This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

Generation of a Precise and Efficient Lane-Level Road Map for Intelligent Vehicle Systems

Gi-Poong Gwon, Woo-Sol Hur, Seong-Woo Kim, *Member, IEEE*, and Seung-Woo Seo, *Member, IEEE*

***Abstract*—The development of intelligent vehicle systems has resulted in an increased need for a high-precision road map. However, conventional road maps that are used for vehicle navigation systems or geographical information systems are insufficient to satisfy new requirements of intelligent vehicle systems such as autonomous driving. There are three primary road map requirements for intelligent vehicle systems: centimeter-level accuracy, storage efficiency and usability. However, no existing researches have met these three requirements simultaneously. In this paper, we propose a precise and efficient lane-level road map generation system that conforms to the requirements all together. The proposed map building process consists of three steps: 1) data acquisition, 2) data processing, and 3) road modeling. The road data acquisition and processing system captures accurate 3D road geometry data by acquiring data with a mobile 3D laser scanner. The road geometry data is then refined to extract meta information, and in the road modeling system, the refined data is represented as sets of piecewise polynomials to ensure storage efficiency and usability of the map. The proposed mapping system has been extensively tested and evaluated on a real urban road and highway. The experimental results show that the proposed mapping system outperforms conventional ones in terms of the road map requirements.**

***Index Terms*—road map, lane-level road map, road modeling, piecewise polynomial approximation.**

I. Introduction

SIGNIFICANT efforts have been made in academia and in industry to develop intelligent vehicle systems that can provide a higher level of safety and convenience [1], [2]. To this end, it is necessary to have a precise lane-level road map to provide various autonomous driving vehicle applications, such as vehicle path or motion planning [3]–[6] and ego-vehicle localization [7]–[12]. In addition, various advanced driver assistance systems (ADASs) including lane keeping, lane change assist, and fuel management systems, benefit from having precise lane-level road maps [13].

From a technical point of view, a map for intelligent and autonomous vehicle systems has to fulfill the following three requirements: accuracy, storage efficiency, and usability. First, a road map should contain the geometry of all lanes with accuracy at the centimeter-level as well as the 3D structure of roads in order to represent various type of roads, including sloped roads, overpasses, etc. Second, the geometry of the road should be expressed in a compact form such that the map can be downloaded and updated via wireless networks. Finally, the road geometry should be expressed in an application-friendly format. Specifically, low computational efforts should be necessary to calculate the road geometry information, including coordinates, tangent angle or curvature from the road map to provide real-time operation.

Conventional road maps only provide macro-scale information for vehicle navigation systems or geographical information systems (GISs), so many studies have been carried out to produce a precise road map intended for use with intelligent vehicle systems. Bétaille *et al.* proposed an Emap structure for meso-scale lane-level driving assistance that utilizes GPS and Dead Reckoning (DR) [14]. However, the quality of the map is easily affected by the data acquisition process and is inadequate for autonomous driving purpose. A map for autonomous driving requires micro-scale precision, but such a precise map requires a large amount of storage space and has a low level of usability [15]. Jo *et al.* proposed a mapping system that can generate a storage-efficient map suitable for autonomous driving purposes without considering lanes, complex junctions and 3D traffic structures such as an overpass and a slope [16].

Few previous studies have considered the latter two requirements while providing centimeter-level functionality. In this paper, we propose a road map generation system that considers the three aforementioned road map requirements altogether. The road map generation system is composed of three subsystems, including data acquisition, processing, and road modeling systems. The data acquisition and processing system acquires accurate road geometry data that fulfills the accuracy requirement, and we introduce a data acquisition system to obtain highly-accurate 3D road geometry by utilizing 3D mobile laser scanning (3D MLS) data. Then, the road geometry data that is obtained is refined to extract the corresponding meta information.

The main contribution of this paper lies in the road modeling system. In the road modeling system, the road geometry data that is obtained is approximated and represented as sets of piecewise parametric polynomials in order to conform to the latter two road map requirements. The piecewise polynomial is more effective in terms of usability than those previously used in other road models, including clothoid or B-spline, be-

The authors are with Seoul National University, Korea (e-mail: gwons@snu.ac.kr, bpoebiapl@snu.ac.kr, sinabrlo@snu.ac.kr, and sseo@snu.ac.kr).

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (2009-0083495), by the Industrial Strategic Technology Development Program (10042808, Development of Driver Assistant System Using Camera, Radar and Road Characteristic) funded by the Ministry of Knowledge Economy(MKE, Korea), and in part by the Institute of New Media and Communications and the Automation and Systems Research Institute, Seoul National University.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

cause various types of road geometry information such as the tangent angle and the curvature of the road can be calculated by conducting simple arithmetic operations. In addition, the piecewise polynomial form is suitable for local modification of the map, since the modification of the coefficients of a piecewise polynomial does not affect the shape of the other piecewise polynomials.

The main problem in representing the road as the sets of piecewise polynomials is the need to minimize the required data size, since the geometry data about the length of the tens-of-thousands kilometers of roads should be stored in a storage device of a vehicle. However, previous piecewise polynomial-based curve approximation algorithms are inappropriate for processing the significantly large amount of road geometry data, since these global optimization algorithms have exponential or $O(n^3)$ computational complexity [17], [18]. In this paper, we propose an efficient road modeling algorithm that has $O(n)$ complexity. The proposed algorithm suboptimally determines the number of piecewise-polynomials of the overall curve and its coefficients by converting the global approximation problem to a combination of small easy-to-solve approximation problems.

The proposed road map generation system has been extensively tested and evaluated on an urban road and a highway, and the results of the experiment show that the proposed map outperforms conventional maps in terms of the road map requirements. Our scheme can thus contribute to building a precise and efficient map for navigation systems that can be embedded in autonomous driving vehicles and updated via wireless networks.

The main contributions of this paper can be summarized as follows:

- We propose a road map generation system to generate a precise and efficient lane-level road map for intelligent and autonomous vehicle systems.
- We propose an efficient curve approximation algorithm that represents the road as the minimum number of piecewise polynomials.
- The feasibility and practicality of the proposed map were evaluated by conducting extensive experiments on an urban road and a highway.

The remainder of this paper is organized as follows. Section II presents related works. Section III presents the overall system architecture of the road map generation system. Section IV introduces the road geometry data acquisition and processing system, and Section V presents the proposed road representation system. Section VI provides the experimental results and Section VII concludes this paper.

II. Related Works

The accuracy, storage efficiency and usability of a map all depend on the combination of data acquisition and road modeling methods that are used. The objective of data acquisition is to obtain accurate data to represent the actual geometry of the road, and the primary purpose of road modeling is to effectively represent the road geometry in terms of the storage efficiency and usability while maintaining a certain level of accuracy. Table I presents the data acquisition and road modeling methods used in previous representative studies and also describes the extent to which they meet the three road map requirements for intelligent vehicle systems. The vertical axis of the table represents the classification of the data acquisition methods based on the level of accuracy that the methods can meet, and the horizontal axis represents the classification of the road models based on the level of usability. The asterisk in the table indicates that the marked method considers the maximization of storage efficiency. The table shows that no prior system has simultaneously achieved the three road map requirements for intelligent vehicle applications. A more detailed description about the previous approaches is presented in the rest of this section.

A. Acquisition of Road Geometry

Various approaches have attempted to acquire accurate road geometry data. For example, an aerial/satellite image-based approach has been extensively used for conventional digital maps and GISs. High-resolution aerial camera images have been acquired from a satellite or a aerial vehicle, to extract road geometries through manual work or through image processing means [22]–[24]. An image-based approach has an advantage in that we can obtain the road geometry for a larger region by processing a single image. However, the resolution of the aerial/satellite images is insufficient for extracting precise lane-level road geometry. Moreover, the elevation information of the roads cannot be acquired because the images contain no depth information. As a consequence, the accuracy of the image-based approach is limited to meter-level.

Many studies have adopted a probe vehicle-based approach to acquire a more accurate road geometry [13], [14], [16], [25]–[27]. In this approach, a probe vehicle equipped with various sensors explores roads and collects sensor data to obtain road geometry information. Of the various sensor configurations that are possible, kinematic GPS-based methods are the most widely used systems that include real-time kinematic (RTK) and post-processing kinematic (PPK) GPSs. In this method, the trajectory of a probe-vehicle driving along the centerline of a road (or lane) is recorded as road geometry data. The accuracy of the road geometry depends mainly on the positioning accuracy of the vehicle, so many previous studies have used algorithms that integrate GPS with other positioning systems, such as dead reckoning (DR) and inertial navigation system (INS), to improve the accuracy and reliability of the vehicle position [16], [28], [29].

Although the GPS-based methods are useful to obtain a greater degree of accuracy in the road geometry than image-based methods, there are some fundamental limitations to using this method. First, it is inefficient for the probe vehicle to capture information for a road multiple by as many times as the number of lanes. Such repetitive work is costly and time-consuming. The second problem is the so-called trajectory error. In a GPS-based method, it is desired for a probe vehicle to drive along the exact centerline of the lane. However, it is practically impossible for a human driver to control the vehicle with centimeter-level precision for an extended period of time.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

TABLE I
COMPARISON TO REPRESENTATIVE PREVIOUS ROAD MAP GENERATION METHODS.

			Usability				
			N/A	Low	Mid.		High
		Road model / Data Acquis.	None	Polygon	Clothoid	B-spline	Set of piecewise polynomials
Accuracy	Meter	Aerial/satellite image		OSM[a] [19]			
	Decimeter	GPS+DR[b] (INS[c])			Bétaille, *et al.* [14]*	Jo, *et al.* [16]* .	
	Centimeter	GPS+DR (INS)+Camera			Ziegler, *et al.* [20]		
		GPS+DR (INS)+3D Lidar	Joshi, *et al.* [21]				Proposed*

*Methods where the maximization of storage efficiency is considered.
[a]Open street map, [b]Dead reckoning, [c]Inertial navigation system.

Therefore, discordance between the vehicle trajectory and the centerline of the lane is unavoidable. Thus the accuracy of using this method is limited to decimeter-level.

Various methods based on perception sensor have been introduced in recent years to tackle the problems of using GPS-based methods [20], [26], [30]–[33]. In these methods, road markings are detected and extracted by using perception sensors that capture road geometry data. Since the road geometry data is acquired directly from lane marking information, the above problems are fundamentally prevented. Various sensor configurations have been used to this end, including a monocular camera [26], stereo-camera [20], and 2D or 3D Lidar [21], [31]–[33]. Camera-based methods have an advantage in that the cost of the sensor can be reduced; however, it is difficult to extract precise 3D road geometry because of the inherent limitation of a camera, which is that the information is represented in a 2D plane. For this reason, the 3D Lidar is the most appropriate way to acquire centimeter-level road geometry since it can provide accurate 3D information about roads. Some studies have extracted road geometry from 3D Lidar data. In [31], an algorithm was proposed to extract road geometry from 3D Lidar data; however, the algorithm focused on extracting road regions rather than lane-level road geometry. In [32] and [33], the extraction of lane markings from 3D Lidar data was considered, but, the clustering among lanes was not addressed. Joshi *et al.* proposed a particle filter-based method to extract lane-level geometry data and clustering among the lanes using 3D Lidar data [21]. However, that method extracts the centerline of lanes rather than the lane marking points, so the shape of the lane markings (e.g., the dashed line) is ignored, which can be useful for intelligent vehicle applications such as vehicle localization. In this paper, we propose a 3D Lidar data processing algorithm to extract and cluster the lane marking points.

B. Modeling of Road Geometry

It is important to have the appropriate road geometry representation to ensure storage efficiency and usability along with map accuracy. Various road geometry models have been previously proposed [13], [14], [16], [26], [34], [35], but the previous models have not considered the three requirements simultaneously. For example, polygons are widely used in conventional digital road maps and various intelligent vehicle applications to represent the road geometry due to their simplicity [19], [36]–[39]. However, since a polygon cannot express curved roads with precision, too many line segments become necessary to express the roads with a large curvature in order to achieve centimeter-level accuracy. Therefore, the accuracy and storage efficiency requirements cannot be simultaneously satisfied. In addition, direct exaction of road geometry information cannot be provided, including the tangent angle and curvature, and thus the usability requirement is also not satisfied.

Various mathematical curve models have been proposed to provide a more accurate and efficient road representation. For example, a clothoid is the best way to represent road geometry since a road is traditionally designed by a set of clothoids [14], [40]. Bétaille *et al.* proposed a clothoid-based road modeling algorithm that satisfies the accuracy and storage efficiency requirements [14]. In this algorithm, the road geometry is represented as a set of the minimum number of clothoid curves while also satisfying a preset accuracy constraint. However, a clothoid is inadequate for use with intelligent vehicle applications since it includes transcendental functions. Thus the calculations for the information required for advanced vehicle applications, such as distance and relative angle between an ego-vehicle and a lane in the map, are computationally intense. As a result, the clothoid does not satisfy the usability requirement. Moreover, a clothoid cannot express the 3D road geometry by itself since the clothoid is only defined on a plane.

An alternative is to use a spline curve. Many studies have used various types of spline curves to represent the road geometry [13], [16], [26], [34], [35]. For example, a cubic B-spline is a representative method for road modeling that has an advantage in that local modifications of the curve do not affect to the entire shape of the curve. This is an important characteristic for road map maintainability, and in addition, many efficient B-spline curve approximation algorithms that have been developed for computer aided design (CAD) make it easy to use B-spline curves for road modeling. For example, in literature [16], a B-spline-based road modeling algorithm was proposed based on a dominant points-based B-spline curve fitting algorithm developed for computer-aided design (CAD) [41]. The control point of the B-spline is sub-optimally determined according to a preset accuracy constraint in order to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

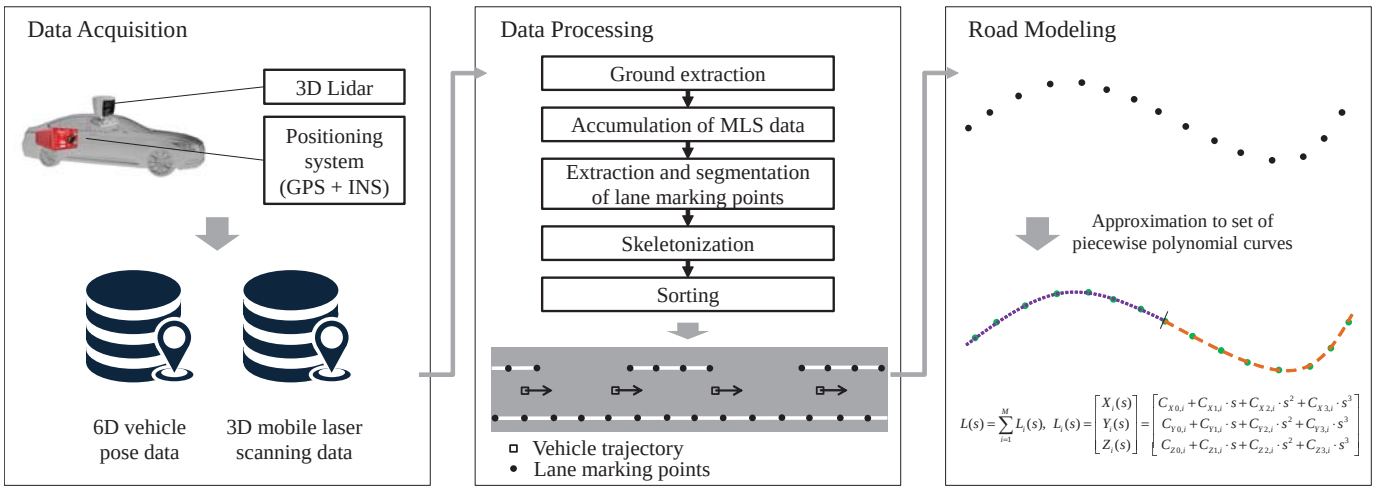


Fig. 1. Overall System Architecture.

simultaneously adhere to the accuracy and storage-efficiency requirements. However, it is difficult to extract the information on the geometry of the road, such as the tangent angle and curvature by using a B-spline curve because the functions of the B-spline are not intuitive, and it is thus difficult to calculate the first and second derivatives of the functions [42]. Therefore, B-splines also do not satisfy the usability constraint.

In this paper, we use the most intuitive spline curve form where a curve segment is expressed as a piecewise cubic polynomial. Since a polynomial can be used to conveniently calculate the derivatives, we can extract the tangent angle and curvature of the road from the proposed spline curve by using simple arithmetic operations. Piecewise polynomial-based curve approximation has not been studied as much as other spline curves such as the B-spline. Moreover, previous piecewise polynomial approximation algorithms are not appropriate for approximating significantly large amount of road geometry data since the previous algorithms have exponential or $O(n^3)$ computational complexity [17], [18]. One solution is to approximate the data as a B-spline curve first and to convert the B-spine curve to a set of piecewise polynomials [43]. However, this approach requires a great number of piecewise polynomials than directly approximating the data as a set of piecewise polynomials as will be shown in Section VI-C2. Thus, this paper proposes an efficient piecewise polynomial curve approximation algorithm that processes a significantly large amount of road geometry data and sub-optimally determines the number of piecewise polynomials and their coefficients with $O(n)$ computational complexity.

III. Overall System Architecture

The overall road map generation system is composed of three subsystems: a data acquisition, a data processing and a road modeling subsystems (see Fig. 1). The data acquisition and processing subsystems acquire accurate and reliable road geometry data by using various sensors during collection on a probe vehicle equipped with a GPS+INS vehicle positioning system and a 3D laser scanner. Then, accurate road geometry data is extracted by integrating the sensor data in the data processing system.

In this paper, two different types of road geometry are acquired depending on the road type. For a road in which lane markings exist, the Cartesian coordinates of the lane markings are obtained as the road geometry data from the 3D laser scanning data (see the black circles in Fig. 1). For a road in which lane markings do not exist, such as when going off-road, the trajectory of a probe vehicle driving along centerline of a road is used in a manner similar to the previous probe vehicle-based approach (see the white squares in Fig. 1). For convenience, we have name the road in which lane markings exist as a *type I* road and the other as a *type II* road in the rest of this paper.

Both types of road geometry data obtained from the data acquisition and processing systems are represented by a large number of points. However, this type of geometry representation is inadequate for a road map in terms of the storage efficiency. Moreover, the point representation makes it difficult to extract the required road geometry information, such as tangent angle and the curvature of a road. To solve these problems, we have applied a mathematical modeling technique to the road geometry point data in order to increase its storage efficiency and usability. The road modeling system approximates a set of geometry point data that correspond to a road line as a cubic spline curve consisting of sequential parametric piecewise cubic polynomials curves (see the right bottom of Fig. 1). A *sequential approximation algorithm* is proposed to efficiently approximate the curve.

IV. Road Geometry Data Acquisition and Processing

In the data acquisition, the required raw sensor data is collected by using a probe vehicle equipped with a positioning system and a 3D Lidar, and the geometry data of the lanes is then obtained from raw data in the data processing step. Detailed description of the two steps is presented as follows.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

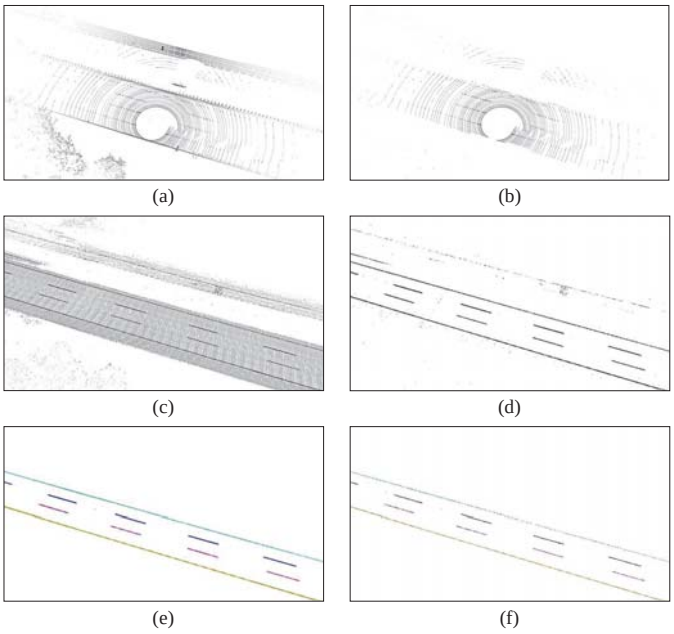


Fig. 2. Overall data processing procedure: (a) raw MLS data, (b) ground extraction result, (c) result of data accumulation, (d) result of intensity thresholding, (e) result of lane marking point extraction and clustering, and (f) skeletonization result.

A. Data Acquisition

During the data acquisition step, two kinds of data are collected by the probe vehicle: 6D vehicle pose data and 3D MLS data. The vehicle pose data includes the accurate 3D global position and 3D attitude (yaw, pitch and roll) with respect to the trajectory of the probe vehicle. In order to obtain the accurate vehicle pose data, a high-precision vehicle positioning system is recommended. In this paper, we use a sensor fusion system that integrates a RTK-GPS and a high-precision INS. The 3D MLS data is obtained by correcting the raw 3D laser scanning data using the INS data. When a vehicle moves, the measurement origin of the laser scanner is changed; thus the laser points are twisted [44]. To resolve this problem, a preprocessor untwists the points using the vehicle motion sensor measurements from the INS.

For *type II* roads, the vehicle pose data is simply used as the road geometry data, and for *type I* roads, MLS data is additionally used. The rest of this section presents the data processing algorithm that is used to obtain accurate road geometry data from the raw data. Note that this paper focuses more on a *type I* road since the road geometry for *type II* roads can be obtained by using the previous probe vehicle-based methods.

B. Data Processing

The road geometry data for a *type I* road can be obtained by integrating 3D MLS data with the vehicle pose data. Fig. 2 depicts the overall procedure for the data processing step. First, ground extraction is applied to each frame of the MLS data to remove unnecessary points from the MLS data, and then, the ground points for every frame of the MLS data are accumulated on a global coordinate system by using 6D vehicle pose data synchronized with MLS data. The accumulated ground points with a high intensity are then extracted as candidate points that correspond to lane markings, and the exact lane marking points are extracted and clustered by using more complex algorithms. Finally, the desired geometry data is obtained by skeletonizing the dense point data.

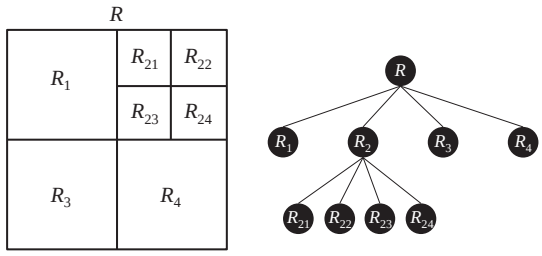


Fig. 3. Quadtree partition structure which is used for ground extraction. Based on the quadtree structure, the point cloud is divided into multiple cells until all the cells are identified as ground cells or non-ground cells.

1) Ground extraction: Since the lane marking points on the ground surface are the only points of interest in this paper, removing off-ground points from each MLS is helpful for later processes in terms of the processing time and performance. Ground extraction is carried out according to an adaptive quadtree partition structure, as shown in Fig. 3. Initially, the MLS data points are divided into multiple cells with a size of w_{init}, and then, each cell is tested to identify whether the cell is a ground cell or not. If it is not certain whether a cell is a ground cell or not, the cell is divided into four child cells, and the test is applied to all child cells. This process is then continued until all the cells have been identified. The test to identify this type of cell is conducted as follows.

First we assume that there is at least one cell among the initial cells that has already been identified as a ground cell. This assumption is valid since it is certain that the cells near the probe vehicle are ground cells. Then, the test is performed starting from the unidentified cells neighboring the identified cells. Let an unidentified cell be R_{UI}. Then the roughness of the cell is calculated to determine whether the cell is a ground cell or not by

$$\Delta z_{cell} = z_{\max}(R_{UI}) - z_{\min}(R_{UI}), \tag{1}$$

where $z_{\max}(\cdot)$ and $z_{\min}(\cdot)$ refer to the z coordinate value of the highest points and the lowest point in the cell, respectively. If the roughness value of the cell is larger than a given threshold, this means that the cell has to include the off-ground points. In this case, the cell is unidentified, and thus is divided into four child cells again. On the other hand, if the roughness value of the cell is smaller than the given threshold, it is then considered to be a candidate ground cell. For a candidate ground cell, the identification whether the cell is actually a ground cell is performed according to the following test:

$$|z_{\text{mean}}(R_{\text{UI}}) - z_{\text{mean}}(R_{\text{I,Nbh}})| < \lambda_{\text{GND}} \tag{2}$$

where $R_{\text{I,Nbh}}$ denotes the ground cell that has already been identified among the neighborhood of the test cell R_{UI}, and $z_{\text{mean}}(\cdot)$ refers to the mean value of heights of the points. If the test result is true, the height difference between the test cell

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

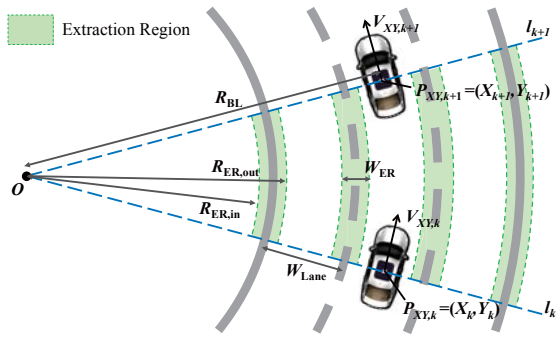


Fig. 4. Lane marking point extraction and clustering. The extraction regions are formed on the vehicle pose data. The points inside the extraction region are extracted as the lane marking points.

and the neighbor ground cell is small, and therefore, the cell is identified as a ground cell. If the test result is false, then the test cell is identified as an off-ground cell. As a consequence, a cell is categorized into three states after this process: a ground cell, an off-ground cell or an unidentified cell. For the unidentified cells, the above process is repeated until all cells have been identified. Fig. 2b shows an example of the ground extraction.

2) Accumulation of MLS data: Every frame of the MLS data from which off-ground points have been removed is accumulated on a global coordinate system by using synchronized vehicle pose data. Basically, the MLS data is represented on a vehicle body coordinate system, and therefore, a rigid body transformation is applied to accumulate these in a global coordinate system based on the Euler angles (yaw, pitch and roll) of the vehicle body, and the position of the vehicle in a global coordinate system. Finally, dense point cloud data representing the ground region is obtained, as shown in Fig. 2(c).

3) Extraction and Clustering of Lane Marking Points: Now, the points corresponding to the lane markings are extracted and are clustered among the points that belong to the same road line. First, rough lane marking points are extracted by using simple intensity-based thresholding, i.e., the points with a higher intensity than the threshold are extracted. We assume that an appropriate static intensity threshold is given for a road, since the intensity value of the Lidar is robust to external factors such as illumination. As we can see in Fig. 2(d), the majority of non-lane marking points are effectively filtered out just through the simple intensity thresholding method. However, there are still exist outliers, and in addition, the points are not clustered by the road line. Therefore, an additional algorithm is applied to remove the remaining outliers and to cluster the points by the road line, and since it is difficult and inefficient to process all points at once, we first vertically divide the point cloud into a set of data blocks by using the vehicle pose data in the XY plane. A block is then defined by two lines, named a split line, where the kth split line, l_k, is a line that is orthogonal to the yaw of the kth vehicle pose and the passing vehicle position $P_{XY,k}$ (see the coarse dashed lines in Fig. 4). With the split lines, all points are divided into N_p-1 blocks, where N_p is the number for the vehicle pose data (number of split lines). These data blocks are then processed separately to extract the lane marking points as follows.

The algorithm starts by finding an intersection for two split lines, l_k and l_{k+1}, and the radius R_{BL} with respect to the intersection point. Note that we assume that the probe vehicle drives along the centerline of a lane as much as possible, and the vehicle trajectory between time k and $k+1$ can be modeled by an arc since the time gap is small enough (10 ms in this paper). After that, the lane marking point extraction regions in which the lane marking points are likely to exist are set using some prior information of the road, e.g, the number of lanes, N_{L}; width of a lane, W_{L}; and the lane number in which the probe vehicle drove, n_{drv} ($n_{\mathrm{drv}} = 1$ for the leftmost lane). This information can be obtained using conventional maps, such as Open Street Map (OSM) [19], or by using camera-based algorithms [45]–[47]. There is an exceptional case where the yaw angles of two consecutive vehicle pose data are exactly the same; thus, the intersection point of two split lines does not exist. In this case, the radius R_{BL} is set to a value that is large enough (e.g., 3000 m), and the origin O is set to a point so that it satisfies $\|P_{XY,k} - O\| = \|P_{XY,k+1} - O\| = R_{\mathrm{BL}}$. This trick is valid since the radius is large enough compared to the length of the arc between points $P_{XY,k}$ and $P_{XY,k+1}$ (about 2.7 m where the speed of the vehicle is 100 km/h and the pose data is acquired at a 10 Hz data rate) As shown in Fig. 4, the extraction region is bounded by two inner and outer arcs, and the radiuses of the arcs are determined by R_{BL}, N_{L} and W_{L} as follows:

$$\begin{aligned} R_{\mathrm{ER},i} &= \{R_{\mathrm{ER},i,\mathrm{in}}, R_{\mathrm{ER},i,\mathrm{out}}\} \\ &= R_{\mathrm{BL}} + (2i - 2n_{\mathrm{drv}} - 1)\frac{W_L}{2} \pm \frac{W_{ER}}{2}, \end{aligned} \tag{3}$$

where i denotes the index of the road lines from the left (e.g., $i = 1$ for the leftmost road line), and the width of the extraction region, W_{ER}, is determined by considering the uncertainty degree with respect to the error between the expected location of the lane marking points and the true location of these. Although the larger value for W_{ER} may decrease the false negative rate, more non-lane marking points may be extracted. On the other hand, a smaller W_{ER} value may exclude non-lane marking points as well, which would decrease the true positive rate. Therefore, it is important to carefully determine W_{ER} value to ensure reliable road marking point extraction. Finally, the points inside of the extraction region are extracted as the lane marking points, and the points belonging to the same region are grouped together. After finishing the extraction procedure, the points belonging to the same region are naturally grouped, and thus the lane marking points are clustered into several clusters that correspond to each of the road lines.

4) Skeletonization: In the previous steps, the lane marking points were obtained as shown in Fig. 2(e). However, since the lane marking points describe thick lane markings at this stage, it is difficult to represent lane markings as lines by using this set of lane marking points. Skeletonization is applied to the lane marking points to represent the lane markings as lines, as shown in Fig. 2(f). Skeletonization is a technique that is used to simplify and abstract a volumetric object to a line shape [48]. In this paper, the skeletonization is applied to lane

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

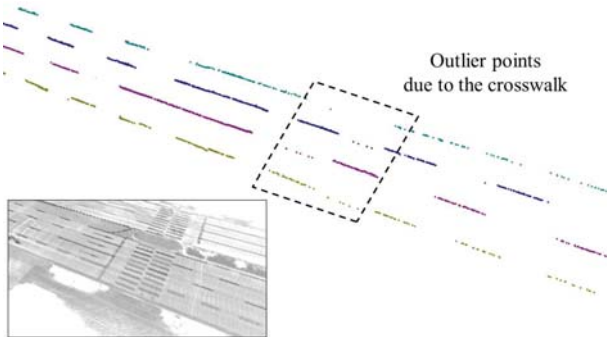


Fig. 5. Example of outlier points. The points corresponding to the crosswalk are extracted as the lane marking points since they are well-aligned with the lane markings, so exist in the extraction region.

marking points for extracting the points that correspond to the centerlines of the lane markings.

5) Sorting: For the final process, the lane marking points within each cluster are sorted and sequentially indexed from the closest point to the farthest point with respect to the starting point of the road. However, applying the sorting algorithm to all of the points is time-consuming. Fortunately, the points were divided into multiple blocks in the lane marking point extraction process. Since the blocks are sequentially arranged from the starting point of the road, we can sort all of the points by sorting the points in each block separately and concatenating the blocks of sorted points. In this paper, the bubble sorting algorithm is used in a block. As a result of the data sorting, the geometry data of the road with respect to the road lines is represented as sets of sequential points as follows:

$$\begin{aligned} \mathcal{G} &= \{\mathcal{G}_i | i = 1, ..., N_L + 1\} \\ \mathcal{G}_i &= \{G_n = (X, Y, Z)_n | n = 1, ..., N_i\}, \end{aligned} \tag{4}$$

where $\mathcal{G}_i$ is a set of sequential lane marking points corresponding to the ith road line, N_L is the number of lanes, G_n is the nth lane marking point, and N_i is the total number of lane marking points in $\mathcal{G}_i$.

C. Outlier Problem

The MLS-based lane marking extraction algorithm is susceptible to outliers that can be extracted with true lane marking points. As shown in Fig. 5, the road markings that exist outside of the extraction region such as arrows are effectively excluded by the extraction algorithm. However, the road markings that exist inside the extraction region (road markings that are well aligned with the lane markings) are extracted with the lane markings. These outliers may prevent the extracted road geometry data points from accurately expressing the true geometry of the road. This problem is supplemented by the road modeling algorithm that will be proposed in a later section. The proposed road modeling algorithm was designed to accurately model the true geometry of the road even though there exist outliers in the data.

V. Road Modeling

The road geometry data $\mathcal{G}$ that is represented by a set of points is not suitable for use as a road map since it consumes a large amount of storage space and the road geometry information is difficult to extract from the data. In this section, we propose a road modeling algorithm that increases the storage efficiency and usability of a road map by representing each set of points, $\mathcal{G}_i$, corresponding to a road line, as a mathematical curve defined as

$$L(s) = \sum_{m=1}^{M} L_m(s), \tag{5}$$

where, the subscript i that refers to the index of the road line was omitted for convenience, and

$$\begin{aligned} L_m(s) &= \begin{bmatrix} X_m(s) \\ Y_m(s) \\ Z_m(s) \end{bmatrix} \\ &= \begin{bmatrix} \sum_{p=0}^{3} C_{Xp,m} \cdot (s - s_m)^p \\ \sum_{p=0}^{3} C_{Yp,m} \cdot (s - s_m)^p \\ \sum_{p=0}^{3} C_{Zp,m} \cdot (s - s_m)^p \end{bmatrix}, \text{for } s \in [s_m, s_{m+1}). \end{aligned} \tag{6}$$

The curve $L(s)$ is a cubic spline curve that is composed of a finite number of sequentially connected piecewise polynomial curves. Each piecewise polynomial curve is parameterized according to $s \in [s_i, s_{i+1}]$ and consists of three cubic polynomials that represent the X, Y and Z coordinates, respectively. The advantage of representing the road geometry data points as a spline curve is obvious since the number points can be expressed as only one parameter value and twelve coefficients of a piecewise polynomial, i.e., s_m, $C_{X0,m} \sim C_{X3,m}$, $C_{Y0,m} \sim C_{Y3,m}$ and $C_{Z0,m} \sim C_{Z3,m}$.

The problem lies in expressing the point data in the spline curve form, that is, how to determine the number of piecewise polynomial curves and coefficients of the curves. There are three main considerations for this problem: storage efficiency, accuracy and outlier handling. First, the number of piecewise polynomial curves that compose the spline curve should be minimized to maximize the storage efficiency. However, there is a tradeoff between the number of piecewise polynomials and the accuracy, and in general, as we use a greater number of polynomial curves, the spline curves can express the point data with higher accuracy. Therefore, the problem is to minimize the number of piecewise polynomial curves while satisfying predefined accuracy constraint. Meanwhile, outliers included in the point data negatively impact the accuracy of the curve approximation. Therefore, the goal of the road modeling system is to find the optimal M, the number of piecewise polynomial curves, $\{s_m | m = 1, ..., M\}$, the set of parameters that indicate the point where the piecewise polynomial curves are divided, and the coefficients of the curves that take the accuracy constraint and outlier points into account.

However, it is very difficult to solve this problem from a global optimization point of view since we need to consider the number of piecewise polynomials, the parameter points at which the curve is divided, and the coefficients of the polynomials altogether. In addition, the accuracy constraint and the outliers should also be considered. Therefore, we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

consider the original problem as a a combination of smaller problems that can be solved relatively easily, and we thus propose the *sequential approximation algorithm* to efficiently find the near-optimal solution. In the rest of this section, we describe the details of the sequential approximation algorithm.

A. Overview of the sequential approximation algorithm

As mentioned above, the basic idea to solve the global optimization problem is to convert the problem into a combination of multiple simpler problems. The problem conversion process is based on the intuition that minimizing the number of piecewise polynomials that model the overall road geometry data is equivalent to maximizing the number of data points that each piecewise polynomial can express. We thus propose the sequential approximation algorithm, which can be summarized as below:

i) A set of road geometry point, $\mathcal{G} = \{G_n = (X, Y, Z)_n | n = 1, ..., N\}$, is given. Note that index i of the point set $\mathcal{G}_i$ is omitted for convenience.
ii) An initial polynomial curve is approximated by using the first few points. Here, the starting point of the curve is fixed for the first data point.
iii) Starting from the second point, a new point is sequentially added at every step, and the coefficients of the polynomial curve are corrected by using a Kalman filter (KF)-based algorithm.
iv) Whenever the polynomial curve is corrected, the accuracy of the curve is verified.
v) If the accuracy is higher than a threshold, step iii) is repeated. If not, adding new data points stops, and the coefficients of the polynomial curve that do not violate the accuracy criterion are saved.
vi) The remaining data points are used to repeat, ii) - v) until the last data point is has been reached.

As can be seen from the above description of the algorithm, the overall algorithm is composed of multiple sequential approximation processes. In an approximation process, a piecewise polynomial curve is generated to expresses as many points as possible within the accuracy bound. This approach is used to sub-optimally determine the set of break parameters, $\{s_m | i = m, ..., M\}$, and the coefficients of the curves. In the rest of this section, a detailed explanation of the sequential approximation algorithm is presented. First, we describe the detailed curve approximation process, and then present the detailed curve transition criterion and algorithm. The overall procedure for the algorithm is described in Algorithm 1.

Algorithm 1: Sequential Approximation Algorithm

input : $\mathcal{G}_i = \{G_{i,n} = \{X, Y, Z\}_{i,n} | n = 1, ..., N_i\}$
output: $L_i = \{L_{i,m} | m = 1, ..., M_i\}$

$m \leftarrow 1$;
$k \leftarrow 1$;
$[C_{0,m}, x_{m,k|k}, P_{m,k|k}] \leftarrow$ `Initialization(`$\mathcal{G}_i$, k`)`;
$[x_{m,k+1|k}, P_{m,k+1|k}] \leftarrow$ `Prediction(`$x_{m,k|k}$, $P_{m,k|k}$`)`;
$s_k \leftarrow 0$

for $k = 2$ **to** N_i **do**
  $s_k \leftarrow s_{k-1}+$`DistanceBetweenPoints(`$G_{i,k-1}$, $G_{i,k}$`)`;
  if `OutlierCheck(`$x_{m,k+1|k}$, $P_{m,k+1|k}$, $G_{i,k}$`)` *false* **then**
    $[x_{m,k|k}, P_{m,k|k}] \leftarrow$ `Update(`$x_{m,k+1|k}$, $P_{m,k+1|k}$, u_k, $\mathcal{G}_{i,k}$`)`;
  else
    $x_{m,k|k} = x_{m,k|k-1}$;
    $P_{m,k|k} = P_{m,k|k-1}$;
  end
  if (`CurveTransitionTrigger(`$\mathcal{G}_i$, $C_{0,m}$, $x_{m,k|k}$`)`) or ($k = N_i$) **then**
    $k \leftarrow k - N_{\text{buff}}$;
    $L_{i,m} \leftarrow \{C_{0,m}, x_{m,k|k}\}$;
    $m \leftarrow m + 1$;
    $[C_{0,m}, x_{m,k|k}, P_{m,k|k}] \leftarrow$ `Initialization(`$\mathcal{G}_i$, k`)`;
  end
  $[x_{m,k+1|k}, P_{m,k+1|k}] \leftarrow$ `Prediction(`$x_{m,k|k}$, $P_{m,k|k}$`)`;
end

$L_i \leftarrow$ `ArcLengthParameterization(`L_i`)`;
/*$C_{0,m} = \{C_{X0,m}, C_{Y0,m}, C_{Z0,m}\}$ */

B. Approximation Process

We use a Kalman filter, a recursive Bayesian estimation approach to approximate the given points to a cubic polynomial curve. The data points are regarded as observations for the estimation system, and the coefficients of the polynomial curve are regarded as a system state that we want to estimate. The reason to use a stochastic method instead of deterministic methods, such as spline interpolations, is that the data points contain stochastic errors even though the data has been refined in the data acquisition and processing systems. In addition, outliers included in the data should also be handled, and with proper parameter settings, a Bayesian filter can be designed to be robust against such outliers.

1) System Model: The system state vector x_k and the measurement vector y_k are defined by

$$\begin{aligned} x_k &= (C_{X1,k} \sim C_{X3,k}, C_{Y1,k} \sim C_{Y3,k}, C_{Z1,k} \sim C_{Z3,k})^{\mathrm{T}}, \\ y_k &= (X_{y,k}, Y_{y,k}, Z_{y,k})^{\mathrm{T}}. \end{aligned} \tag{7}$$

Note that the constant terms of the polynomials are excluded from the system state vector to ensure the geometrical continuity between neighbouring piecewise curves. However, we do not regulate both C_1- and C_2-*continuity* since there can be points where the tangent or the curvature are discontinuous in real roads.

Based on the state and the measurement vector definitions, we establish linear prediction and measurement models that are suitable for Kalman filtering. The prediction model is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

defined as

$$\begin{aligned} & x_k = x_{k-1} + w_{k-1},\ w_{k-1} \sim N(0, Q_{k-1}) \\ \Rightarrow & \begin{bmatrix} \mathbf{C}_{X,k} \\ \mathbf{C}_{Y,k} \\ \mathbf{C}_{Z,k} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{X,k-1} \\ \mathbf{C}_{Y,k-1} \\ \mathbf{C}_{Z,k-1} \end{bmatrix} + w_{k-1}, \end{aligned} \tag{8}$$

and the measurement model is defined as

$$\begin{aligned} y_k &= H_k \cdot x_k + u_k + v_k,\ v_k \sim N(0, R_k) \\ y_k &= \begin{bmatrix} X_{y,k} \\ Y_{y,k} \\ Z_{y,k} \end{bmatrix} = \begin{bmatrix} C_{X0} + C_{X1,k} \cdot s_k + C_{X2,k} \cdot s_k^2 + C_{X3,k} \cdot s_k^3 \\ C_{Y0} + C_{Y1,k} \cdot s_k + C_{Y2,k} \cdot s_k^2 + C_{Y3,k} \cdot s_k^3 \\ C_{Z0} + C_{Y1,k} \cdot s_k + C_{Y2,k} \cdot s_k^2 + C_{Y3,k} \cdot s_k^3 \end{bmatrix} + v_k \\ &= \begin{bmatrix} s_k & s_k^2 & s_k^3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_k & s_k^2 & s_k^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s_k & s_k^2 & s_k^3 \end{bmatrix} \cdot x_k + \begin{bmatrix} C_{X0} \\ C_{Y0} \\ C_{Z0} \end{bmatrix} + v_k, \end{aligned} \tag{9}$$

where k denotes the iteration step of the Kalman filter, s_k is a system input denoting the parameter value of the polynomial curve, and w_k and v_k are the prediction and measurement noises with covariance Q_k and R_k, respectively. Under ideal conditions where the prediction noise is ignored, the system state does not change during the prediction step. This is reasonable since the coefficients of the polynomial curve are only affected by the data points that are observed. The coefficients should be adjusted only when a new data point has been added.

Measurement matrix H_k is a function of the time-varying system input s_k. Ideally, s_k is an arc-length parameter of the spline curve that corresponds to the measurement point y_k. However, since it is impossible to calculate the arc-length of the curve before finishing the approximation process, we approximate the curve by using arbitrary, strictly-increasing parameters first, and then we re-parameterize the curve by the arc-length after finishing the approximation process. To simplify the re-parameterization process, we use the chord-length from the initial point as the parameter value, s_k, which is an approximated value of the arc-length.

2) Observability Proof: The suitability of the proposed system model can be proven by checking the observability of the system. Observability is a necessary condition for the Kalman filter to work correctly, and we say that a linear system is observable if there is a finite number of steps n so that knowledge about the input sequence $u_0, ..., u_{n-1}$ and the output sequence $y_0, ..., y_{n-1}$ is sufficient to determine the initial state of the system, x_0. Therefore, the system is observable if x_0 is uniquely determined by

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \end{bmatrix} = O_n x_0 + T_n \begin{bmatrix} u_0 \\ \vdots \\ u_{n-1} \end{bmatrix}, \tag{10}$$

where O_n is a matrix that maps the initial state x_0 into the resulting output over $[0, n-1]$ and T_n is a matrix that maps the input to the output over $[0, n-1]$. For our system, the above problem can be represented as

$$\begin{bmatrix} y_0 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} H_0 \\ \vdots \\ H_{n-1} \end{bmatrix} x_0 + \begin{bmatrix} C_0 \\ \vdots \\ C_0 \end{bmatrix}, \tag{11}$$

where $C_0 = [C_{X0}, C_{Y0}, C_{Z0}]^{\mathrm{T}}$. For $n = 3$, the initial state $x(0)$ is uniquely determined if and only if $N(O_3) = 0$ and, equivalently, $\mathrm{Rank}(O_3) = 9$. The matrix O_3 is composed of three observation matrices: H_0, H_1 and H_2. It can be easily proved that $\mathrm{Rank}(O_n) = 9$. By definition, since $s_k > s_{k-1}$ for every k, every row vector for matrix O_n is independent relative to the other row vectors. Therefore, we can prove that the proposed system is observable.

3) Error Covariance: In order to obtain good results from the Kalman filter, the proper error covariance matrices, Q_k and R_k, need to be set up. First, the prediction error should be zero (and thus $Q_k = \mathbf{0}$) for this problem since the statistics of the state values do not vary during the prediction step.

The measurement error covariance matrix, R_k, is determined according to the statistics of the vehicle pose data and the MLS data. For example,

$$R_k = \begin{bmatrix} 0.05^2 & 0 & 0 \\ 0 & 0.05^2 & 0 \\ 0 & 0 & 0.05^2 \end{bmatrix}. \tag{12}$$

4) Processing Outlier Points: The Kalman filter has the inherent ability to handle outliers by allocating small gains to measurements that are far from the predicted system state. However, outliers that are too large or too frequent may force the Kalman filter to diverge. To address this problem, outliers are detected and isolated so that they are not used for state update, by applying a normalized innovation squared (NIS) test to every measurement point before updating the system state using the measurement point. The NIS test is defined as follows:

$$\mathrm{NIS}_k = (y_k - H_k \hat{x}_{k|k-1})^T S_k^{-1} (y_k - H_k \hat{x}_{k|k-1}) > \lambda, \tag{13}$$

where

$$S_k = H_k P_{k|k-1} H_k^T + R_k. \tag{14}$$

The measurement point that is larger than a threshold λ is considered to be an outlier, and it is discarded. The threshold value λ should be carefully determined since a value that is too large for λ negatively impacts the ability to detect for an outlier, while the Kalman filter may diverge with a too small value for λ due to frequent removal of normal measurement points.

C. Curve Transition

Based on the proposed system and measurement model, covariance matrices, and outlier point process algorithm, the KF approximates the road geometry point data that are given one by one to a polynomial curve. At a certain point, the KF should stop the current approximation and should start to generate a new polynomial curve since one cubic polynomial curve cannot express all points. The condition and procedure for the curve transition are discussed below.

At the end of every step of the KF, two kinds of maximum distance between the curve and the data points are calculated in order to decide whether the new curve approximation should be started, i.e.,

$$\begin{aligned} D_{\max,XY} &= \max_i \left\| L_{XY}(\bar{s}_i) - G_{XY,i} \right\|,\ i = 1, ..., k, \\ D_{\max,Z} &= \max_i \left\| L_Z(\bar{s}_i) - G_{Z,i} \right\|,\ i = 1, ...k, \end{aligned} \tag{15}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

where

$$L_{XY}(s) = \begin{bmatrix} C_{X0} + C_{X1} \cdot s + C_{X2} \cdot s^2 + C_{X3} \cdot s^3 \\ C_{Y0} + C_{Y1} \cdot s + C_{Y2} \cdot s^2 + C_{Y3} \cdot s^3 \end{bmatrix} \quad (16)$$

and

$$L_Z(s) = C_{Z0} + C_{Z1} \cdot s + C_{Z2} \cdot s^2 + C_{Z3} \cdot s^3 \quad (17)$$

are the polynomial curve on the XY- and Z-planes, respectively, while $G_{XY,i}$ and $G_{Z,i}$ refer to the i-th data point on the XY- and Z-plane, respectively. Therefore, $D_{\max,XY}$ and $D_{\max,Z}$ refer to the maximum distance errors between the curve and the data points on the XY- and Z-plane, respectively. The reason to calculate the distance error on the XY- and Z-plane separately is that the accuracy requirements for the XY- and Z-plane can be different. In general, autonomous vehicle systems require accuracy at the centimeter-level in the XY-plane since the map accuracy in the XY-plane affects vehicle safety in applications that involve vehicle localization or vehicle motion planning. On the other hand, a relatively low accuracy is required for the Z-plane since generally the height information of the road is generally used in non-safety-related applications such as for fuel management systems.

Ideally, the curve transition should be conducted at the moment when at least one of the two distance error exceeds a pre-defined threshold, i.e.,

$$D_{\max,XY} > \chi_{XY} \text{ or } D_{\max,Z} > \chi_Z, \quad (18)$$

However, the maximum distance error can occasionally exceed the threshold even though all data points belong to one polynomial curve due to stochastic errors that are included in the data. In this case, the errors have to be ignored, and the KF should be continued. For this reason, a heuristic test is applied for the curve transition as follows:

i) A counter is initialized to zero before starting the KF.
ii) At the end of the every step of the KF, the maximum distance errors, $D_{\max,XY}$ and $D_{\max,Z}$, are calculated.
iii) The counter increases if $D_{\max,XY} > \chi_{XY}$ or $D_{\max,\mathrm{Z}} > \chi_Z$ and it is set to zero otherwise.
iv) If the counter exceeds a threshold N_{buff}, the curve transition is triggered.

In summary, the curve transition is triggered when the maximum distance errors exceed the threshold N_{err} times in a row. Once the curve transition is triggered at step k, the curve approximated at step $k - N_{\text{buff}}$ is saved since the curve approximated from step $k - N_{\text{buff}}$ to k is inaccurate. Therefore, state vectors and covariance matrices for the KF for recent N_{buff} steps should be stored in the buffer, and after the current polynomial curve is saved, the KF is initialized and a new curve approximation process begins.

In addition to the normal case, there is another case where the curve transition is needed. In a road, the number of lanes can vary, as shown in Fig. 7. In this case, the curve approximation process has to be stopped at the point where the road line disappears, and a new curve has to be started from the point where the road line reappears. In order to handle this case, at every step of the approximation process, the distance between the current data point and the next point is calculated. If the distance is larger than a threshold, i.e.,

$$\|G_{i+1} - G_i\| > \eta, \quad (19)$$

which means that the road line disappears at the current point, the curve transition is triggered. Since the new curve is discontinuous with the previous curve, the curve parameter representing the arc-length is set to zero at the start point of the new curve.

Fig. 6. Probe vehicle equipped with a 3D Lidar and a GPS+INS positioning system.

D. Arc length parameterization

The remaining portion of the procedure for road modeling involves arc-length parameterization. As mentioned in the previous sections, the curve generated from the sequential approximation algorithm is parameterized according to the chord length as an approximation of the arc-length. However, since accurate arc-length information is useful for many intelligent vehicle applications, we parameterize the curve in terms of the arc-length as the final task for the road modeling. Many studies have investigated the arc-length parameterization task for a curve, and in this paper, we use the algorithm proposed in [49] since it can process a large quantity of map data in a simple and efficient manner. The arc-length parameterization of a curve can be constructed by adhering to the following two-step process: After conducting the arc length parametrization, we obtain the final arc length-parameterized cubic spline curve for the data points of the road geometry.

VI. Experimental Validation

In this section, we provide the results of the experiment that was carried out to evaluate the performance of the proposed road map generation system. First, we introduce the probe vehicle and sensor configuration used in the experiments. We then present the results of 3D Lidar-based data acquisition and data processing. We also evaluate the accuracy, storage efficiency and usability performance of the road modeling system. Finally, we provide an autonomous driving test based on the proposed road map to validate the practicality of the proposed road map system.

A. Experimental Setup

The raw data required to evaluate the proposed road map generation system was collected using a probe vehicle equipped with a 3D Lidar (Velodyne HDL-64E) and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

TABLE II
SPECIFICATION OF THE 3D LIDAR.

Horizontal field field of view	Vertical field of view	Distance accuracy	Measurement rate
360 deg.	26.8 deg.	$<$2 cm	$>$1.3 M points/sec.

TABLE III
SPECIFICATION OF THE VEHICLE POSITIONING SYSTEM.

Measurement	Frequency	Accuracy (RMS)
Position	100 Hz	2 cm
Yaw	100 Hz	0.1 deg.
Roll/pitch	100 Hz	0.03 deg.

a GPS+INS vehicle positioning system (OXTS RT3002) as shown in Fig. 6. The Velodyne HDL-64E is a 64-layer Lidar, and it was configured to rotate 360 degrees to emit 64-layer laser beams with a minimum of 1.3 million points measured per second. OXTS RT3002 is a unified GPS+INS system that provides a highly accurate 6D vehicle poses by combining the GPS and INS data. The specifications of these two measurement system are summarized in Tables II and III, respectively. In this paper, raw data was collected with a 10 Hz rate, and the prove vehicle was driven at a normal driving speed of 30 km/h to 80 km/h depending on the the road type and conditions. All of the experiments were carried out on a PC with a 3.40 GHz i7-4770 CPU. The parameters χ_{XY} and χ_Z in Eqs. (18) were set to 0.1 m and 0.3 m respectively, and η in Eqs. (19) was set to 10m.

B. Data Acquisition and Processing

Two different data sets were collected from the Incheon International Airport Express (IIAE) and Seoul National University (SNU) ring road, respectively. The IIAE is a *type I* road that has four lanes, and thus is suitable to evaluate the road map generation system including the MLS-based data acquisition system. The IIAE data set covers a road length of approximately 13 km and contains about 5,900 frames of synchronized 3D MLS (7.7 billion points) and 6D vehicle pose data. Fig. 7 shows a qualitative result of the data processing algorithm when applied to the IIAE data set. From total 7.7 billion laser points, about 65,000 were extracted after executing the data processing algorithm. It is impossible to present the quantitative accuracy of the data acquisition and processing result, since there is no way to measure more accurate coordinates of the lane markings than the RTK-GPS + 3D Lidar configuration that was used in this paper. The data processing operates in off-line mode. In the experiment, it took about 38 seconds to process 1 km road of the IIAE data set, where the algorithm was implemented based on the point cloud library [50].

The SNU ring road is a *type II* road and is more appropriate to evaluate the road modeling system since it contains various curves and height variations. The SNU data set consists of a road with a distance of approximately 3.7 km and contains about 6,900 vehicle pose data in intervals of about 0.5 m.

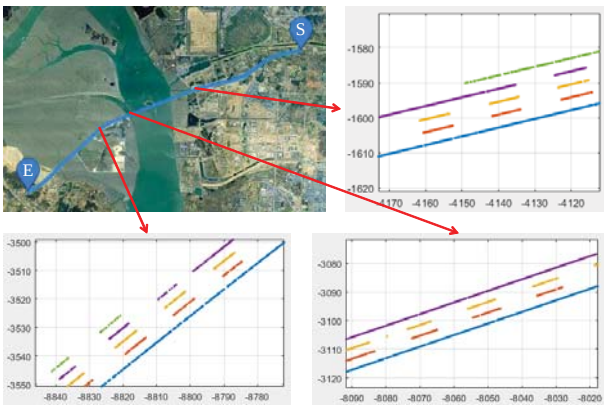


Fig. 7. Data processing result for the IIAE data set. Both the solid and dashed lane markings were extracted clearly, and the variation of the number of lanes is handled properly.

C. Road Modeling

Both the IIAE and SNU data sets were modeled using spline curves by using the proposed road modeling algorithm. Figs. 8 and Fig. 9 show the results of road modeling where the XY-plane and Z-axis accuracy thresholds, $D_{\max,\mathrm{XY}}$ and $D_{\max,\mathrm{Z}}$, were set to 0.1 m and 0.3 m, respectively. Since the IIAE data set contains four lanes, a total of five spline curves were generated, whereas only one spline curve was generated for the SNU data set. The dots in the figures denote the point data, and the lines denote the approximated spline curve. The circles indicate the break points of the spline curves. The road modeling process operates in off-line mode. In the experiment, it took about seven seconds to process 1 km road of the IIAE data set.

To quantitatively evaluate the performance of the proposed road modeling system in terms of the accuracy, storage efficiency and usability, we conducted a comparison with a B-spline-based state-of-the-art algorithm that was proposed in [16]. The B-spline-based algorithm had been proven to outperform various previous algorithms, including algorithms using polygons, natural cubic splines, or cubic B-splines with a constant interval.

1) Accuracy: In terms of the accuracy, a major goal of the road modeling system is to generate a curve model for the distance errors between the road geometry data points and the curve not to exceed a preset tolerance, e.g., 0.1 m. Fig. 10 shows the modeling accuracy of the proposed algorithm and the B-spline-based algorithm when the SNU data set is used with 0.1m XY-plane accuracy tolerance. The horizontal axis of the figure refers to each data point and the vertical axis refers to the distance error between the data points and the generated curve. Note that two different results are presented for the proposed algorithm depending on whether or not the outlier rejection scheme was used. In the case where the outlier rejection was used, six points were found to violate the 0.1m tolerance. However, exceeding the tolerance does not necessarily mean that the accuracy is degraded but rather that the outliers were detected and rejected by the algorithm. We found that all points that violated the tolerance coincide with outliers that had been detected during the road modeling

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

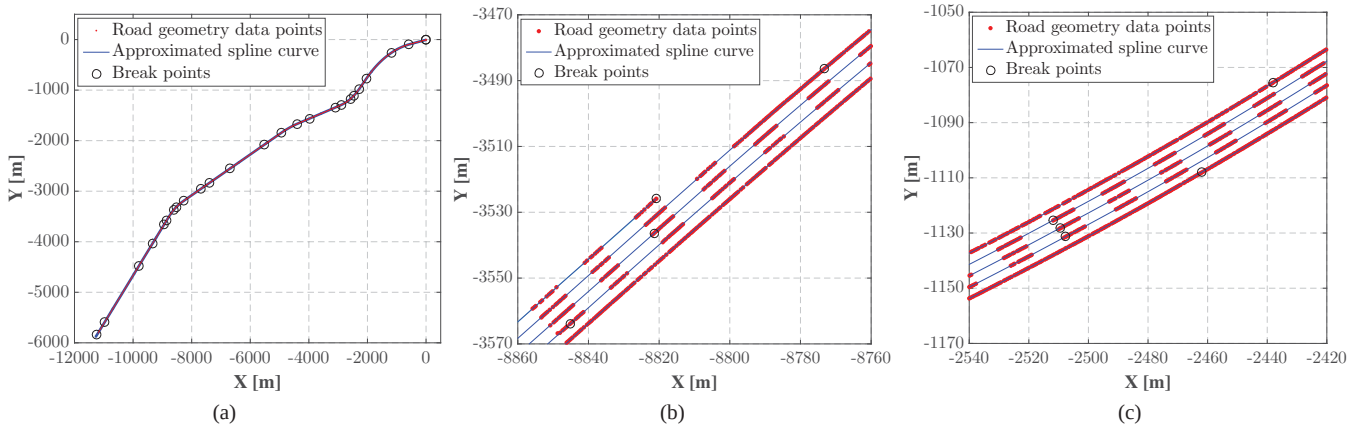


Fig. 8. The result of the road modeling for the IIAE data set. (a) Overall map, and (b) and (c) enlarged maps.

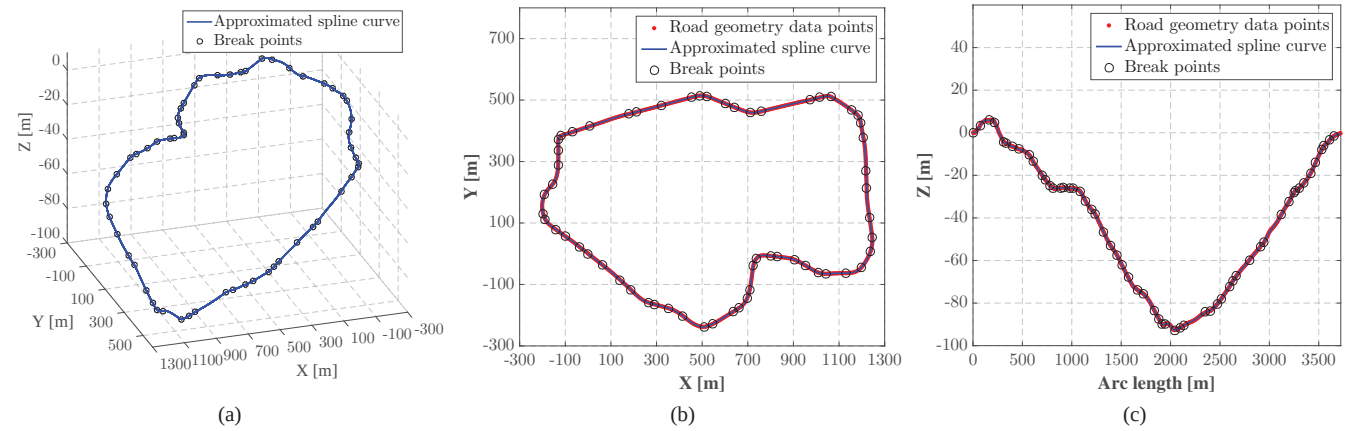


Fig. 9. The result of the road modeling for the SNU data set. (a) 3D road map, (b) road map projected on XY plane, and (c) height of the road.

process, as the example in Fig. 11 shows. The outlier rejection affects the storage efficiency of the map, and in this example, the entire road was modeled with 55 piecewise polynomials in the case where outliers were rejected while a total of 67 piecewise polynomials were used in the case where outliers were not rejected. As a consequence, we can see that the proposed road modeling algorithm successfully detects and rejects outliers that are contained in data points and also accurately and efficiently represents data points.

Accurate coordinates as well as accurate tangent angle and curvature information is very important for various intelligent vehicle applications. Such information is essential for safety-related systems, including motion planning systems. Adhering to the tolerance requirements does not necessarily ensure the accuracy of the tangent angle and curvature, and therefore, we also evaluated the road modeling algorithm in terms of the accuracy of the tangent angle and curvature. It is difficult to evaluate the accuracy of the tangent angle and curvature by using real road geometry data points since it is impossible to know the exact tangent angle and curvature values of the real road. Thus, we generated artificial road geometry data points for which we know the exact tangent angle and curvature values by first generating an artificial spline curve, shown in Fig. 12(a), and then we sampled the curve at every 1 m. To ensure that the data is close to the real values, Gaussian random noise with a standard deviation of 0.05m was added to all points. The artificial road geometry data was approximated with a set of polynomials that make use of the proposed road modeling algorithm, and the geometry information that is extracted from the curve was compared to the true value of the road geometry. Fig. 12 presents the results, and we can see that the road model accurately expresses all the true road geometries.

A quantitative evaluation was also carried out. Table IV shows the accuracy of the tangent angle and curvature of the road models that were generated using the proposed algorithm and B-spline-based algorithm, where the tolerance of the algorithm was set to 0.1 m. The results show that the proposed algorithm generated a more accurate road model than the B-spline-based algorithm. Furthermore, the levels of maximum error in the tangent angle and curvature were 10^{-1} and 10^{-3} respectively. In general, such levels are acceptable for the errors in intelligent and autonomous vehicle applications.

2) Storage Efficiency and Usability: The storage efficiency and usability were evaluated together. First, the SNU data set was modeled using the proposed algorithm and the B-spline-based algorithm to compare the storage efficiency and usability of the two models. In this paper, the storage efficiency is measured by the number of floating-point numbers that are used to express the road model. In the case with the proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

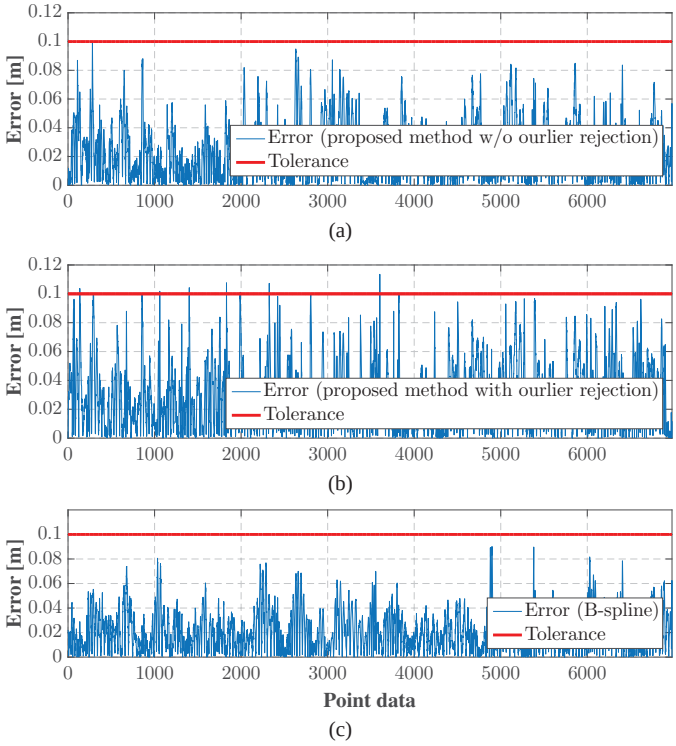


Fig. 10. Errors between road geometry data points and road models generated by (a) proposed algorithm without outlier rejection, (b) proposed algorithm with outlier rejection, and (c) B-spline-based algorithm. Both methods satisfy the 0.1 m accuracy constraint. Note that the error violations in subfigure (b) do not mean the accuracy degradation. Instead, it means that the outliers were successfully detected and rejected.

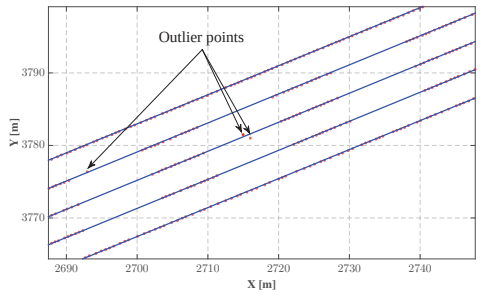


Fig. 11. Example of the outlier rejection which shows that the outliers were effectively detected and rejected.

TABLE IV
ACCURACY OF TANGENT ANGLE AND CURVATURE OF THE PROPOSED ROAD MODEL AND THE B-SPLINE-BASED ROAD MODEL.

		Mean	Std.	RMS	Max
Tangent angle error [deg.]	Proposed	0.898E-02	1.88E-02	1.59E-02	2.67E-01
	B-spline	3.54E-02	3.86E-02	5.24E-02	3.13E-01
Curvature error [m^{-1}]	Proposed	1.67E-05	0.682E-04	0.496E-04	0.841E-03
	B-spline	7.98E-05	1.23E-04	1.47E-04	1.24E-03

set of piecewise polynomial forms, a piecewise polynomial requires thirteen floating-point numbers (one beginning parameter and twelve polynomial coefficients). Therefore, the set of piecewise polynomial forms requires totally 13×[*number of piecewise polynomials*] floating-point numbers. The number of required floating-point numbers for the B-spline form depends on the number of knots and control points. Since

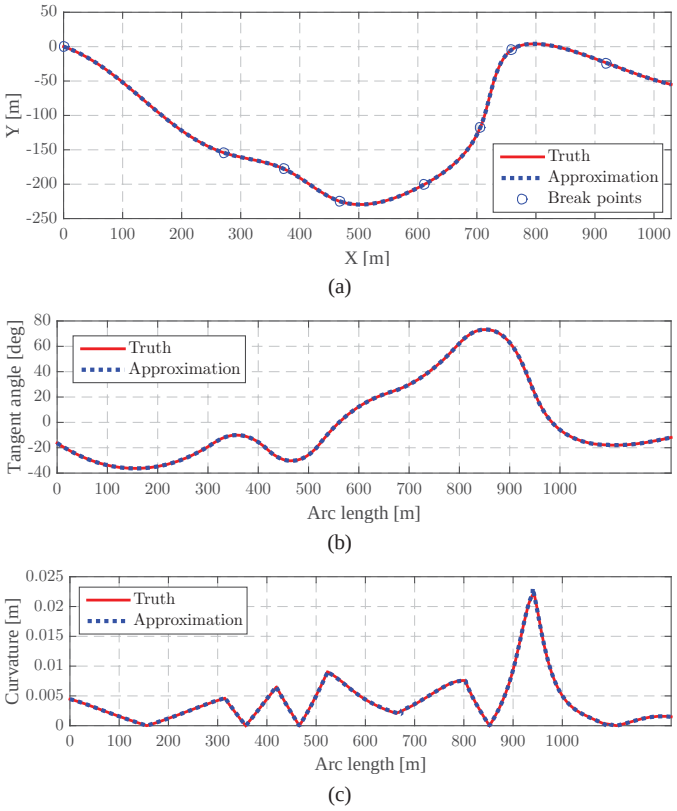


Fig. 12. Qualitative result of road modeling in terms of (a) coordinates, (b) tangent angle, and (c) curvature.

each control point contains three floating point numbers (X,Y and Z coordinates), the B-spline requires a total of [number of knots]+3×[*number of control points*] floating-point numbers.

The usability of such a system for intelligent and autonomous vehicles is inversely proportional to the computational time required to calculate the essential information from the road models. Four essential pieces of information were selected: position, tangent angle and curvature at an arbitrary point on a curve, and closest point on a curve from an external point. The position, tangent angle and curvature information can be analytically calculated using both the proposed and the B-spline curves. However, the B-spline curve requires complex recursive calculations to obtain this information [42], while the proposed method only requires simple calculations since the first and second derivative of the curve are easily calculated from the polynomials. The closet point on a curve should be calculated using a numerical method for both curves. In this paper, we used Newton's method to evaluate both curves on equal terms (with 10^{-6} tolerance), and the experiments were carried out on a PC with a 3.40 GHz i7-4770 CPU.

Table V shows the results of the experiment. Note that the B-spline was evaluated using two different modes. In the normal mode, only the knots and control points of the B-spline curve itself were stored in the memory. On the other hand, in the second mode, the knots and control points for the first and second derivatives of the B-spline curve were precomputed and additionally stored in the memory. The precomputation of the derivatives increases the storage space that is required, however, it decreases the computational time. The results

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2016.2535210, IEEE Transactions on Vehicular Technology

TABLE V
STORAGE EFFICIENCY AND USABILITY OF THE PROPOSED ROAD MODEL AND THE B-SPLINE-BASED ROAD MODEL.

	# of segments	# of floating-point numbers	Computational time (μs)			
			Position	Tangent angle	Curvature	Closest point
Proposed	55	715	0.68	0.47	0.59	16
B-spline	178	728	32	82	146	630
B-spline (precomputed derivatives)	178	1805	32	29	52	280

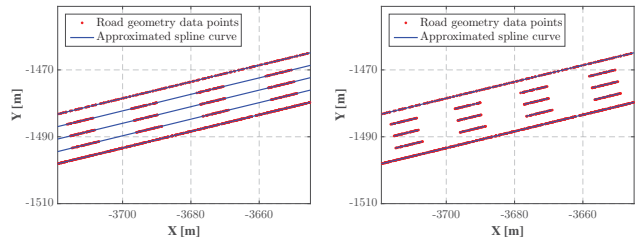


Fig. 13. (a) The original road map. (b) The augmented road map that represent the dashed road lines.

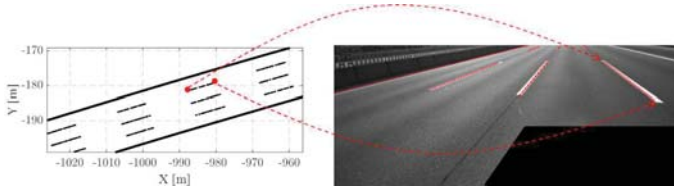


Fig. 14. Transformation of the local map data to image plane. This example show a case where the map data and the image is matched well.

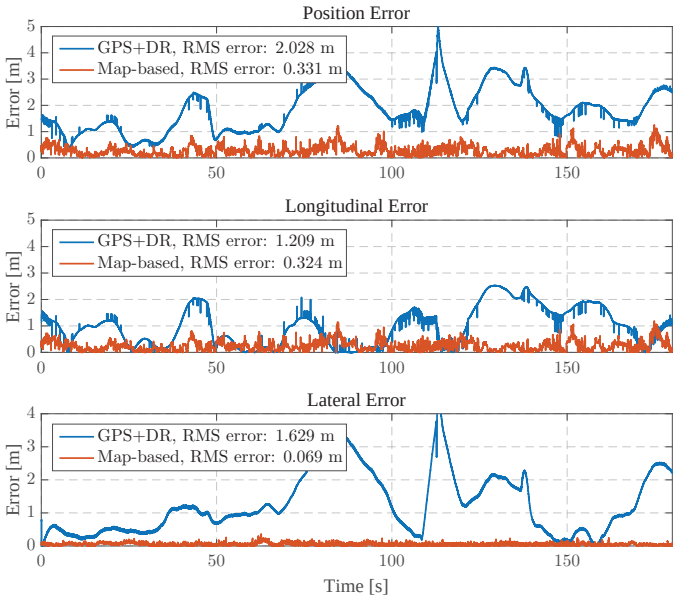


Fig. 15. The localization errors of the lane-level map-based localization system and the conventional GPS/DR-based localization system.

show that the proposed algorithm outperforms the B-spline-based algorithm in both modes in terms of both the storage efficiency and usability. In particular, the proposed road model dramatically decreases the computational time needed to calculate the various pieces of information for intelligent and autonomous vehicles. In terms of storage efficiency, note that the B-spline-based algorithm requires a total of 178 curve segments, whereas the proposed algorithm requires only 55 curve segments. This means that, in the case in which the curve is first approximated as a B-spline curve and converted to the piecewise-polynomial form, a large amount of storage is required than when using the proposed algorithm.

D. Practicality of the Proposed Road Map: Map-Aided Vehicle Localization

Vehicle localization is one of the primary applications of the precise lane-level road map. In order to show the feasibility and practicality of the proposed road map, we conducted a vehicle localization experiment using the map. First, in order to utilize the road map more effectively, we somewhat augmented the map so that it can represent the dashed road lines as shown in Fig. 13. This was accomplished by finding the curve parameters that correspond to the start and end points of the dashed lane markings using the road geometry point data presented in Section IV, and the corresponding parameter values were stored with the coefficients of the polynomial curves. Using this method, the road map can represent the dashed road lines efficiently.

We designed a localization system using the augmented road map and some affordable sensors, such as a low-cost GPS, in-vehicle network sensors (yaw rate and wheel speed sensors), and a monocular camera. Since the goal of this section is to show the practicality of the proposed road map, we focus on showing how the map is used for accurate vehicle localization rather than explaining the details of the localization algorithm. A detailed localization algorithm will be presented in our future paper.

The core of the localization system is feature matching between the given road map and the lane marking detection result from the monocular camera image. Whenever an image is updated from the camera, the road map data around the vehicle is transformed to the image plane with respect to some candidates for true vehicle position. Then, a position candidate for which the transformed map data and lane markings are well matched is selected as the estimated value of the vehicle position (see Fig. 14).

The performance of the vehicle localization system based on the proposed road map was evaluated through an experiment conducted on about a 5 km section of the IIAE. Fig. 15 shows the localization errors of the localization system using the proposed road map and the conventional GPS/DR-based localization system [51]. The result shows that the proposed

road map-based localization system provides sufficient longitudinal localization accuracy for vehicle route guidance, and sufficient lateral localization accuracy for lateral vehicle control of autonomous vehicles.

VII. Conclusion

In this paper, we have presented a road map generation system that simultaneously considers the accuracy, storage efficiency and usability to generate high-precision lane-level road maps for use in intelligent vehicle systems. The overall system is composed of three subsystems, including data acquisition, data processing and road modeling systems. The MLS data-based data acquisition and processing system efficiently acquires accurate road geometry data by integrating 6D vehicle pose and 3D MLS data. The main contribution of this paper is to present a road modeling algorithm that models road geometry data as a form consisting of sets of piecewise polynomial curves that are more appropriate for use in intelligent vehicle systems than those of previous road models, such as clothoid or B-splines in terms of usability. The proposed road modeling algorithm maximizes the storage efficiency of the map by minimizing the number of piecewise polynomials needed to express the map. The experiments were conducted to demonstrate that the map that is generated by the proposed algorithm is more efficient than those obtained using previous algorithms in terms of the storage efficiency and usability.

References

[1] R. Bishop, *Intelligent vehicle technology and trends*. Boston, Mass. : Artech House, 2005.

[2] O. Gusikhin, D. Filev, and N. Rychtyckyj, "Intelligent vehicle systems:applications and new trends," in *Informatics in Control Automation and Robotics*, ser. Lecture Notes Electrical Engineering, J. Cetto, J.-L. Ferrier, J. M. Costa dias Pereira, and J. Filipe, Eds. Springer Berlin Heidelberg, 2008, vol. 15, pp. 3–14.

[3] S.-W. Kim, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "The impact of cooperative perception on decision making and planning of autonomous vehicles," *Intelligent Transportation Systems Magazine, IEEE*, vol. 7, no. 3, pp. 39–50, 2015.

[4] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[5] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 4, pp. 1599–1616, Dec 2012.

[6] J. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro, "Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 438–450, Sept 2008.

[7] R. Toledo-Moreo, D. Bétaille, F. Peyret, and J. Laneurit, "Fusing GNSS, dead-reckoning, and enhanced maps for road vehicle lane-level navigation," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 3, no. 5, pp. 798–809, Oct 2009.

[8] R. Toledo-Moreo, D. Bétaille, and F. Peyret, "Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 100–112, March 2010.

[9] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman, "Mapping and localization using gps, lane markings and proprioceptive sensors," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 406–412.

[10] N. Suganuma and T. Uozumi, "Precise position estimation of autonomous vehicle based on map-matching," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, June 2011, pp. 296–301.

[11] M. Schreiber, C. Knoppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 449–454.

[12] A. Schindler, "Vehicle self-localization with high-precision digital maps," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 141–146.

[13] E. Consortium *et al.*, "Enhanced digital mapping project. final report." EDMap Project eSafety Forum.(2005). Digital maps Working Group Final Report. European Commission (eSafety Forum), Brussels, 2004.

[14] D. Bétaille and R. Toledo-Moreo, "Creating enhanced maps for lane-level vehicle navigation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 4, pp. 786–798, Dec 2010.

[15] B. Schwarz, "Lidar: Mapping the world in 3D," *Nature Photonics*, vol. 4, no. 7, pp. 429–430, 2010.

[16] K. Jo and M. Sunwoo, "Generation of a precise roadway map for autonomous cars," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 3, pp. 925–937, June 2014.

[17] K. Ichida, T. Kiyono, and F. Yoshimoto, "Curve fitting by a one-pass method with a piecewise cubic polynomial," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 2, pp. 164–174, 1977.

[18] M. Plass and M. Stone, "Curve-fitting with piecewise parametric cubics," in *ACM SIGGRAPH Computer Graphics*, vol. 17, no. 3. ACM, 1983, pp. 229–239.

[19] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *Pervasive Computing, IEEE*, vol. 7, no. 4, pp. 12–18, 2008.

[20] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller *et al.*, "Making bertha drive-an autonomous journey on a historic route," *Intelligent Transportation Systems Magazine, IEEE*, vol. 6, no. 2, pp. 8–20, 2014.

[21] A. Joshi and M. R. James, "Generation of accurate lane-level maps from coarse prior maps and lidar," *Intelligent Transportation Systems Magazine, IEEE*, vol. 7, no. 1, pp. 19–29, 2015.

[22] Y.-W. Seo, C. Urmson, and D. Wettergreen, "Exploiting publicly available cartographic resources for aerial image analysis," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '12. New York, NY, USA: ACM, 2012, pp. 109–118.

[23] J. Hu, A. Razdan, J. Femiani, M. Cui, and P. Wonka, "Road network extraction and intersection detection from aerial images by tracking road footprints," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 45, no. 12, pp. 4144–4157, Dec 2007.

[24] M. Amo, F. Martinez, and M. Torre, "Road extraction from aerial images using a region competition algorithm," *Image Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1192–1201, May 2006.

[25] S. Rogers, "Creating and evaluating highly accurate maps with probe vehicles," in *Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE*, 2000, pp. 125–130.

[26] A. Schindler, G. Maier, and F. Janda, "Generation of high precision digital maps using circular arc splines," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June 2012, pp. 246–251.

[27] S. Schroedl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson, "Mining GPS traces for map refinement," *Data mining and knowledge Discovery*, vol. 9, no. 1, pp. 59–87, 2004.

[28] F. Caron, E. Duflos, D. Pomorski, and P. Vanheeghe, "GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects," *Information Fusion*, vol. 7, no. 2, pp. 221 – 230, 2006.

[29] H. Qi and J. Moore, "Direct Kalman filtering approach for GPS/INS integration," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 38, no. 2, pp. 687–693, Apr 2002.

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

[31] 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.

[32] A. Mancini, E. Frontoni, and P. Zingaretti, "Automatic road object extraction from mobile mapping systems," in *Mechatronics and Embedded Systems and Applications (MESA), 2012 IEEE/ASME International Conference on*. IEEE, 2012, pp. 281–286.

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

[34] S. Brummer, F. Janda, G. Maier, and A. Schindler, "Evaluation of a mapping strategy based on smooth arc splines for different road types," in *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, Oct 2013, pp. 160–165.

[35] A. Schindler, G. Maier, and S. Pangerl, "Exploiting arc splines for digital maps," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct 2011, pp. 1–6.

[36] J. Goldbeck and B. Huertgen, "Lane detection and tracking by video sensors," in *Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on*, 1999, pp. 74–79.

[37] E. Safra, Y. Kanza, Y. Sagiv, and Y. Doytsher, "Efficient integration of road maps," in *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*. ACM, 2006, pp. 59–66.

[38] A. Vatavu, R. Danescu, and S. Nedevschi, "Environment perception using dynamic polylines and particle based occupancy grids," in *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 239–244.

[39] S. Sivaraman and M. M. Trivedi, "Dynamic probabilistic drivability maps for lane change and merge driver assistance," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 15, no. 5, pp. 2063–2073, 2014.

[40] V. Gikas and J. Stratakos, "A novel geodetic engineering method for accurate and automated road/railway centerline geometry extraction based on the bearing diagram and fractal behavior," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 115–126, March 2012.

[41] H. Park and J.-H. Lee, "B-spline curve fitting based on adaptive curve refinement using dominant points," *Computer-Aided Design*, vol. 39, no. 6, pp. 439 – 451, 2007.

[42] P. Sherar, "Variational based analysis and modelling using B-splines," Ph.D. dissertation, Cranfield University, 2004.

[43] D.-S. Kim, J. Ryu, H.-C. Lee, and H. Shin, "The conversion of a dynamic b-spline curve into piecewise polynomials in power form," *Computer-Aided Design*, vol. 34, no. 4, pp. 337–345, 2002.

[44] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, p. 0278364913491297, 2013.

[45] L. Soomok, K. Seong-Woo, , and S. Seung-Woo, "Accurate ego-lane recognition utilizing multiple road characteristics in a bayesian network framework," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*, June 2015, pp. 543–548.

[46] S.-N. Kang, S. Lee, J. Hur, and S.-W. Seo, "Multi-lane detection based on accurate geometric lane estimation in highway scenarios," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 221–226.

[47] J. Hur, S.-N. Kang, and S.-W. Seo, "Multi-lane detection in urban driving environments using conditional random fields," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1297–1302.

[48] Y. Zhou and A. Toga, "Efficient skeletonization of volumetric objects," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 5, no. 3, pp. 196–209, Jul 1999.

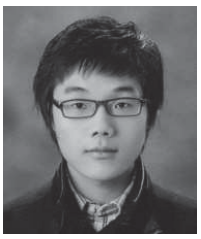
[49] H. Wang, J. Kearney, and K. Atkinson, "Arc-length parameterized spline curves for real-time simulation," *Curve and surface design: Saint-Malo 2002.*, pp. 387–396, 2003.

[50] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Point cloud library," *IEEE Robotics & Automation Magazine*, vol. 1070, no. 9932/12, 2012.

[51] W.-W. Kao, "Integration of gps and dead-reckoning navigation systems," in *Vehicle Navigation and Information Systems Conference, 1991*, vol. 2. IEEE, 1991, pp. 635–643.

Gi-Poong Gwon received the B.S. degree in information and communication engineering from Sungkyunkwan University, Suwon, Korea, in 2009, and the Ph.D. degree in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 2016.

He was a Researcher with the Intelligent Vehicle IT (IVIT) Research Center funded by the Korean Government and Automotive Industries. He is now a Senior Researcher at LG Electronics, Inc., Korea. His current research areas include vehicular electronics for intelligent vehicles, and mapping and localization for automated vehicles.



Woo-Sol Hur received the B.S. degree in electrical engineering from Pohang University of Science and Technology, Pohang, Korea, in 2013. He is currently working toward the Ph.D. degree in the Department of Electrical Engineering and Computer Science, Seoul National University, Korea.

He is a Researcher with the Intelligent Vehicle IT (IVIT) Research Center, Korea, funded by the Korean Government and Automotive Industries. His current research areas include 3D map construction and maintenance for intelligent vehicles.

Seong-Woo Kim (M'11) received the B.S. and M.S. degrees in electronics engineering from Korea University, Seoul, Korea, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering and computer science from Seoul National University in 2011. He was a postdoctoral associate with the Singapore-MIT Alliance for Research and Technology. In 2014, he joined the Seoul National University, where he is currently a Research Assistant Professor.

Dr. Kim received the Best Student Paper Award at the 11th IEEE International Symposium on Consumer Electronics and the Outstanding Student Paper Award at the First IEEE International Conference on Wireless Communication, Vehicular Technology, Information Theory, and Aerospace and Electronic Systems Technology and the Best Paper Award Finalist at the 3rd International Conference on Connected Vehicles & Expo.

Seung-Woo Seo (M97) received the B.S. and M.S. degrees from Seoul National University, Seoul, Korea, and the Ph.D. degree from Pennsylvania State University, University Park, PA, USA, all in electrical engineering.

He is a Professor of Electrical Engineering with Seoul National University and Director of the Intelligent Vehicle IT (IVIT) Research Center funded by the Korean Government and Automotive Industries. He was with the Faculty of the Department of Computer Science and Engineering, Pennsylvania State University, and served as a Member of the Research Staff with the Department of Electrical Engineering, Princeton University, Princeton, NJ. In 1996, he joined the Faculty of the School of Electrical Engineering, Institute of New Media and Communications and Automation and Systems Research Institute, Seoul National University. He has served as Chair or a Committee Member at various international conferences and workshops, including INFOCOM, GLOBECOM, PIMRC, VTC, MobiSec, Vitae, etc. He was the general co-chair of IEEE Intelligent Vehicle Symposium in 2015. He also served for five years as a Director of the Information Security Center in Seoul National University. His research areas include autonomous driving, machine learning, and security of vehicular communication.