Speed and Accuracy Tradeoff for LiDAR Data Based Road Boundary Detection

Guojun Wang, Jian Wu, Rui He, and Bin Tian

***Abstract*—Road boundary detection is essential for autonomous vehicle localization and decision-making, especially under GPS signal loss and lane discontinuities. For road boundary detection in structural environments, obstacle occlusions and large road curvature are two significant challenges. However, an effective and fast solution for these problems has remained elusive. To solve these problems, a speed and accuracy tradeoff method for LiDAR-based road boundary detection in structured environments is proposed. The proposed method consists of three main stages: 1) a multi-feature based method is applied to extract feature points; 2) a road-segmentation-line-based method is proposed for classifying left and right feature points; 3) an iterative Gaussian Process Regression (GPR) is employed for filtering out false points and extracting boundary points. To demonstrate the effectiveness of the proposed method, KITTI datasets is used for comprehensive experiments, and the performance of our approach is tested under different road conditions. Comprehensive experiments show the road-segmentation-line-based method can classify left, and right feature points on structured curved roads, and the proposed iterative Gaussian Process Regression can extract road boundary points on varied road shapes and traffic conditions. Meanwhile, the proposed road boundary detection method can achieve real-time performance with an average of 70.5 ms per frame.**

***Index Terms*—3D-LiDAR, autonomous vehicle, object detection, point cloud, road boundary.**

I. Introduction

In the autonomous vehicle (AV) field, environmental perception is a prerequisite for other functional modules, and road boundary detection is an important part of this perception. Road boundary can be used to distinguish on-road area from non-road area, which provides an important basis for AVs to understand scenes and make decisions. In addition, when GPS data is inaccurate, or perhaps has been lost in an urban environment, the road boundary can be used as an effective feature to locate AV. Therefore, accurate and reliable road research communities have invested much time into this area. According to sensors, current methods can be divided into vision-based and LiDAR-based methods.

With the development of computer vision technology, vision-based methods are widely used in road boundary detection [1]–[3]. Effective detection results can be achieved using vision-based methods, but performance is greatly affected by light and weather conditions. In addition, it is difficult to obtain accurate depth information, which makes it difficult to meet the needs of autonomous driving applications.

Compared with vision sensors, LiDAR can provide the accurate depth information and is immune to illumination and shadow. Thus, LiDAR has become an indispensable sensor for AV at this stage. In [4], the method based on the 2D-LiDAR interactive multi-model was used to detect road boundaries. In [5], road boundary points were extracted as line segments in polar coordinates based on 2D-LiDAR. However, due to the sparsity of 2D-LiDAR data, it is difficult to meet the needs of environmental perception. Compared with 2D-LiDAR, 3D-LiDAR has the advantages of a 360-degree scanning range, which can provide a large quantity of data, known as point cloud, and thus plays an increasingly important role in AV. In previous the Defense Advanced Research Projects Agency (DARPA) challenges, many teams used 3D-LiDARs for environmental perception [6], [7]. In recent years, Waymo, Baidu, and other companies have chosen 3D-LiDAR as the main sensor to obtain environmental information [8]–[10].

Although 3D-LiDAR has many advantages, there are still two great challenges for road boundary detection: the classification of road boundary points on curved roads and obstacle occlusions. On curved roads, it is difficult to distinguish left and right boundary points even if all boundary points are extracted, which also has an important effect on autonomous driving safety. Besides, when obstacle occlusions exist, false points caused by obstacles will also affect road boundary detection.

In order to solve these problems, we propose a speed and accuracy tradeoff method to extract road boundaries from the point cloud. The proposed method can filter out false points caused by obstacles under non-congested traffic conditions and can correctly classify left and right boundary points on curved roads. The main contributions of this paper are as follows:

1) This paper presents a road-segmentation-line-based

Manuscript received December 27, 2019; revised March 21, 2020, June 2, 2020; accepted June 25, 2020. This work was supported by the Research on Construction and Simulation Technology of Hardware in Loop Testing Scenario for Self-Driving Electric Vehicle in China (2018YFB0105103J). Recommended by Associate Editor Lei Shu. *(Corresponding author: Jian Wu.)*

Citation: G. J. Wang, J. Wu, R. He, and B. Tian, "Speed and accuracy tradeoff for LiDAR data based road boundary detection," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 6, pp. 1210–1220, Jun. 2021.

G. J. Wang, J. Wu, and R. He are with the State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130022, China (e-mail: 839977837wgj@gmail.com; wujian@jlu.edu.cn; herui@jlu.edu.cn).

B. Tian is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, and also with the Qingdao Academy of Intelligent Industries, Shandong, China (e-mail: bin.tian@ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2020.1003414

classification method for classifying feature points. The feature points are rough road boundary candidate points extracted by spatial and geometrical features. The road segmentation line is determined by a beam band model and an improved peak-finding algorithm. The method can classify left and right feature points accurately on curved roads up to 70 m away.

2) This paper proposes a distance filter and random sample consensus (RANSAC) filter for candidate point extraction and seed point extraction. The candidate points and seed points would be used for the subsequent feature points filtering based on an iterative Gaussian Process.

3) This paper proposes an iterative Gaussian Process Regression (GPR) based feature points filtering method. The GPR algorithm can be applied to various road shapes without assuming that road boundaries are parametric models. At this same time, the algorithm can effectively remove false points caused by obstacle occlusions. Because GPR is a nonparametric model, it significantly enhances the adaptability to various road shapes and the robustness for obstacle occlusions.

The road-segmentation-line-based classification method and the iterative GPR effectively addressed the problem of curved road and obstacle occlusions in structured environments. The proposed road boundary detection method achieves excellent accuracy while ensuring real-time performance.

The remainder of this paper is organized as follows. Section II describes a review of related research. Section III introduces the methodologies for road boundary detection. Section IV presents comprehensive experiments and evaluations. Section V summarizes the major contributions of this research, and its future work is presented.

II. Related Work

A. Related Work on Feature Points Classification

For the classification of left and right road boundary points, most of the existing literature focuses on the problem of straight road structures in which left and right boundary points are classified just according to lateral coordinates, such as in [11]–[15]. Xu *et al.* [16] extracted feature points based on energy functions and classified left and right boundary points using the least-cost path model. This method can work well on curved roads, but it requires manual addition of source points, making this algorithm unusable in practice. In [17], super voxels were used to obtain boundaries, and then trajectory data was used to classify left and right boundary points, which can only work offline and are not suitable for on-line applications. In [18] and [19], in order to solve the problem of boundary detection on curved roads, clustering methods were proposed to divide left and right boundary points. However, these methods need to iterate at each possible segmentation angle to maximize the classification score, which requires a lot of computation. In [20], road boundary points are searched and separated by the predicted trajectory of autonomous vehicles. In this method, the cumulative error of the predicted trajectory would increase, which would affect the accuracy of road boundary points detection. In [21], the parametric active contour and snake model were used to extract the left and right road boundary, and navigation information was necessary to initialize the snake model.

B. Related Work on Feature Points Filtering

For the problem of obstacle occlusions, various filtering methods have been proposed. Sun *et al.* [11] and Yang *et al.* [14] firstly classified feature points into segments with k-nearest neighbor (KNN) method, then segments of less than three points were considered as false points and eliminated, which may filter out true isolated boundary points. In [12], a RANSAC line fitting algorithm was used to filter out false points, which assumed that roads were straight, thus making it not suitable for curved roads. In [15], a regression filter was introduced to make the detection robust to occlusions. In [19], a RANSAC quadratic polynomial fitting algorithm was used to remove false points. References [12], [15] and [19] modeled road boundaries as a linear or quadratic polynomial parameterized model, and were not suitable for various road shapes. In [20], road boundary points are extracted only based on spatial features, which is not enough to remove false points caused by obstacles. In [22], feature points were extracted based on local normal saliency, and distance to trajectory was used to filter out false points that cannot filter out false points inside the road. In [23], region growing–based filtering was used to extract true road boundary points based on the similarity of height and intensity. However, in a structured environment, the materials used for road boundaries and road surfaces are usually the same, and the intensity was not effective. In [24], multiple parameterized RANSAC models were used to extract road boundary points.

III. Methodology

As shown in Fig. 1, the proposed method consists of four main steps. It takes a frame of raw point cloud as input and outputs road boundary points.

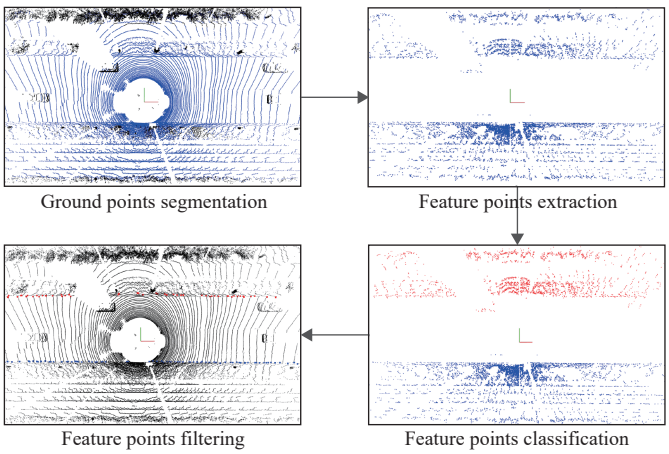

Fig. 1. The pipeline of road boundary detection method.

A. Ground Points Segmentation

The proposed method is used to process point cloud from Velodyne HDL-64E LiDAR [25]. The LiDAR coordinate systems are defined, as shown in Fig. 2. The directions are

given from the drivers' view, X: forward, Y: left, Z: up, and the coordinate system is right-handed. In this paper, we consider the range of $[-3, 1] \times [-40, 40] \times [-70, 70]$ meters along for the Z-axis, Y-axis, X-axis, respectively as the detection regions for road boundaries.

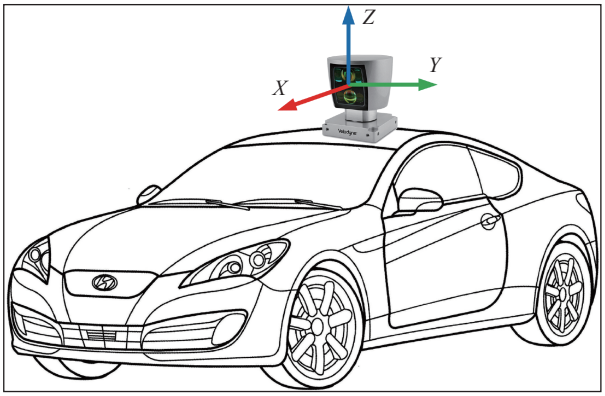

Fig. 2. The cartesian coordinate system of LiDAR.

Since road boundaries are a part of the ground, off-ground and on-ground points can be separated by ground segmentation. In this paper, a piecewise plane-based fitting method is used. For convenience, we use P to represent a frame of raw point cloud. First, point cloud P is divided into several segments along the x-axis. Then, a RANSAC plane fitting method is applied in each segment to extract on-ground points [26]. Because the point cloud is divided into segments, we assume that the slope of the ground in each segment does not fluctuate greatly and can be approximated to a plane. Besides, we focus on road boundary detection, so the ground point segmentation does not require high precision, and the distance threshold is set to 28 cm, which is determined through a trial-and-error method. The distance threshold should ensure that the ground point contains all road boundary points, which generally in the range of 15 cm to 30 cm.

Fig. 3 shows the results of a ground point segmentation, where on-ground points are used to extract road boundary points, and off-ground points are used to compute road segmentation lines for classifying road boundary feature points. For convenience, let P_{on} denote on-ground points and P_{off} off-ground points. All figures pertaining to point cloud illustrations in this paper are from the top view.

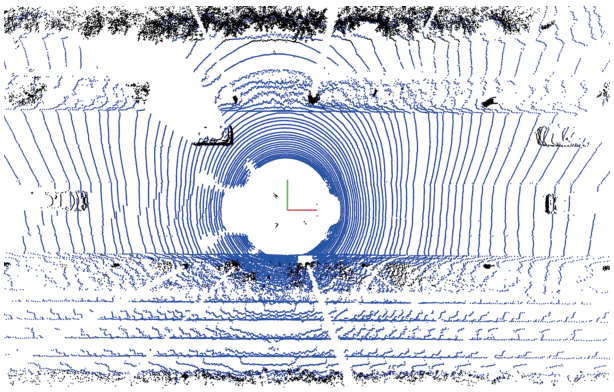

Fig. 3. An example of ground point segmentation. On-ground points are in blue, off-ground points are in black.

B. Feature Points Extraction

The algorithm of feature point extraction is mainly inspired by [27] and [28]. In order to extract all feature points as much as possible, three spatial features with loose thresholds are used to extract feature points. The on-ground points P_{on} are divided into 64 scan layers to extract feature points, and each layer contains all points from the corresponding laser. For convenience, let $P_{li} = [x_{li}\ y_{li}\ z_{li}]$ denote each point, where l is the corresponding layer ID. The spatial features applied in this paper are described in the following.

1) Height Difference

Let $Z_{\max}$ and $Z_{\min}$ denote the maximum and minimum z values of neighbors of p_{li}, respectively. Thus, the height difference feature is defined as

$$T_{height1} \le Z_{\max} - Z_{\min} \le T_{height2}$$

$$\sqrt{\frac{\sum (z_{li} - \mu)^2}{n_{height}}} \ge T_{height3}$$

方差，可以想象成椭球的横轴阈值

where $\mu = \sum z_{li} / n_{height}$, n_{height} is the number of neighbors, z_{li} is the z values of each neighbor where l is the layer ID, and $T_{height1}$, $T_{height2}$, *and* $T_{height3}$ are thresholds.

2) Smoothness

The feature proposed by [28] is used to describe the smoothness of the area around some point. For any point p_{li}, let S represent its neighbors. Thus, the smoothness feature is defined as

$$s = \frac{1}{|S|\|P_{li}\|} \cdot \left\| \sum_{\substack{P_{lj} \in S \\ j \ne i}} (p_{li} - p_{lj}) \right\|$$

$$s \ge T_{smoothness}$$

where s is the smoothness value of p_{li}, $|S|$ is the cardinality of S, and $T_{smoothness}$ is the threshold.

3) Horizontal Distance

The horizontal distance feature is proposed by [27] to represent the horizontal distance between two adjacent points in the same layer. It sets a horizontal distance threshold $\delta_{xy,l}$ to determine whether point p_{li} is a feature point. It is defined by

$$\delta_{xy,l} = H_s \cdot \cot \theta_l \cdot \frac{\pi \theta_a}{180}$$

where H_s is the absolute value of the height of point p_{li}, θ_l is the vertical azimuth of scanning layer l, and θ_a is the horizontal angular resolution of LiDAR. If p_{li} is selected as a feature point, the horizontal distance between p_{li} and its adjacent points should be smaller than $\delta_{xy,l}$.

All the features are tested in experiments, and loose thresholds are determined by a trial-and-error method. The height thresholds $T_{height1}$, $T_{height2}$, and $T_{height3}$ are determined based on the height jump of road boundaries. $T_{height1}$ and $T_{height3}$ are generally drawn from [0.01, 0.05], and $T_{height2}$ is generally drawn from [0.15, 0.35]. The smoothness threshold is determined based on the smoothness of road boundary

areas, so $T_{smoothness}$ is generally drawn from [0.001, 0.005].

C. Feature Points Classification

In order to solve the problem of classification for road boundary points, especially on curved roads, this paper proposes a road-segmentation-line-based method to classify feature points, which is mainly inspired by [27]. The principle of the method is that the beam model [29] is established based on off-ground points P_{off}, from which the length of each beam is calculated. The longest beams in the front and rear regions of the AV are regarded as road segmentation lines, which indicates the direction of the road. Compared with [27], an improved peak-finding algorithm is proposed to filter out outliers and determine two truly dominant extremes in the distance function $\delta(k)$. Thus, the proposed improved peak-finding algorithm in this paper is more robust for the sparsity of point cloud and losses of laser return. Then, feature points $P_{feature}$ are divided into left and right feature points according to the road segmentation lines. Compared with the clustering methods used in our previous work [19], the proposed method can achieve real-time performance, and it can classify feature points accurately under diverse road conditions.

1) Beam Band Model

The beam model was proposed by Thrun [29] in 2005 and has been widely used in the field of robotics. In order to determine the accurate angle of road segmentation, the angular resolution θ_{model} of the beam model is set to 1 degree. In classical beam models, the beam length is the distance from the launching point to the nearest point. In this paper, we expand the beam into a beam band Z_k

$$Z_k = \{(x,y) | (k-1) < \arctan\left(\frac{y-y_b}{x-x_b}\right) \cdot \frac{180}{\pi} \le k\} \tag{1}$$

where (x_b, y_b) is the launching point, which is the LiDAR origin in this paper. The LiDAR scans clockwise, where zero degrees is on the x-axis. (x,y) denotes off-ground points, and $k \in \{1,2,\ldots,360\}$. The beam length of each band is defined as the shortest distance of the points in this band to the launching point, that is

$$d(k) = \min_i \sqrt{(x_i - x_b)^2 + (y_i - y_b)^2} \tag{2}$$

where $(x_i, y_i) \in (Z_k \cap P_{off})$. The beam length of each band $d(k)$ is normalized to get distance function $\delta(k)$:

$$\delta(k) = \frac{d(k)}{\max_{x,y \in Z_k} \sqrt{(x-x_b)^2 + (y-y_b)^2}} \tag{3}$$

where if there is no point in the k-th band, $\delta(k) = 1$. So, in order to determine road segmentation lines, especially in roads with large curvatures, the longest beam will be used as road segmentation line in front region ($\{0 \le k \le 90 \cup 270 \le k < 360\}$) and rear region ($\{90 \le k < 270\}$) of the AV. After that, an improved peak-finding method is used to determine the road segmentation line.

2) Improved Peak-Finding Algorithm

The peak-finding algorithm was first proposed to find the number of dominant extremes that represent road direction from aerial images in [30]. For example, Figs. 4 and 5 show a beam model based on off-ground points P_{off} and corresponding distance function $\delta(k)$. It is clear that because point cloud data is sparse, the distance function has many false dominant extremes ($\delta(k) = 1$). However, we just need two true dominant extremes to determine the road segmentation line. To solve this problem, we present an improved peak-finding algorithm. The algorithm consists of two steps: the first median filter is applied to process the distance function, then each dominant extreme is evaluated based on the extreme width.

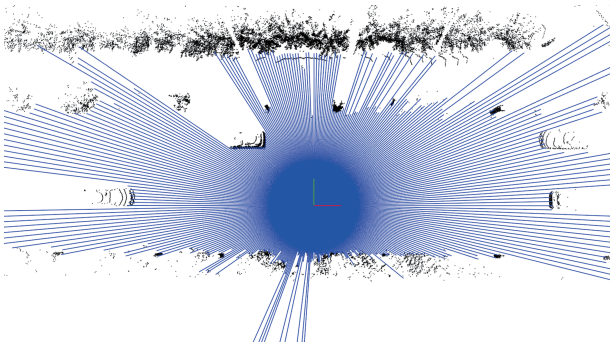

Fig. 4. Beam model based on off-ground points. The blue lines are middle lines of each beam band.

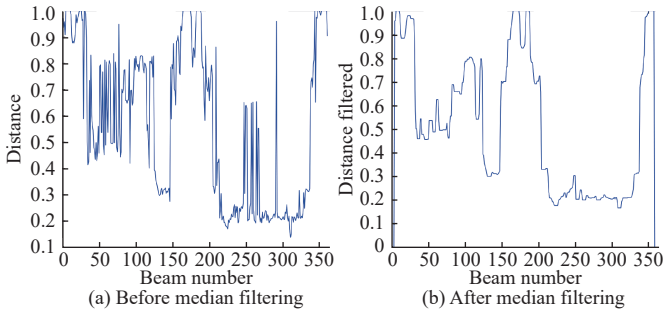

Fig. 5. An example of distance function for beam band model.

a) Median filtering

Median filtering is a nonlinear digital filtering technique which is often used to remove noise from an image or signal [31]. With the beam model established above, a distance function $\delta(k)$ is obtained, and each $\delta(k)$ is replaced by the median of its corresponding neighbors. Figs. 5(a) and 5(b) are the distance function before and after filtering, respectively.

b) Evaluating dominant extremes based on extreme width

For dominant extreme $\delta(k)$, extreme width w is defined as follows:

$$w = (w_r + w_l) \tag{4}$$

$$w_l = |k_l - k| \quad w_r = |k_r - k| \tag{5}$$

where k_l is the nearest beam band whose distance $\delta(k_l)$ is less than $\delta(k)$ on the left of k, and k_r is the nearest band whose distance $\delta(k_r)$ is less than $\delta(k)$ on the right of k. The evaluation of dominant extremes is as follows:

i) Remove the dominant extremes whose w is less than a pre-defined threshold T_w

ii) Merge the dominant extremes that are close. For two

extremes $\delta(k_i)$ and $\delta(k_j)$, $|k_i - k_j| < T_{distance}$ or $|k_i - k_j + 360| < T_{distance}$, then, the extreme whose width w is larger will be chosen. If the extreme widths are the same, the dominant extreme whose value of $|w_l - w_r|$ is smaller will be chosen.

According to the above method, the respective dominant extremes are found in two intervals $\{0 \le k \le 90 \cup 270 \le k < 360\}$ and $\{90 \le k < 270\}$. The middle lines of bands that correspond to the two extremes are used as road segmentation lines. Based on the road segmentation line, the feature points located on the left side of the road segmentation line can be classified as one class, and the feature points located on the right side of the road segmentation line can be classified as another class.

In order to demonstrate the effectiveness of the proposed method, experiments are carried out under three different road curvature conditions, as shown in Fig. 6 , the green line represents the road segmentation line, where the left feature points $P_{feature,l}$ are blue, and the right feature points $P_{feature,r}$ are red. We can see that the proposed improved peak-finding method can determine road segmentation lines accurately on different curved roads.

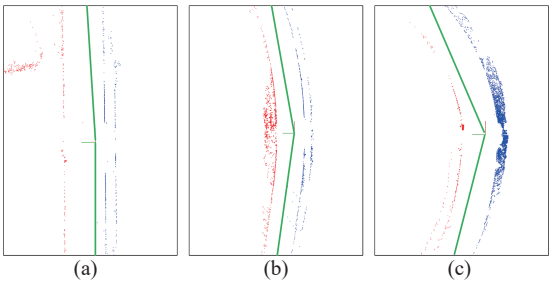

Fig. 6. Results of feature points classification on three different road curvatures. The green line represents the road segmentation line, the left feature points are in blue, and the right feature points are in red. (a) Small. (b) Moderate. (c) Large.

D. Feature Points Filtering

After feature points are extracted and classified, there are still many false points, including vehicles, pedestrians, railway tracks, adjacent roads, and so on. Besides, even on structured road environments, the shape of road boundaries is irregular and cannot be accurately modeled with parametric models. The GPR model is a nonparametric model, which has both powerful approximate and outlier rejection abilities. Based on this, an iterative GPR algorithm is proposed to model road boundary and filter out false points. Before the iterative GPR, a filtering-based method is proposed in this paper to extract candidate points and seed points.

1) Candidate Points and Seed Points Extraction

The filtering-based method includes a distance filter and a RANSAC filter. Feature points $P_{feature,l}$ and $P_{feature,r}$ are filtered by the distance filter to obtain candidate points $P_{candidate,l}$ and $P_{candidate,r}$, and seed points $P_{seed,l}$ and $P_{seed,r}$ are obtained by the RANSAC filter.

a) Distance filter: The distance filter is used to remove false points caused by obstacles outside roads. In general, we assume that road boundaries are the nearest obstacle to vehicles. So, the method is to divide feature points into segments based on x coordinates, and to find the nearest point to the x-axis of LiDAR as a candidate point in each segment:

$$S_i = \left\{(x,y)\,|\,(i-1) \le \frac{x}{w_s} < i\right\}, \quad i \in Z \tag{6}$$

$$f_{dist}(S_i) = \underset{S_i}{\operatorname{argmin}} |y_{i,j}| \tag{7}$$

where S_i is segment i, $y_{i,j}$ is the y coordinate of j-th point of segment i, w_s is the width of segments which need to be given, and $f_{dist}(S_i)$ is the index of the nearest point in segment i. For each segment, only one nearest point is preserved. The result of the distance filter for $P_{feature,l}$ and $P_{feature,r}$ is $P_{candidate,l}$ and $P_{candidate,r}$, as shown in Fig. 7.

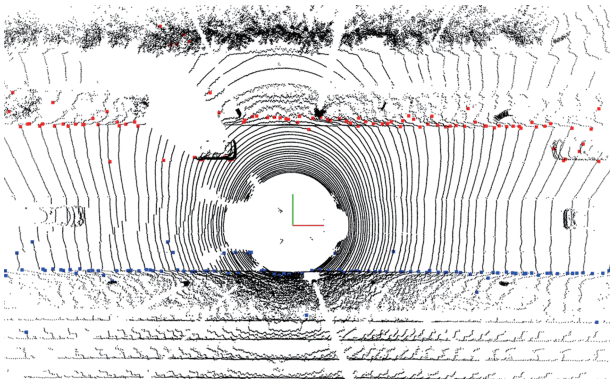

Fig. 7. Candidate points (left is in red, right is in blue).

b) RANSAC filter: In order to extract seed points, the false points caused by obstacles inside roads such as pedestrians and vehicles must be filtered out. The filter models road boundaries as a quadratic polynomial, then a RANSAC algorithm is applied to fit the model [26]. Then, all the points whose distances from the fitted model are less than the threshold are used as seed points. The result of RANSAC filter for $P_{candidate,l}$ and $P_{candidate,r}$ is $P_{seed,l}$ and $P_{seed,r}$, as shown in Fig. 8. It is clear that the RANSAC filter also removes some true boundary points because of the assumption that road boundaries fit a quadratic polynomial model. In order to improve the robustness of road boundary detection, iterative GPR is further applied to extract road boundary points.

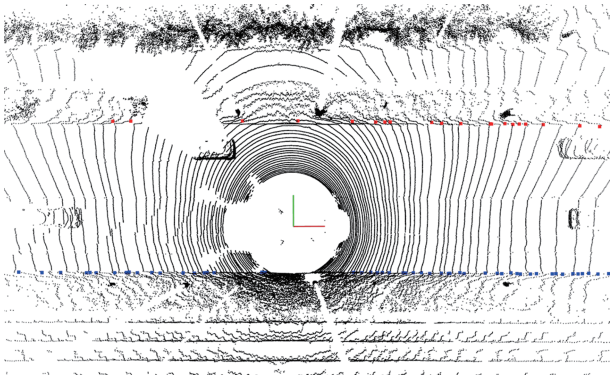

Fig. 8. Seed points (left is in red, right is in blue).

2) Iterative Gaussian Process Regression

Due to the diversity of road shapes in urban environments, a single parametric model cannot be used to accurately model road boundaries. Besides, there are many false points in candidate points, and iterative GPR happens to have strong outlier rejection abilities. To improve the robustness of the road boundary detection algorithm, this paper proposes to use GPR to model road boundary, which is inspired by [32].

Gaussian Process (GP) is a combination of random variables, where finite random variables have a joint Gaussian distribution [33]. Given a 2D training set $\boldsymbol{D} = \{(x_i, y_i),\ i = 1, \ldots, n\}$, and that observation y_i contains noise ε, the Gaussian noise model is $y = f(x) + \varepsilon$, where ε is an independent random variable and $\varepsilon \sim N(0, \sigma_n^2)$. The priori distribution of the observations $\boldsymbol{Y}$ is obtained as follows:

$$\boldsymbol{Y} \sim N(\boldsymbol{\mu}, \boldsymbol{K} + \sigma_n^2 \boldsymbol{I}_n) \tag{8}$$

where $\boldsymbol{X} = [x_1, x_2, \ldots, x_n]$, $\boldsymbol{Y} = [y_1, y_2, \ldots, y_n]$, $\boldsymbol{I}_n$ is the identity matrix and $\boldsymbol{K}$ is covariance matrix which is defined as

$$\boldsymbol{K} = \left(k_{ij}\right) = \left(k\left(x_i, x_j\right)\right) \tag{9}$$

$$k(x_1, x_2) = \sigma_f^2 \exp\left(-\frac{(x_i - x_j)^2}{2l^2}\right) \tag{10}$$

where l is the length-scale, and σ_f^2 is the signal covariance.

Based on the definition of GPR, the joint Gaussian distribution of the training set $\boldsymbol{Y}$ and the output y_* at test input x_* meets:

$$\begin{bmatrix} \boldsymbol{Y} \\ y_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I_n} & \boldsymbol{K}(\boldsymbol{X}, x_*) \\ \boldsymbol{K}(x_*, \boldsymbol{X}) & \boldsymbol{K}(x_*, x_*) \end{bmatrix}\right). \tag{11}$$

Thus, the predictive value for test input x_* is

$$\bar{y_*} = \boldsymbol{K}(\boldsymbol{X}, x_*)^T \left[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}_n\right]^{-1} \boldsymbol{Y} \tag{12}$$

$$cov(y_*) = \boldsymbol{K}(x_*, x_*) - \boldsymbol{K}(\boldsymbol{X}, x_*)^T \left[\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2 \boldsymbol{I}_n\right]^{-1} \boldsymbol{K}(\boldsymbol{X}, x_*) \tag{13}$$

where $\bar{y_*}$ and $\operatorname{cov}(y_*)$ are the mean and covariance of the output y_*, $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) = \left(k\left(x_i, x_j\right)\right)_{1 \le i, j \le n}$, $\boldsymbol{K}(\boldsymbol{X}, x_*) = \boldsymbol{K}(x_*, \boldsymbol{X})^T = (k(x_i, x_*))$, $\boldsymbol{K}(x_*, x_*) = k(x_*, x_*)$.

There are still many false points in the candidate points. We must separate the true road boundary points $P_{boundary,l}$ and $P_{boundary,r}$ from candidate points $P_{candidate,l}$ and $P_{candidate,r}$ based on seed points $P_{seed,l}$ and $P_{seed,r}$. However, GPR assumes that all data in candidate points are boundary points without any outliers. An improved iterative GPR is applied to model road boundaries [34], which not only has a strong approximating ability but can also eliminate false points. The detailed algorithm is shown in Table I.

The algorithm mainly consists of two steps: the GPR modeling with covariance function and the evaluation of test points P_{test}.

The GPR modeling with covariance function corresponds to the 5th and 14th lines of Table I. The evaluation of test points P_{test} corresponds to the 7th and 16th lines of Table I. For each point $x_{*,i}, y_{*,i}$ in test points P_{test}, the corresponding mean $\bar{y}_{i,*}$ and variance $\operatorname{cov}(y_{i,*})$ are calculated by (12) and (13). The criteria proposed in [34] are used to determine whether the point $x_{*,i}, y_{*,i}$ is a new road boundary point:

$$cov(y_*) \le t_{model} \tag{14}$$

$$\frac{\left|y_{*,i} - \bar{y}_*\right|}{\sqrt{\sigma_n^2 + \operatorname{cov}(y_*)}} \le t_{data} \tag{15}$$

An example of road boundary extraction based on GPR is shown in Fig. 9. Compared with [32], this paper extracts left road boundary points $P_{boundary,l}$ and right road boundary points $P_{boundary,r}$ from left candidate points $P_{candidate,l}$ and right candidate points $P_{candidate,r}$, respectively, and they do not interfere with each other. At the same time, the computational burden of eval function *eval* is greatly reduced in iterative process.

TABLE I
BOUNDARY POINTS EXTRACTION BASED ON ITERATIVE GPR

Input: $P_{candidate,l}$, $P_{candidate,r}$, $P_{seed,l}$, $P_{seed,r}$, l, σ_n, σ_f, t_{model}, t_{data}

Output: $P_{boudary,l}$, $P_{boundayr,r}$

1 //* Extracting left boundary points*//
2 $P_{new} = P_{seed,l}$, $P_{temp} = \emptyset$
3 **while** size (P_{new})>0 **do**
4 $P_{temp} = P_{temp} \cup P_{new}$;
5 $model = GPR(P_{temp}, l, \sigma_n, \sigma_f)$;
6 $P_{test} = P_{candidate,l} - P_{temp}$;
7 $P_{new} = \text{eval}(model, P_{test}, t_{model}, t_{data})$;
8 end while
9 $P_{boudary,l} = P_{temp}$
10 //* Extracting right boundary points*//
11 $P_{new} = P_{seed,r}$, $P_{temp} = \emptyset$
12 **while** size (P_{new})>0 **do**
13 $P_{temp} = P_{temp} \cup P_{new}$;
14 $model = GPR(P_{temp}, l, \sigma_n, \sigma_f)$;
15 $P_{test} = P_{candidate,r} - P_{temp}$;
16 $P_{new} = \text{eval}(model, P_{test}, t_{model}, t_{data})$;
17 end while
18 $P_{boudary,r} = P_{temp}$

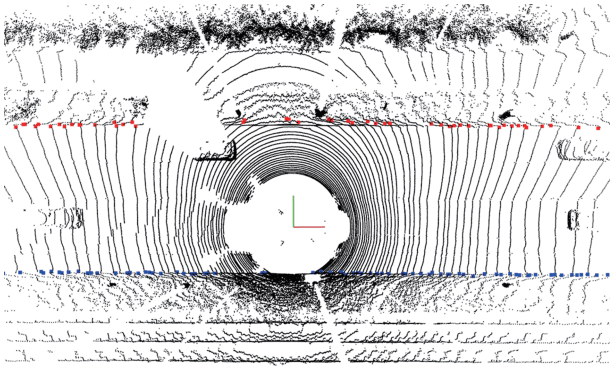

Fig. 9. Road boundary points extracted by GPR.

3) Learning Hyper-Parameters of GPR

In order to obtain the hyperparameter $\theta = \{l, \sigma_f, \sigma_n\}$ used in iterative GPR, the M =800 frames road boundary data (including left and right road boundaries) labeled manually is used as training samples, $\boldsymbol{D_{train}} = \{(x^m, y^m), m = 1, \ldots, M\}$, each frame (x^m, y^m) is used as a training sample, $y^m = \{y_1^m, \ldots, y_{n_m}^m\}$ is the sequence of y coordinates in frame m, and $x^m = \{x_1^m, \ldots, x_{n_m}^m\}$ is the sequence of x coordinates in frame m. We assume that these training examples are conditionally independent, so hyperparameter $\theta = \{l, \sigma_f, \sigma_n\}$ can be computed by maximizing the pseudo log marginal likelihood:

$$\sum_{m=1}^{M} \log p(y^m | x^m, \theta) = -\frac{1}{2} \sum_{m=1}^{M} (y^m)^T \boldsymbol{K_m^{-1}} y^m - \frac{1}{2} \log \left(\prod_{m=1}^{M} |\boldsymbol{K_m}| \right) - \frac{\log 2\pi}{2} \sum_{m=1}^{M} n_m \tag{16}$$

where $\boldsymbol{K_m}$ is the covariance matrix for the observations y^m in frame m and can be computed with (10).

IV. Experiment Evaluation and Analysis

To evaluate the proposed method in this paper, experiments were conducted on different road scenes. The experiments are divided into two parts. The first part is the qualitative experiment on the validity of the proposed method, and the second part is a quantitative experiment measuring accuracy and real-time performance. The KITTI Velodyne dataset [35] is used to evaluate the proposed method, which mainly covers urban and highway scenes. To demonstrate the effectiveness of our method on different road conditions, the proposed method is tested and evaluated on four typical road scenes (straight road, curved road, road with obstacles, and road with varying width). For each road scene, we manually labeled 500 frames for the road boundaries with the SemanticKITTI labeling tool [36]. The method is implemented by C++ and point cloud library (PCL) on Ubuntu 16.04.

In order to demonstrate the effectiveness of the proposed method, Zhang's [27] and Sun's [20] methods are employed for comparison. The values of parameters in our road boundary detection method are defined in Table II.

TABLE II
Parameters for Our Experiments

Parameters	Value
$T_{height1}$	0.03
$T_{height2}$	0.30
$T_{height3}$	0.01
$T_{smoothness}$	0.005
$T_{distance}$	10
T_w	5
t_{data}	3
t_{model}	5
l	16.11
σ_f	6.18
σ_n	0.06

A. Qualitative Evaluation of Road Boundary Methods

For straight roads, as shown in Fig. 10, all three methods work well, and the results are similar because the road conditions are simple, and road boundaries are regular. Only spatial features and constraints can extract road boundary points well.

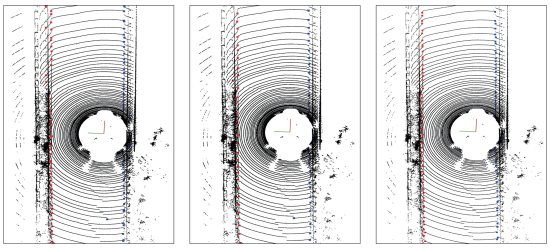

Fig. 10. The results of three methods on straight roads (left is in red, right is in blue). (a) Zhang. (b) Sun. (c) Proposed.

For a straight-curved road, as shown in Fig. 11, Zhang's has some misclassified points, which shows the peak-finding method is not robust for the sparsity of point cloud. The proposed improved peak-finding method and Sun's achieve accurate classification for road boundary points on curved roads.

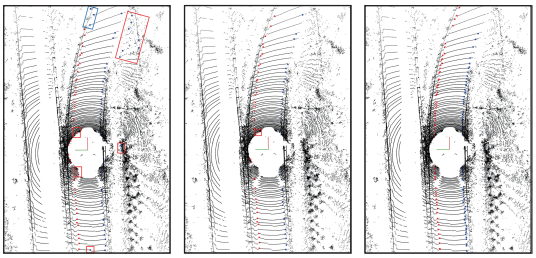

Fig. 11. The results of three methods on straight-curved roads (left is in red, right is in blue). In the red box are false points, and in the blue box are the points that have been misclassified. (a) Zhang. (b) Sun. (c) Proposed.

From Fig. 11 through Fig. 14, we can see that Zhang's and Sun's have false points caused by surrounding obstacles, such as vehicles inside roads, grass, shrubs, and railway tracks because they only remove false points based on spatial and geometric features. On the contrary, the proposed iterative GPR method can filter false points caused by obstacles. The qualitative evaluation experiments illustrate that the proposed method is accurate and robust for various road scenes and achieves better performance than other methods.

B. Quantitative Evaluation of Road Boundary Methods

1) Accuracy

The three different methods are also numerically evaluated on four typical road environments: straight roads, curved roads, roads with varying widths, and roads with obstacles. For these labeled frames, the labeled boundary points are projected onto a local grid map, which covers the detection region defined in Section III with the resolution of 0.15 m. Each cell of the grid map is classified into the left boundary,

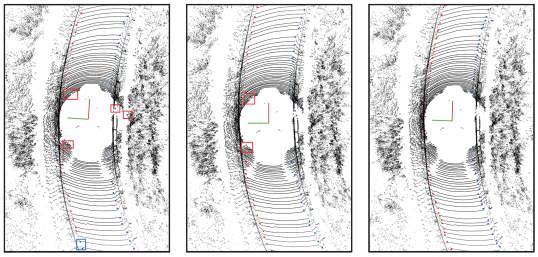

Fig. 12. The results of three methods on curved roads (left is in red, right is in blue). In the red box are false points, and in the blue box are the points that have been misclassified. (a) Zhang. (b) Sun. (c) Proposed.

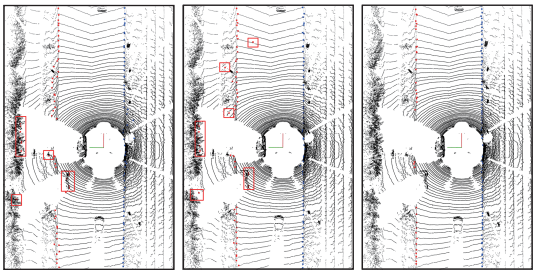

Fig. 13. The results of three methods on roads with obstacles (left is in red, right is in blue). In the red box are false points. (a) Zhang. (b) Sun. (c) Proposed.

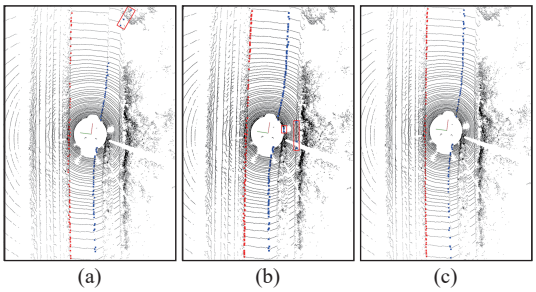

Fig. 14. The results of three methods on roads with varying widths (left is in red, right is in blue). In the red box are false points. (a) Zhang. (b) Sun. (c) Proposed.

right boundary, or none; thus the ground truth of each cell is obtained. With the same method, the detection results of each cell can be obtained. When one cell is detected as the left boundary, and is also labeled as left boundary, we believe that this cell is detected correctly.

To quantitatively evaluate our algorithm, three quantitative metrics in [37] and [38] are introduced for a comprehensive evaluation.

Precision, which denotes the proportion of cells detected correctly in all detected cells.

$$\boldsymbol{Precision} = \frac{TP}{TP+FP}$$

where TP is the number of cells detected correctly, and FP is the number of cells detected incorrectly.

Recall, which denotes the proportion of the cells detected correctly in the labeled cells.

$$\boldsymbol{Recall} = \frac{TP}{TP+FN}$$

where FN is the number of cells that are missed detections.

$\boldsymbol{F_1}$, which denotes the harmonic average of ***Precision*** and ***Recall***.

$$\boldsymbol{F_1} = 2 \cdot \frac{\boldsymbol{Precision} \cdot \boldsymbol{Recall}}{\boldsymbol{Precision} + \boldsymbol{Recall}}$$

The quantitative evaluation and comparison are given in Table III. We can see that the ***Recall*** of our method is just slightly lower than Zhang's and Sun's because of the distance filter used in our method. Only one nearest point is preserved in the distance filter. However, because iterative GPR is used to model road boundaries and remove outliers, the ***Precision*** and $\boldsymbol{F_1}$ of our method are the highest. It shows that the iterative GPR algorithm can effectively filter out false points and keep most boundary points. In addition, for the four different scenarios, the ***precision***, ***recall*** , and $\boldsymbol{F_1}$ of the proposed method do not vary much, reflecting the robustness of the proposed algorithm.

In Zhang's method [27], the bottom-layer beam model and the top-layer beam model were created based on the feature of the LiDAR sensor. Then, a rule-based peak-finding method was used to determine the road segmentation line. This method can achieve high ***Precision*** and ***Recall*** on straight roads. However, this method cannot handle outliers in distance function $\delta(k)$ which may cause misclassification for boundary points. On the contrary, the improved peak-finding algorithm in this paper can easily deal with this problem and select the two most suitable dominant extremes. Besides, boundary points are directly extracted by three spatial features, and the obstacle occlusions were not considered. So, the ***Precision*** and $\boldsymbol{F_1}$ of Zhang's method are lower than the proposed method, especially on curved roads and those with obstacles.

In Sun's method [20], first ground points are extracted by the polar-grid method, then road boundary points are directly detected based on a defined feature by searching along the predicted trajectory of vehicles. Because the trajectory was used to search boundary points, this method can correctly classify boundary points on various curved roads. However, the cumulative error of the predicted trajectory becomes larger as the scope of the prediction goes further, which would affect the accuracy of road boundaries. Similarly, the obstacle occlusions are also not considered, Sun's method has higher ***Recall*** and significantly lower ***Precision*** . On the contrary, because iterative GPR has both powerful approximate and outlier rejection abilities, our method achieves significantly higher ***precision*** and $\boldsymbol{F_1}$ with a small recall loss.

In conclusion, our proposed method can handle road boundary detection over various road scenes and achieves better performance. The other two methods achieve relatively poorer performance for obstacle occlusions and classification.

2) Runtime

We also evaluate the real-time performance of our method based on 1170 frames chosen from the KITTI dataset. Our experiments are performed on a 3.70 GHz Intel Core i7-8700 K processor with 16 GB of RAM. The runtime of our method is shown in Fig. 15 . We can see that the running time of our

TABLE III
QUALITATIVE EVALUATION RESULTS IN THE KITTI DATASET

Mean	Methods	Straight road	Curved road	Road with obstacles	Road with varying width
Precision	Proposed	**0.9214**	**0.9003**	**0.8973**	**0.8938**
	Zhang [27]	0.8874	0.8594	0.8093	0.8572
	Sun [20]	0.8858	0.8864	0.8294	0.8732
Recall	Proposed	0.8537	0.8616	0.7919	0.7845
	Zhang [27]	**0.8656**	0.8390	0.7598	0.7734
	Sun [20]	0.8528	**0.8695**	**0.8034**	**0.7934**
F_1	Proposed	**0.8863**	**0.8805**	**0.8413**	**0.8356**
	Zhang [27]	0.8715	0.8491	0.7838	0.8131
	Sun [20]	0.8690	0.8779	0.8162	0.8314

method is longer compared with Sun's and Zhang's, and the time spent on each frame varies greatly. The road boundaries of each frame are different, and the number of iterations of GPR varies, which results in violent fluctuations on time spent per frame. On the contrary, the methods of Sun and Zhang do not have false points filtering process. So, they have fewer and more consistent running times. However, our method can also perform with less than 100 ms per frame, with an average processing time of 70.5 ms. Because the scan period of LiDAR is 100 ms, real-time performance is achieved.

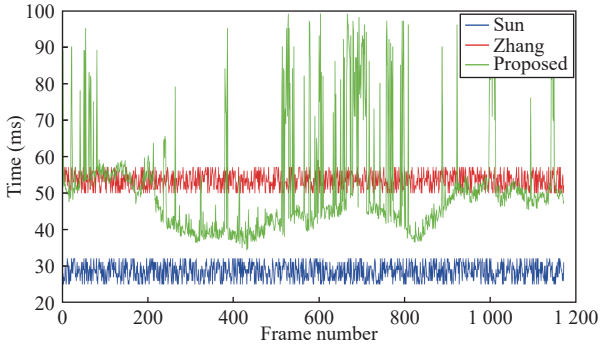

Fig. 15. The real-time performance comparison of three methods.

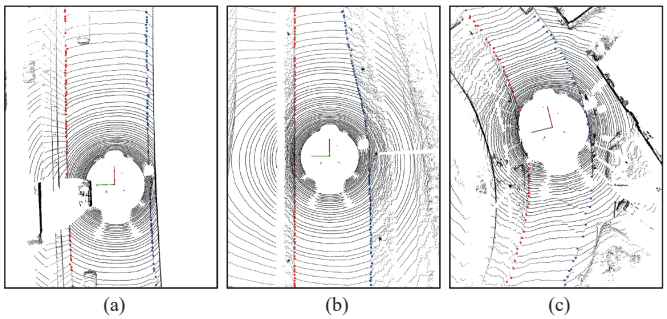

Fig. 16. Successful cases of the proposed road boundary method. (a) Roads with obstacles. (b) Roads with varying widths. (c) Roads with large curvatures.

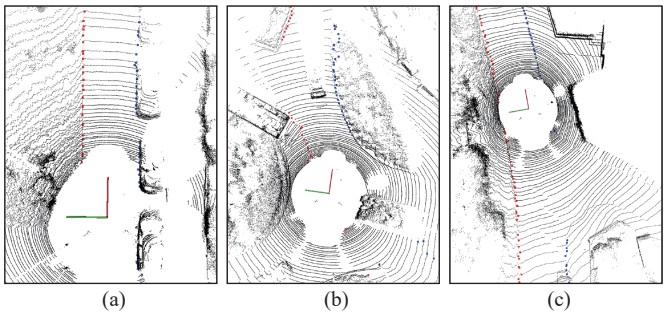

Fig. 17. Failure cases of the proposed road boundary method. (a) Roads with traffic congestion. (b) Roads with intersection. (c) *Y*-shape roads.

C. Failure Case

The method proposed in this paper is mainly aimed at road boundary detection in a structured non-congested traffic environment. The method achieved excellent performance under occluded, curved, and variable-width road conditions, as shown in Fig. 16. However, the proposed method achieves relatively poorer performance in heavy traffic environments and at complex intersections, as shown in Fig. 17. In Fig. 17(a), due to severe occlusion, our method incorrectly identified the side of the vehicles as the right road boundary.

V. CONCLUSIONS

This paper presents an effective road boundary detection method with speed and accuracy tradeoff in structured environments. A multi-feature loose-threshold method is applied for feature points extraction. In order to classify feature points, road segmentation lines are determined based on the beam band model and improved peak-finding algorithm. The road segmentation line can be determined exactly under curved roads up to 70 m away. Then, the distance filter and RANSAC filter are applied to extract candidate points and seed points. Finally, an iterative GPR is proposed to filter out false points and extract boundary points. Based on seed points, the iterative GPR can remove false points caused by road users. Because iterative GPR is a nonparametric model, it can handle various road shapes well. Comprehensive experiment evaluations clearly demonstrate that our proposed method can obtain real-time performance and achieve better accuracy, especially on occluded and curved roads.

The method proposed in this paper is mainly aimed at road boundary detection in a structured non-congested traffic

environment. In future works, we will focus on solving road boundary detection under traffic jams and complex intersections. In addition, we will simultaneously solve the problem of long-distance road boundary detection, which is necessary for high-speed driving.

REFERENCES

[1] F. Oniga, S. Nedevschi, and M. Michael Meinecke, "Curb detection based on a multi-frame persistence map for urban driving scenarios," in *Proc. IEEE Conf. Intelligent Transportation Systems*, 2008, pp. 67–72.

[2] J. Siegemund, D. Pfeiffer, U. Franke, and W. Forstner, "Curb reconstruction using conditional random fields," in *Proc. IEEE Conf. Intelligent Vehicles Symposium*, 2010, pp. 203–210.

[3] L. Wang, T. Wu, Z. Xiao, L. Xiao, D. Zhao, and J. Han, "Multi-cue road boundary detection using stereo vision," in *Proc. IEEE Conf. Vehicular Electronics and Safety*, 2016, pp. 48–53.

[4] Y. Kang, C. Roh, S. Suh, and B. Song, "A LiDAR-based decision-making method for road boundary detection using multiple Kalman filters," *IEEE Trans. Industrial Electronics*, vol. 59, no. 11, pp. 4360–4368, Nov. 2012.

[5] J. Han, D. Kim, M. Lee, and M. Sunwoo, "Road boundary detection and tracking for structured and unstructured roads using a 2D LiDAR sensor," *Int. Journal of Automotive Technology*, vol. 15, no. 4, pp. 611–623, 2014.

[6] M. Buehler, K. Iagnemma, and S. Singh, "Junior: The Stanford Entry in the Urban Challenge," in *The DARPA urban challenge*: *autonomous vehicles in city traffic*, vol. 56, Berlin, Heidelberg, Germany: Springer, 2009, pp. 91–123.

[7] G. Seetharaman, A. Lakhotia, and E. Blasch, "Unmanned vehicles come of age: The DARPA grand challenge," *IEEE Computer Society*, vol. 39, no. 12, pp. 26–29, 2006.

[8] S. Verghese, "Self-driving cars and LiDAR," in *Proc. Conf. on Lasers and Electro-Optics*, California, USA, 2017, pp. AM3A-1.

[9] J. Fang, F. Yan, T. Zhao, F. Zhang, D. Zhou, R. Yang, Y. Ma, and L. Wang, "Simulating LiDAR Point Cloud for Autonomous Driving using Real-world Scenes and Traffic Flows," *arXiv: 1811.07112*, 2018.

[10] K. Bimbraw, "Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology," in *Proc. Int. Conf. on Informatics in Control*, *Automation and Robotics*, 2015, pp. 191–198.

[11] P. Sun, X. Zhao, Z. Xu, and H. Min, "Urban curb robust detection algorithm based on 3D-LiDAR," *Journal of ZheJiang University*, vol. 52, no. 3, pp. 504–514, Mar. 2018.

[12] K. Hu, T. Wang, Z. Li, D. Chen, and X. Li, "Real-time extraction method of road boundary based on three-dimensional LiDAR," *Journal of Physics*: *Conf. Series*, vol. 1074, no. 1, pp. 1–8, 2018.

[13] W. Yao, Z. Deng, and L. Zhou, "Road curb detection using 3D LiDAR and integral laser points for intelligent vehicles," *Soft Computing and Intelligent Systems* (*SCIS*) *and 13th Int. Symp. on Advanced Intelligent Systems* (*ISIS*), pp. 100–105, 2012.

[14] B. Yang, L. Fang, and J. Li, "Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 79, pp. 80–93, 2013.

[15] A. Y. Hata and D. F. Wolf, "Feature detection for vehicle localization in urban environments using a multilayer LiDAR," *IEEE Trans. Intelligent Transportation Systems*, vol. 17, no. 2, pp. 420–429, 2016.

[16] S. Xu, R. Wang, and H. Zheng, "Road Curb Extraction from Mobile LiDAR Point Clouds," *IEEE Trans. Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 996–1009, 2017.

[17] D. Zai, J. Li, Y. Guo, M. Cheng, and Y. Lin, "3D road boundary extraction from mobile laser scanning data via supervoxels and graph cuts," *IEEE Trans. Intelligent Transportation Systems.*, vol. 19, no. 3, pp. 802–813, 2018.

[18] M. Wu, Z. Liu, and Z. Ren, "Algorithm of real-time road boundary detection based on 3D LiDAR," *Journal of Huazhong University of Science and Technology* (*Natural Science Edition*), vol. 39, no. 2, pp. 351–354, 2011.

[19] G. Wang, J. Wu, R. He, and S. Yang, "A Point Cloud-Based Robust Road Curb Detection and Tracking Method," *IEEE Access*, vol. 7, pp. 24611–24625, 2019.

[20] P. Sun, X. Zhao, Z. Xu, R. Wang, and H. Min, "A 3D LiDAR Data-Based Dedicated Road Boundary Detection Algorithm for Autonomous Vehicles," *IEEE Access*, vol. 7, pp. 29623–29638, 2019.

[21] P. Kumar, C. P. McElhinney, P. Lewis, and T. McCarthy, "An automated algorithm for extracting road edges from terrestrial mobile LiDAR data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 85, pp. 44–55, 2013.

[22] H. Wang, H. Luo, C. Wen, J. Cheng, and P. Li, "Road Boundaries Detection Based on Local Normal Saliency from Mobile Laser Scanning Data," *IEEE Geoscience Remote Sensing Letters*, vol. 12, no. 10, pp. 2085–2089, 2015.

[23] M. Yadav, A. K. Singh, and B. Lohani, "Extraction of road surface from mobile LiDAR data of complex road environment," *Int. Journal of Remote Sensing*, vol. 38, no. 16, pp. 4655–4682, 2017.

[24] Z. Liu, J. Wang, and D. Liu, "A new curb detection method for unmanned ground vehicles using 2D sequential laser data," *Sensors*, vol. 13, no. 1, pp. 1102–1120, 2013.

[25] Velodyne. (2019). HDL-64E. [Online]. Available: https://velodynelidar.com/hdl-64e.html, Accessed on: Apr. 16, 2020.

[26] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[27] Y. Zhang, J. Wang, X. Wang, and J. Dolan, "Road-segmentation-based curb detection method for self-driving via a 3D-LiDAR sensor," *IEEE Trans. Intelligent Transportation System*, vol. 19, no. 12, pp. 3981–3991, Dec. 2018.

[28] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Proc. Conf. Robotics: Science and Systems*, 2014.

[29] S. Thrun, W. Burgard, and D. Fox, "Robot Perception," in *Probabilistic Robotics*, London, England: MIT press, 2005, pp. 149–187. [Online]. Available: https://mitpress.mit.edu/books/probabilistic-robotics.

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

[31] Wikipedia, "Median filter," The Free Encyclopedia, USA. [Online]. Available: https://en.wikipedia.org/wiki/Median_filter, Accessed on: Nov. 5, 2019.

[32] T. Chen, B. Dai, D. Liu, J. Song, and Z. Liu, "Velodyne-based curb detection up to 50 meters away," in *Proc. IEEE Conf. Intelligent Vehicles Symp*., 2015, pp. 241–248.

[33] C. K. Williams and C. E. Rasmussen, "Regression," in *Gaussian Processes for Machine Learning*, London, UK: MIT press, 2006.

[34] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3D LiDAR point clouds." in *Proc. Conf. Robotics and Automation*, 2011, pp. 2798–2805.

[35] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite." in *Proc. Conf. Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[36] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE Conf. Computer Vision*, 2019, pp. 9297–9307.

[37] Powers and D. Martin, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, Feb. 2011.

[38] Y. Sasaki, "The truth of the F-measure," *Teach Tutor mater*, vol. 1, no. 5, pp. 1–5, 2007.

Guojun Wang received the B.S. degree in vehicle engineering from Yanshan University, Qinhuangdao, China in 2014. He is currently a Ph.D. candidate in vehicle engineering at Jilin University, Changchun, China. His current research interests include computer vision, automated driving, and deep learning.

Jian Wu received the B.S. and Ph.D. degrees from Jilin University, China in 1999 and 2010, respectively. He is a Professor of College of Automotive Engineering at Jilin University, China. His research interests are mainly in vehicle control system, electric vehicles, and intelligent vehicles. He is the authors of over 40 peers-reviewed papers in international journals and conferences and has been in charge of numerous projects funded by national government and institutional organizations on vehicles.

Rui He received the B.S. and Ph.D. degrees from Jilin University, Changchun, China in 2007 and 2012, respectively. He is currently an Associate Professor with the State Key Lab of Automotive Simulation and Control, Jilin University, China. His research interests are mainly in vehicle control system, electric vehicles, and intelligent vehicles.

Bin Tian received the B.S. degree from Shandong University, Jinan, China in 2009 and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China in 2014. He is currently an Associate Professor of the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, China. His current research interests include computer vision, machine learning, and automated driving.