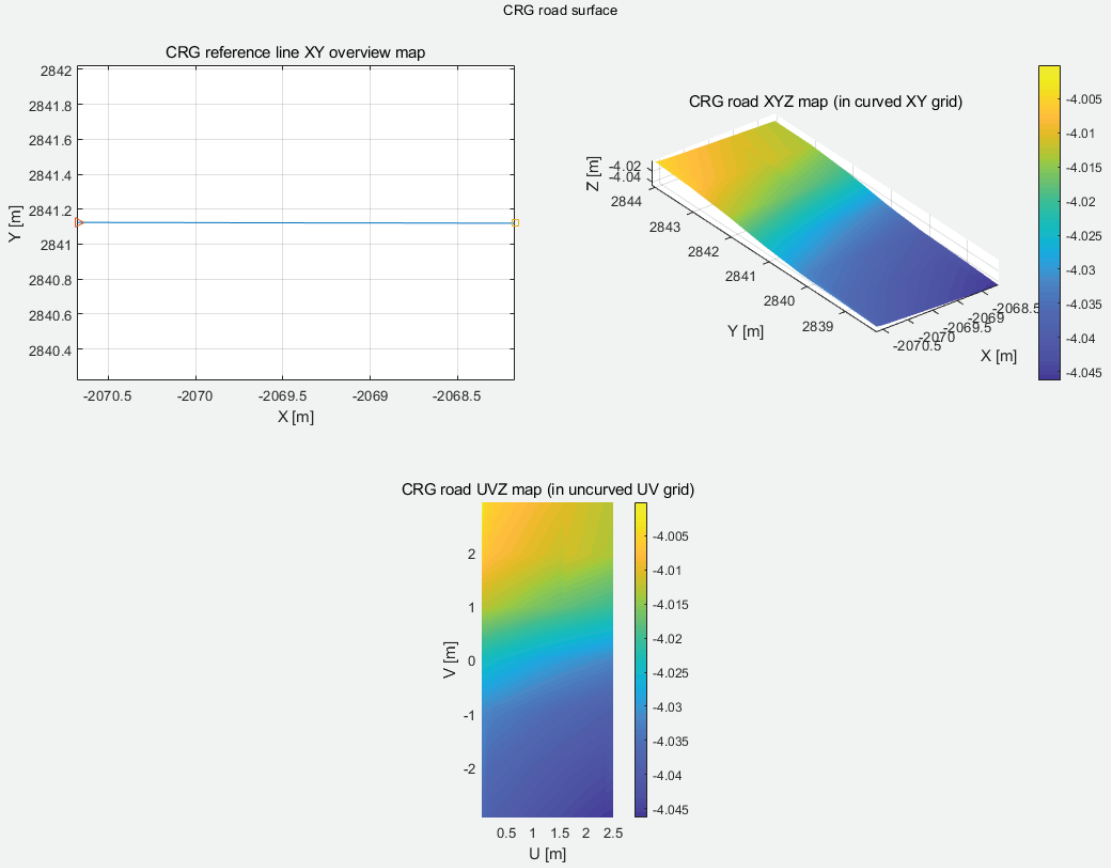
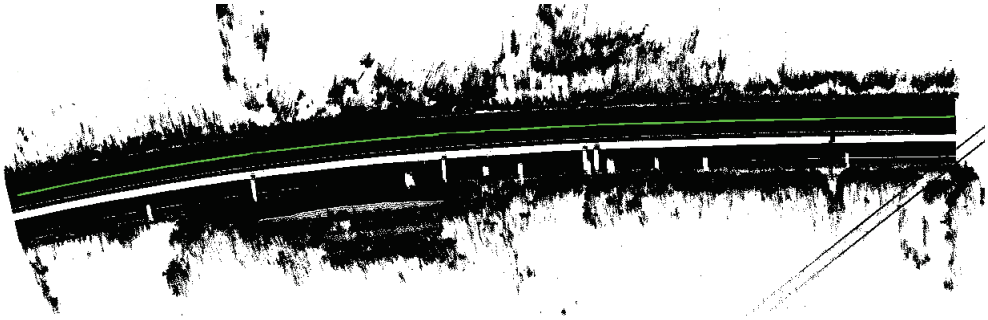
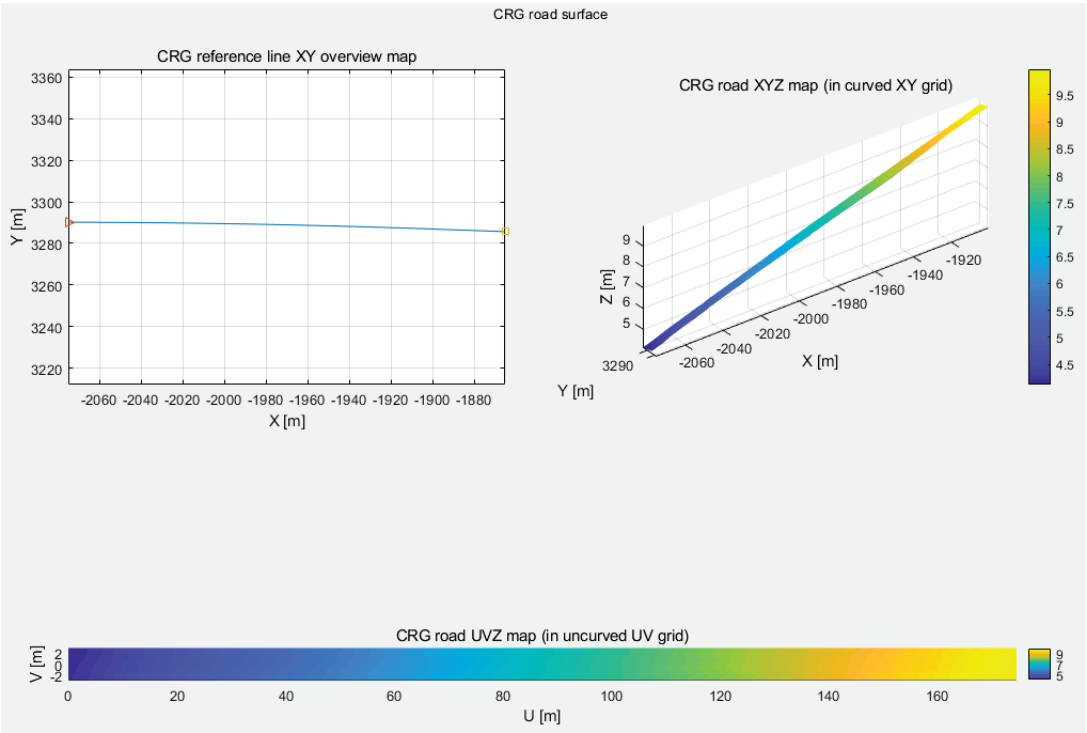


Fig. 4.11 Example of a clothoid road segment visualization.



(a) The circular road segment input point cloud. Green points are the reference line points.



(b) Result of circular road segment visualization.

Fig. 4.12 Example of a circular road segment visualization.

4.4 Conclusions

The self-driving car techniques has advanced so quickly in recent years, and it is foreseeable that autonomous vehicles will become a common feature in the near future. Since autonomous vehicles have high demand on high precision road model with detailed information about the surrounding environment, we propose an improved approach to create CRG model from mobile mapping data.

Based on our previous work, precise road surface region point cloud and trajectory are used to define the road surface. Therefore we can set accurate reference line parameters

in the process of building CRG model. On another hand, to represent a high accuracy road surface model, a two-step approach was used to create the regular grid of elevation for CRG model instead of create regular grid from point cloud directly. The elevation value are more accurately estimated by using bilinear interpolation. The experiment results show that the proposed method can create CRG model of the road in a very high resolution, and the resolution can also be customized. The visualized CRG model contains a microscopic view of road surface. It can be play an important role in the design and development of vehicles. Moreover, the proposed method is able to build road model without any manual intervention throughout the process. The next step in future development, we may further build the road network database with the support of OpenCRG and OpenDRIVE, and hence we may establish the connection to the vehicle dynamics simulators.

CHAPTER 5

Conclusions and Scope for Future Work

In this study, we have developed a workflow to build high-precision road model from the mobile mapping data.

In order to create a 3D model of lane marks, we use the combination of the information from color images and point cloud data. To precisely define the road surface region, line-based region growing algorithm is introduced to extract boundary points in the first step. We then specialized the scan-line algorithm to the input dataset. The 3D lane mark models are detected from the accurately extracted road surface region. The observation in the comparison of results shows the accuracy advantage.

Based on the road surface region and lane mark extraction, the 3D lane mark models are used to estimate trajectory points. Due to the difficulty in finding missing points between broken white lines, we introduced a length-angle space to make 3D problem 2D. Experiment results shows that the proposed method obtains the road trajectory point with continuous-curvature profile. The produced road trajectories contribute to comparison with the vehicle trajectories, and thus analyze the ADAS reliability.

As described in chapter 4, we can set accurate reference line parameters in the process of building CRG model based on precise road trajectory. Another key role in creating the CRG model, the regular elevation grid, is estimated by the bilinear interpolation. However, the existing methods cannot obtain a model that denser than input data. The numerical analysis proved the effectiveness and accuracy of our method.

For the next step in future study, we have development plan which is further build the road network database with the support of OpenCRG and OpenDRIVE, and hence we may establish the connection to the vehicle dynamics simulators.

CHAPTER 6

Bibliography

Bibliography

[1] I. Puente, H. González-Jorge, J. Martínez-Sánchez and P. Arias, "Review of mobile mapping and surveying technologies", Measurement, vol. 46, no. 7, pp. 2127-2145, 2013.

[2] K. Ishikawa, Y. Amano, T. Hashizume, J. Takiguchi and N. Kajiwara, "A Mobile Mapping System for Precise Road Line Localization Using a Single Camera and 3D Road Model", Journal of Robotics and Mechatronics, vol. 19, no. 2, pp. 174-180, 2007.

[3] S. Cavegn and N. Haala, "Image-Based Mobile Mapping for 3D Urban Data Capture", Photogrammetric Engineering & Remote Sensing, vol. 82, no. 12, pp. 925-933, 2016.

[4] S. Murray, S. Haughey, C. Deegan, M. Brogan, C. Fitzgerald and S. McLoughlin, "Mobile mapping system for the automated detection and analysis of road delineation", IET Intelligent Transport Systems, vol. 5, no. 4, pp. 221-230, 2011.

[5] N. Sairam, S. Nagarajan and S. Ornitz, "Development of Mobile Mapping System for 3D Road Asset Inventory", Sensors, vol. 16, no. 3, p. 367, 2016.

[6] J. Su, R. Miyazaki, T. Tamaki and K. Kaneda, "3D modeling of lane marks using a combination of images and mobile mapping data," International Journal of Automation Technology, vol. 12, no. 3, pp. 386-394, 2018.

[7] Y. Tan and Y. Li, "UAV Photogrammetry-Based 3D Road Distress Detection", ISPRS Int. J. Geo-Inf, vol. 8, no. 409, 2019.

[8] Z. Shen, X. Chen, X. Tang and H. Zhang, "Road Damage Feature Extraction in Image Based on Fractal Dimension", Applied Mechanics and Materials, vol. 256-259, pp. 2971-2975, 2012.

[9] L. Gézero and C. Antunes, "Road Rutting Measurement Using Mobile LiDAR Systems Point Cloud", ISPRS Int. J. Geo-Inf, vol. 8, no. 404, 2019.

[10] J. Balado, J. Martínez-Sánchez, P. Arias and A. Novo, "Road Environment Semantic Segmentation with Deep Learning from MLS Point Cloud Data", Sensors, vol. 19, no. 3466, 2019.

[11] M. Lehtomäki, A. Jaakkola, J. Hyyppä, A. Kukko and H. Kaartinen, "Detection of Vertical Pole-Like Objects in a Road Environment Using Vehicle-Based Laser Scanning Data", Remote Sensing, vol. 2, no. 3, pp. 641-664, 2010.

[12] H. Yokoyama, H. Date, S. Kanai and H. Takeda, "Pole-like Objects Recognition from Mobile Laser Scanning Data using Smoothing and Principal Component Analysis", ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. -512, pp. 115-120, 2012.

[13] B. Yang, L. Fang, Q. Li and J. Li, "Automated Extraction of Road Markings from Mobile Lidar Point Clouds", Photogrammetric Engineering & Remote Sensing, vol. 78, no. 4, pp. 331-338, 2012.

[14] L. Yan, H. Liu, J. Tan, Z. Li, H. Xie and C. Chen, "Scan Line Based Road Marking Extraction from Mobile LiDAR Point Clouds", Sensors, vol. 16, no. 6, p. 903, 2016.

[15] Y.N. Lien, T.A. Teo, C.T. Chen and P. Y. Huang, "Recognizing the Road Points and Road Marks From Mobile LIDAR Point Clouds", REMOTE SENSING ASIAN CONFERENCE. 33RD 2012, vol. 2, no. 4, pp. 1054-1059, 2012.

[16] Z. Shi, Z. Kang, Y. Lin, Y. Liu and W. Chen, "Automatic Recognition of Pole-Like Objects from Mobile Laser Scanning Point Clouds", Remote Sens, vol. 10, no. 1891, 2018.

[17] S. Tanaka, K. Nakamura, Y. Yamamoto, R. Imai, S. Kubota and W. Jiang, "Research for Generating Road Alignment of Highway Bridges with Point Cloud Data using MMS", Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, vol. 28, no. 5, pp. 826-845, 2016.

[18] G. Mastorakis and E. Davies, "Improved line detection algorithm for locating road lane markings", Electronics Letters, vol. 47, no. 3, p. 183, 2011.

[19] X. Li, X. Fang, C. Wang and W. Zhang, "Lane Detection and Tracking Using a

Parallel-snake Approach", Journal of Intelligent & Robotic Systems, vol. 77, no. 3-4, pp. 597-609, 2014.

[20] C.-F. Wu, C.-J. Kin, and C.-Y. Lee, "Applying a Functional Neurofuzzy Network to Real-Time Lane Detection and Front-Vehicle Distance Measurement", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 4, pp. 577-589, 2012.

[21] P. Kumar, C. McElhinney, P. Lewis and T. McCarthy, "Automated road markings extraction from mobile laser scanning data", International Journal of Applied Earth Observation and Geoinformation, vol. 32, pp. 125-137, 2014.

[22] A. Gressin, B. Cannelle, C. Mallet and J. Papelard, "Trajectory-based registration of 3d lidar point clouds acquired with a moible mapping system," ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 1-3, pp. 117-122, 2012.

[23] J. Thomas, K. Stiens, S. Rauch and R. Rojas, "Grid-based Online Road Model Estimation for Advanced Driver Assistance Systems". In Proceedings of 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, South Korea, 28 June-1 July 2015; pp. 2127-2145.

[24] J. Thomas and R. Rojas, "Sensor-based Road Model Estimation for Autonomous Driving". In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium, Los Angeles, CA, USA,11–14 June 2017.

[25] O. Florin and S. Nedevschi, "Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle, and Obstacle Detection", IEEE Transactions on Vehicular Technology, vol. 59, no. 3, pp. 1172-1182, 2010.

[26] A. Cappalunga, S. Cattani, A. Broggi, M. S. McDaniel and S. Dutta, "Real time 3D Terrain Elevation Mapping using Ants Optimization Algorithm and Stereo Vision". In Proceedings of 2010 IEEE Intelligent Vehicles Symposium, San Diego, USA, 21-24 June 2010; pp. 902-909.

[27] C. Boucher and J.-C. Noyer, "A General Framework for 3-D Parameters Estimation of Roads Using GPS, OSM and DEM Data", Sensors, vol. 18, no. 41, 2018.

[28] Z. Liu, J. Wang and D. Liu, "A New Curb Detection Method for Unmanned Ground Vehicles Using 2D Sequential Laser Data", Sensors, vol. 13, pp. 1102-1120, 2013.

[29] V. Potó, A. Csepinszky and Á. Barsi, "Representing Road Related Laserscanned Data in Curved Regular Grid: A Support to Autonomous Vehicles", ISPRS Inter-

national Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 2, pp. 917-921, 2018.

[30] A. Barsi, V. Potó and V. Tihanyi, "Creating OpenCRG Road Surface Model from Terrestrial Laser Scanning Data for Autonomous Vehicles". In Vehicle and Automotive Engineering 2. VAE 2018.; K. Jármai, B. Bolló, Eds.; Springer, Cham, pp. 361–369, 2018, ISBN 978-3-319-75677-6.

[31] OpenDRIVE website. 2015. Available online: http://www.opendrive.org/ (accessed on 18 November 2019).

[32] OpenCRG website. 2018. Available online: http://www.opencrg.org/ (accessed on 18 November 2019).

[33] OpenSCENARIO website. 2018. Available online: http://www.openscenario.org/ (accessed on 18 November 2019).

[34] A. Boyko and T. Funkhouser, "Extracting roads from dense point clouds in large scale urban environment", ISPRS Journal of Photogrammetry and Remote Sensing, vol. 66, no. 6, pp. S2-S12, 2011.

[35] R. Miyazaki, M. Yamamoto, E. Hanamoto, H. Izumi and K. Harada, "A line-based approach for precise extraction of road and curb region from mobile mapping data", ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. -5, pp. 243-250, 2014.

[36] R. Miyazaki, M. Yamamoto and K. Harada, "Line-Based Planar Structure Extraction from a Point Cloud with an Anisotropic Distribution", International Journal of Automation Technology, vol. 11, no. 4, pp. 657-665, 2017.

[37] E. Dougherty and R. Lotufo, "Hands-on morphological image processing," SPIE, Bellingham, Washington, pp. 29-49, 2003.

[38] V.S. Bottazzi, P.V. Borges and B. Stantic, "Adaptive regions of interest based on HSV histograms for lane marks detection". In Robot Intelligence Technology and Applications 2, pp. 677-687, Springer International Publishing, 2014.

[39] C. Mu and X. Ma, "Lane Detection Based on Object Segmentation and Piecewise Fitting", TELKOMNIKA Indonesian Journal of Electrical Engineering, vol. 12, no. 5, 2014.

[40] P. Li, H. Wang, M. Zhu and J. Liu, "Modeling and Validation of Free Road with Geometric Parameter Representation for wheeled mobile robots". In Proceedings of 2018 IEEE 14th International Conference on Automation Science and Engineering

(CASE), Munich, Germany, 20-24 August 2018; pp. 1152-1157.

[41] P. Ortiz-Coder and A. Sánchez-Ríos, "A Self-Assembly Portable Mobile Mapping System for Archeological Reconstruction Based on VSLAM-Photogrammetric Algorithm", Sensors, vol. 19, no. 3952, 2019.

[42] N. Sairam, S. Nagarajan and S. Ornitz, "Development of Mobile Mapping System for 3D Road Asset Inventory", Sensors, vol. 16, no. 367, 2016.

[43] Z. Wei, M. Yang, L. Wang, H. Ma, X. Chen and R. Zhong, "Customized Mobile LiDAR System for Manhole Cover Detection and Identification", Sensors, vol. 19, no. 2422, 2019.

[44] S. Cavegn and N. Haala, "Image-Based Mobile Mapping for 3D Urban Data Capture", Photogrammetric Engineering & Remote Sensing, vol. 82, no. 12, pp. 925-933, 2016.

[45] Y. Ma, Y. Zheng, S. Easa, M. Hou and J. Cheng, "Automated Method for Detection of Missing Road Point Regions in Mobile Laser Scanning Data", ISPRS Int. J. Geo-Inf, vol. 8, no. 525, 2019.

[46] H. Mattias and K. Laban, "White paper: A Method for Road Description Based on Map Data and Road Measurements". Available online: http://www.vipsimulation.se/upload/files/file_20160923141203/ViP_PM_2015-2_Known_Roads_White_paper_Public_2016-09-16.pdf (accessed on 18 November 2019).

[47] G. Suhyeon, "Flexible and Smooth Trajectory Generation based on Parametric Clothoids for Nonholonomic Car-like Vehicles", *PhD Thesis*, Université Clermont Auvergne, France, 2013.

[48] L. Heidi and F. Uwe, "B-Spline-Based Road Model for 3D Lane Recognition", 13th International IEEE Conference on Intelligent Transportation Systems, pp. 91-98, 2010.

[49] R. Raguram, O. Chum, M. Pollefeys, J. Matas and JM. Frahm, "USAC: A Universal Framework for Random Sample Consensus", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 2022-2038, 2013.

[50] J. Su, R. Miyazaki, T. Tamaki and K. Kaneda, "Precise Road Trajectory Estimation from Mobile Mapping Data". Presented at the International Workshop on Frontiers of Computer Vision, Gangneung, South Korea, February 2019.

[51] E. Bertolazzi and M. Frego, "On the G2 Hermite Interpolation Problem with

clothoids". Journal of Computational and Applied Mathematics, vol. 341, pp. 99-116, 2018.

[52] S. Krishnan, C. Baru and C. Crosby, "Evaluation of MapReduce for Gridding LIDAR Data", Cloud Computing Technology and Science, pp. 33-40, 2010.

[53] VIRES Simulationstechnologie GmbH. OpenCRG User Manual. 2017. Available online: http://www.vires.com/opencrg/docs/1.1.rc1/OpenCRGUserManual.pdf (accessed on 18 November 2019).

[54] Japan Government Ordinance on Road Structure Ordinance. 2014. Available online: https://elaws.e-gov.go.jp/search/elawsSearch/elaws_search/lsg0500/detail?lawId=345CO0000000320 (accessed on 18 November 2019).