

Paper:

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

Jingxin Su^{*,†}, Ryuji Miyazaki^{**}, Toru Tamaki^{*}, and Kazufumi Kaneda^{*}

^{*}Graduate School of Engineering, Hiroshima University
1-4-1 Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8527, Japan

[†]Corresponding author, E-mail: jingxinsu-d@hiroshima-u.ac.jp

^{**}Faculty of Psychological Science, Hiroshima International University, Hiroshima, Japan

[Received September 14, 2017; accepted March 30, 2018]

When we drive a car, the white lines on the road show us where the lanes are. The lane marks act as a reference for where to steer the vehicle. Naturally, in the field of advanced driver-assistance systems and autonomous driving, lane-line detection has become a critical issue. In this research, we propose a fast and precise method that can create a three-dimensional point cloud model of lane marks. Our datasets are obtained by a vehicle-mounted mobile mapping system (MMS). The input datasets include point cloud data and color images generated by laser scanner and CCD camera. A line-based point cloud region growing method and image-based scan-line method are used to extract lane marks from the input. Given a set of mobile mapping data outputs, our approach takes advantage of all important clues from both the color image and point cloud data. 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 boundary points are converted into 2D geometry. The image-based scan line algorithm is designed specifically for environments where it is difficult to clearly identify lane marks. Therefore, we use the boundary points acquired previously to find the road surface region from the color image. The experiments show that the proposed approach is capable of precisely modeling lane marks using information from both images and point cloud data.

Keywords: point cloud, mobile mapping system, region growing, scan-line algorithm, lane mark extraction

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 [6], road damage information is extracted from image data. In [7, 8], laser scanning data are used to detect vertical pole-like objects beside the road. In [9], 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, 10–16, 19, 20]. There are two main types of approaches that have been proposed: (i) approaches based on image processing [10–12, 19, 20], and (ii) methods using 3D point cloud data [13–15]. Only a few researchers have combined both images and point clouds for lane mark detection [2, 16]. 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 [16] 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 us-



ing 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 be 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 reduce the overall perspective of road track.

The rest of this paper is organized as follows. Section 2 presents the proposed approach in detail. The experimental results of extractions are provided in Section 3, and Section 4 concludes this paper.

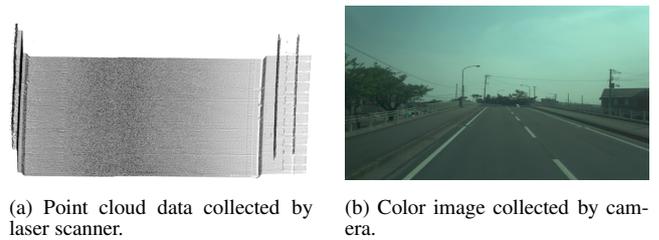


Fig. 1. Data obtained by MMS.

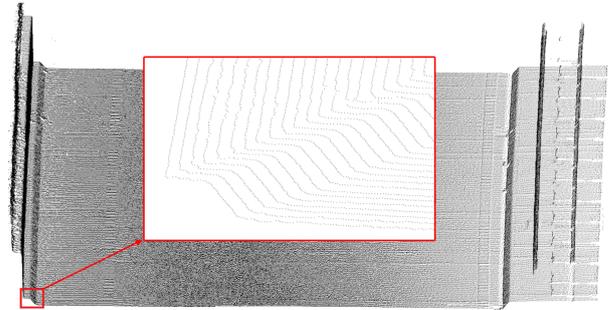


Fig. 2. The difference in point density according to the direction.

2. Method

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 structure the points. From this information, we use laser the 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 **Fig. 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 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 (**Fig. 2**).

This uneven distribution causes a problem when calculating geometric information using neighboring points because it is necessary to define a neighborhood range

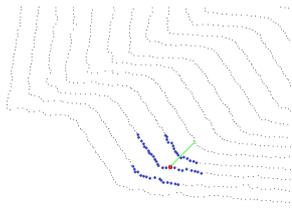


Fig. 3. The estimated normal vector at the point around the boundary. The green line is the normal vector at the point colored red. Small blue dots are the neighborhood points used for estimating the normal vector.

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. **Fig. 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 Section 2.3.

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 **Fig. 4**.

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 [17, 18] 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 [17, 18]. The normal vector in the local best-fit plane is used as the normal vector of the line segment. **Fig. 5** 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 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.

2.4. 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 **Fig. 6**.

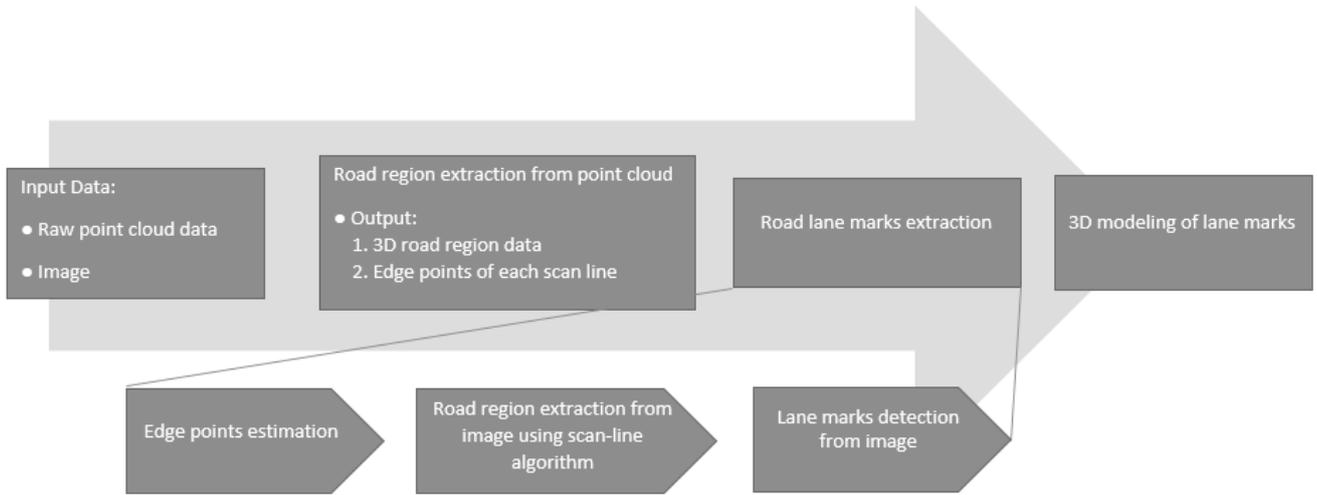


Fig. 4. Processing pipeline.

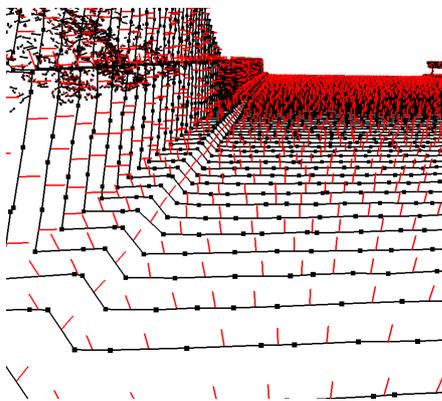


Fig. 5. Example of the line segments and normal vector estimation.

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. Fig. 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 need to be created.

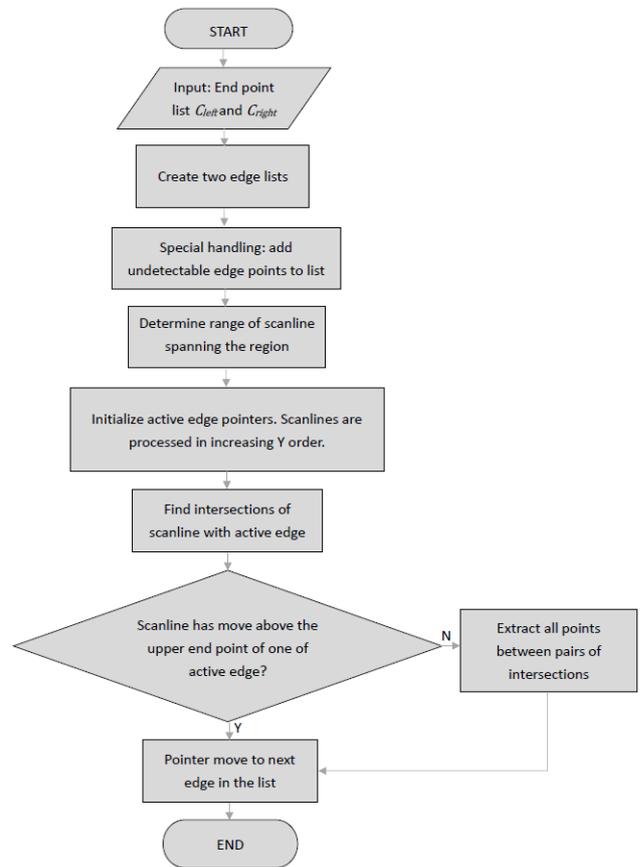


Fig. 6. Flowchart of the scan-line algorithm.

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 Fig. 8(a). To handle the parallel sides of road surface region, preprocessing is needed. The parallel side edges in Fig. 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

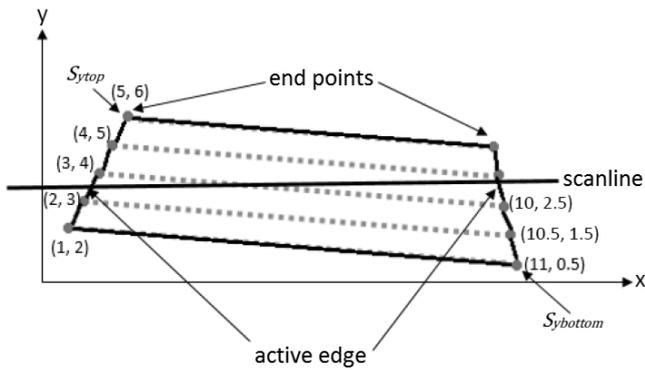
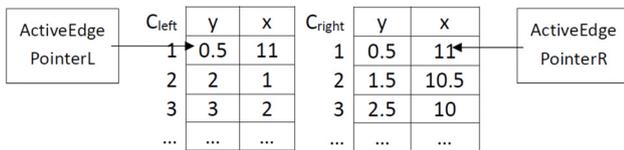


Fig. 7. Idea of the scanline algorithm with example data.

C_{left}	y	x	C_{right}	y	x
1	2	1	1	0.5	11
2	3	2	2	1.5	10.5
3	4	3	3	2.5	10
...

(a) Data structure of edge lists.



(b) Parallel sides handling.

Fig. 8. Data structure and special handling of the scan-line algorithm.

of the left edge list will be added to the bottom of the right edge list. Fig. 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 $(S_{y_{bottom}}, S_{y_{top}})$, where $S_{y_{bottom}}$ and $S_{y_{top}}$ 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 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 $S_{y_{top}}$ and obtains the road surface region from the image. Pseudocode for this scan-line algorithm is shown in Fig. 9.

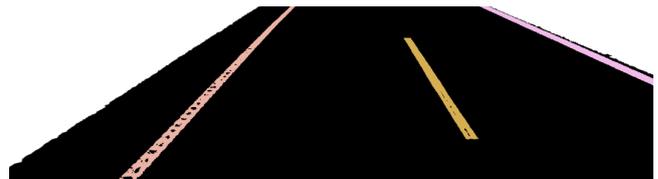
2.5. Binarization and Lane Marks Detection

Because a white lane line is a bright object against a dark background, a binarization method is carried out to

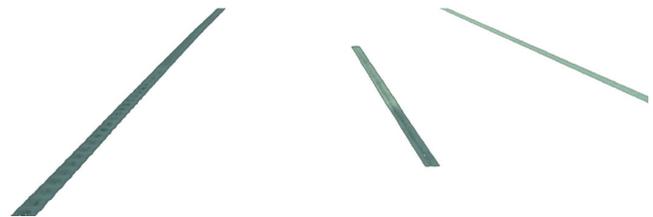
Algorithm 1 Pseudo-code for Scan-line algorithm

- 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. 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. 10. Result of lane mark extraction.

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 Fig. 10(a), where each lane mark is indicated by a different color. Fig. 10(b) shows the result of the extracted lane marks.

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 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.

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

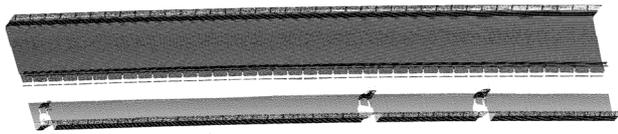


Fig. 11. The input point cloud.

Z+F IMAGER 5010 laser scanning system. **Fig. 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.

Figure 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.

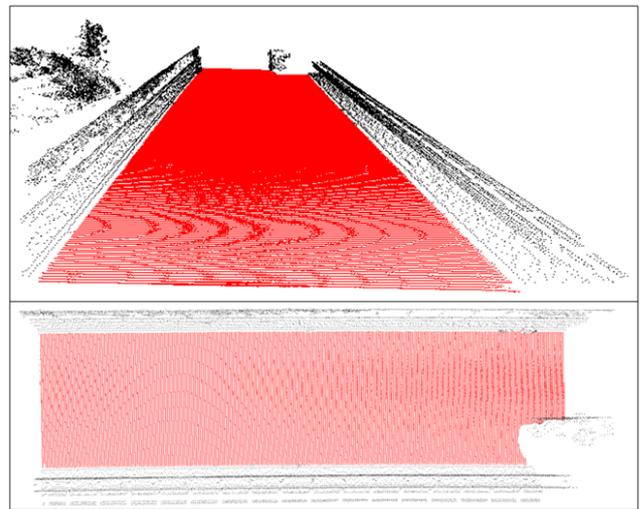
Because our region growing method uses points located at the boundary between road surface and curb, the limitation of our approach is that geometrical discontinuity is a necessary boundary of the road, as shown in **Fig. 12(b)**. In situations like this, the region may exceed the appropriate boundary.

3.1. Road Surface Region Extraction

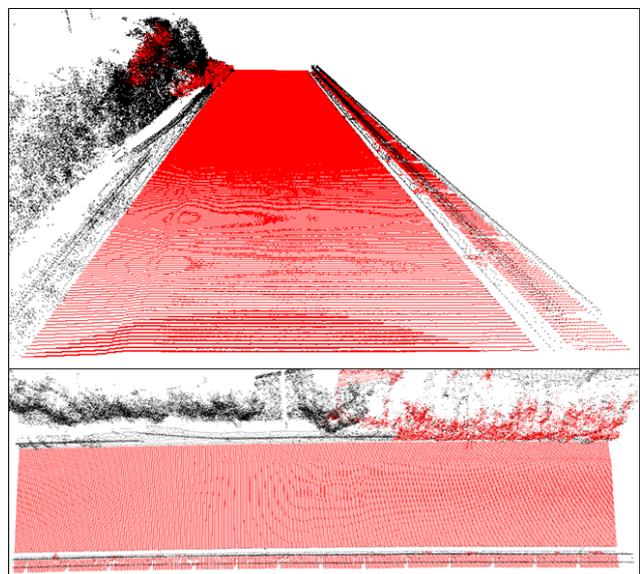
For evaluating the effect of the scan-line algorithm, we have extracted the road surface region from the image either with or without the scan-line algorithm. As a preprocessing step, we only need to project the end points to the image by using the scan-line algorithm. In contrast, without scan-line 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 1**. In the 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 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. **Fig. 13** shows a sample result from the scan-line algorithm process on a road segment.

3.2. Lane Mark Extraction

In [9, 16, 17], a manually selected ROI is used to define the road surface region in the image. Defining an ROI can



(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. 12. The results of road surface region extraction from point cloud.

Table 1. Comparison of execution time.

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

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. **Fig. 14(a)** shows the result of lane mark detection with a precise road surface region. **Fig. 14(b)** shows the result of lane mark extraction using a mask as



Fig. 13. The result of road surface region extraction from image.



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



(b) The result of lane mark detection using ROI.

Fig. 14. Comparison of lane mark detection.

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.

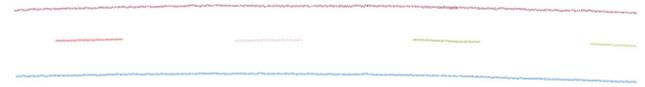
In this experiment, we do not have any ground truth extraction results to compare with the experimental results. Thus, the reference data were manually created. To evaluate the performance, we use precision, recall and *F*-measure scores as evaluation metrics. Quantitative evaluation was conducted and the results are illustrated in **Table 2**. It can be seen from **Table 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** and **Fig. 14(a)**, it can be seen that the pixels were incorrectly extracted only because of faded lane marks.

3.3. Result of 3D Modeling of Lane Marks

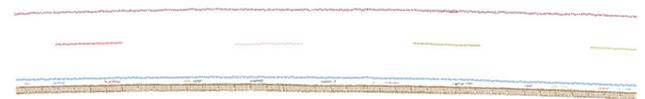
Figure 15 provides a comparison of 3D lane mark modeling results with a longer road area. The length of the road is approximately 60 m. **Fig. 15(a)** presents the result of 3D modeling the lane marks by the proposed method, in which points are grouped into 6 lane marks and each lane mark is indicated by a different color. This result indicates that the proposed method can guarantee a precise lane mark location extraction. **Fig. 15(b)** shows the result of 3D lane mark modeling using the ROI method. The

Table 2. Quantitative evaluation results.

	Our result in Fig. 14(a)	ROI result in Fig. 14(b)	Our result in Fig. 15(a)	ROI result in Fig. 15(b)
Precision	0.965	0.368	0.981	0.690
Recall	0.963	0.963	0.974	0.974
<i>F</i> -measure	0.964	0.533	0.977	0.808



(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. 15. Comparison of 3D lane mark modeling.

input images used are obtained through the same ROI filter as in **Fig. 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.

4. Conclusions and Future Work

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. In the future, we will enhance the applicability, make our approach suitable for more complex environments, and create a precise 3D road track model.

Acknowledgements

The mobile mapping datasets used in this work were provided by Sanei Co. Japan.

References:

- [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," *J. 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] 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.
- [7] M. Lehtomäki, A. Jaakkola, J. Hyypä, 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.
- [8] 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 – Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol.512, pp. 115-120, 2012.
- [9] A. Boyko and T. Funkhouser, "Extracting roads from dense point clouds in large scale urban environment," *ISPRS J. of Photogrammetry and Remote Sensing*, Vol.66, No.6, pp. S2-S12, 2011.
- [10] G. Mastorakis and E. Davies, "Improved line detection algorithm for locating road lane markings," *Electronics Letters*, Vol.47, No.3, p. 183, 2011.
- [11] X. Li, X. Fang, C. Wang, and W. Zhang, "Lane Detection and Tracking Using a Parallel-snake Approach," *J. of Intelligent & Robotic Systems*, Vol.77, No.3-4, pp. 597-609, 2014.
- [12] 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 Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol.42, No.4, pp. 577-589, 2012.
- [13] P. Kumar, C. McElhinney, P. Lewis, and T. McCarthy, "Automated road markings extraction from mobile laser scanning data," *Int. J. of Applied Earth Observation and Geoinformation*, Vol.32, pp. 125-137, 2014.
- [14] 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.
- [15] 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.
- [16] 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 Conf. 33rd 2012*, Vol.2, No.4, pp. 1054-1059, 2012.
- [17] 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.
- [18] R. Miyazaki, M. Yamamoto, and K. Harada, "Line-Based Planar Structure Extraction from a Point Cloud with an Anisotropic Distribution," *Int. J. of Automation Technology*, Vol.11, No.4, pp. 657-665, 2017.
- [19] V. S. Bottazzi, P. V. Borges, and B. Stantic, "Adaptive regions of interest based on HSV histograms for lane marks detection," *Robot Intelligence Technology and Applications 2*, pp. 677-687, Springer Int. Publishing, 2014.
- [20] C. Mu and X. Ma, "Lane Detection Based on Object Segmentation and Piecewise Fitting," *TELKOMNIKA Indonesian J. of Electrical Engineering*, Vol.12, No.5, 2014.



Name:
Jingxin Su

Affiliation:
Doctor's Course Student, Graduate School of Engineering, Hiroshima University

Address:
1-4-1 Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8527, Japan

Brief Biographical History:
2014- Graduate School of Engineering, Hiroshima University



Name:
Ryuji Miyazaki

Affiliation:
Associate Professor, Hiroshima International University

Address:
555-36 Gakuendai, Kurose, Higashi-hiroshima, Hiroshima 739-2695, Japan

Brief Biographical History:
2001- Research Associate, Hiroshima International University
2009- Lecturer, Hiroshima International University
2017- Associate Professor, Hiroshima International University

Main Works:

- "Line-Based Planar Structure Extraction from a Point Cloud with an Anisotropic Distribution," *Int. J. Automation Technology*, Vol.11, No.4, pp.657-665, 2017.
- "Polygonal Model Creation with Precise Boundary Edges from a Mobile Mapping Data," *Int. J. of Computer Science and Network Security*, Vol.16, No.11, pp.124-133, November, 2016.

Membership in Academic Societies:

- Japan Society for Precision Engineering (JSPE)
- Institute of Electronics, Information and Communication Engineers (IEICE)
- Information Processing Society of Japan (IPSJ)



Name:

Toru Tamaki

Affiliation:

Associate Professor, Graduate School of Engineering, Hiroshima University

Address:

1-4-1 Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8527, Japan

Brief Biographical History:

2005- Associate Professor, Graduate School of Engineering, Hiroshima University

2001- Assistant Professor, Niigata University

Main Works:

- T. Hirakawa, T. Tamaki, T. Kurita, B. Raytchev, K. Kaneda, C. Wang, and L. Najman, "Tree-wise Discriminative Subtree Selection for Texture Image Labeling," IEEE Access, Vol.5, pp. 13617-13634, 2017. doi: 10.1109/ACCESS.2017.2725319

Membership in Academic Societies:

- Institute of Electrical and Electronics Engineers (IEEE)
 - Institute of Electronics, Information and Communication Engineers (IEICE)
 - Information Processing Society of Japan (IPSJ)
-



Name:

Kazufumi Kaneda

Affiliation:

Professor, Department of Information Engineering, Hiroshima University

Address:

1-4-1 Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8527, Japan

Brief Biographical History:

1986- Joined Hiroshima University

1991-1992 Visiting Researcher, Brigham Young University

2004- Professor, Hiroshima University

2014- Graduate School of Engineering, Hiroshima University

Main Works:

- "Tristimulus Value Rendering of Wavelength-Dependent Phenomena Using Importance Sampling," Research on computer graphics and bio-medical visualization, IEEJ Trans. on Image Electronics and Visual Computing, Vol.5, No.2, pp. 74-82, 2017.

Membership in Academic Societies:

- Association for Computing Machinery (ACM)
 - Information Processing Society of Japan (IPSJ)
 - Institute of Electronics, Information and Communication Engineers (IEICE)
-