IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 19, 2019, accepted February 3, 2019, date of publication February 14, 2019, date of current version March 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2898689

A Point Cloud-Based Robust Road Curb Detection and Tracking Method

GUOJUN WANG, JIAN WU, RUI HE, AND SHUN YANG
State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130022, China

Corresponding author: Rui He (herui@jlu.edu.cn)

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 under Grant 2018YFB0105103.

ABSTRACT Road curb detection is essential for autonomous vehicles to locate themselves and make a rational decision, especially under road discontinuities, obstacle occlusions, and curved road scenarios. However, an effective and systematic solution to this problem has remained elusive. In this paper, a robust 3D-LiDAR-based method for road curb detection and tracking in a structured environment is proposed. The proposed method consists of four main stages: 1) a multi-feature based method is applied to extract candidate points; 2) a density-based clustering method is proposed for classifying left and right candidate points; 3) a candidate points filter (including distance filter and RANSAC filter) is proposed to remove false points; and 4) a least-square algorithm is used to obtain road curb curve and the amplitude-limiting Kalman filter is deployed to prevent false detection and miss detection. The comprehensive experiment evaluations show that the proposed method can deal with straight and curved road without being influenced by surrounding obstacles.

INDEX TERMS Autonomous vehicle, 3D-LiDAR, curb detection, point cloud.

I. INTRODUCTION

Both industry and research communities in autonomous vehicle (AV) field are now putting increasing emphasis on environmental perception technology because of its fundamental role to other functional modules. Road curb detection, which always refers to the methods that extracting features and distinguishing between driving corridors and restricted areas, is the foremost part in environment perception. Road curb features are also used to locate self-driving cars when GPS signals are blocked by trees, bridges and tunnels in urban environments. To get a robust and precise curb detection result, various approaches have been introduced.

Vision-based methods have been extensively studied with the development of computer vision technology. In [1], a rectangular elevation map was built by a stereovision system Edge detection and Hough transformation methods were used to extract curb segments. In [2], the raw 3D points were assigned to curb adjacent surfaces with Conditional Random Field. Based on the result, curbs and surfaces were reconstructed. A Naive Bayes framework was used to fuse multiple features to get the highest probability road curb points in [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Razi Iqbal.

All these vision-based methods are effective under simple driving scenarios; however, these methods are sensitive to light and weather conditions and not robust enough in complex scenarios. Besides, accurate distance information cannot be guaranteed, which leads to that vision is difficult to fully meet practical applications

Compared with vision-based method, LiDARs can provide accurate distance information and are not affected by light conditions [4]. In [5], an interacting multiple model method is proposed to detect road curb based on 2D-LiDAR. However, the sparse data can hardly meet environmental perception demand. In comparison with 2D-LiDAR, 3DLiDAR plays a more and more important role as it provides a large number of point cloud data with a 360-degree coverage as shown in Fig. 1 In previous DARPA Challenge, many teams were equipped with 3D-LiDARs to obtain environment information [6]–[8]. Recently, Waymo, Baidu, and General Motors choose 3D-LiDARs as main sensor to perception environment [9]–[11] The 3D-LiDAR has several advantages in AV domain, however, when used in road curb detection, there are also two major challenges that affect robustness: obstacle interference and classification of left and right candidate points in curved road. To address these two challenges, this paper uses three steps to extract curb points, including

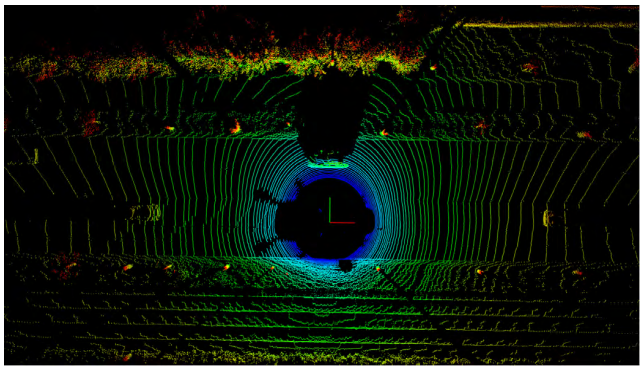

FIGURE 1. The point cloud obtained from 3D-LiDAR.

candidate points extraction, candidate points classification and candidate points filtering.

A. RELATED WORK ON CANDIDATE POINTS EXTRACTION

For the problem of candidate points extraction, Sun *et al.* [12] obtained candidate points based on height-jumping and slope features in each layer, these two features are mutually redundant which will contain too many false points and cannot remove road surface points effectively Hu *et al.* [13] extracted candidate points by curvature change and height difference in raw point cloud, which would contain too many obstacle points and increase computational burden. In [14] and [15], candidate points were extracted based on local normal saliency feature or supervoxels, which is not suitable for sparse point cloud In [16] line segment feature in sliding window was used to extract candidate points in each layer, which did not work well when point cloud became sparse in the distance.

All the aforementioned methods shared the drawback that a single or multiple redundant feature are adopted to extract candidate points, resulting in miss detection with strict threshold while false detection with loose threshold.

B. BRELATED WORK ON CANDIDATE POINTS CLASSIFICATION

For the problem of classification of left and right candidate points, most of researchers focused on the problem of straight roads in which left and right candidate points are classified just based on the direction of lateral coordinates, such as [12], [13], and [16]–[18]. Xu *et al.* [20] extracted candidate points based on energy function and refined left and right candidate points using least cost path model. This method can work well on curved road, but it needs add source point manually which cannot be used in practice. In [15] supervoxels were used to obtain candidate points, then the trajectory data was used to classify left and right candidate points which can only work offline and are not suitable for on-line applications in self-driving. In [21], in order to solve the problem of curb detection on curved road, a clustering method based on linear discriminant analysis (LDA) [22] was proposed to divide left and right candidate points. In [23] sliding-beam method was proposed to obtain road segmentation line by using off-ground data. And the road segmentation line was used to classify left and right candidate points. Both [21] and [23] can just linearly classify candidate points and they cannot work well on large curvature road.

All the aforementioned methods shared the drawback of being affected by road curvature, they all cannot correctly distinguish left and right candidate points

C. RELATED WORK ON CANDIDATE POINTS FILTERING

For the problem of candidate points filtering, Sun *et al.* [12] and Yang *et al.* [17] firstly classified candidate points into segments with k-nearest neighbor method, then the segments of less than three points were eliminated, which may filter out isolated curb points. In [13] RANSAC line fitting algorithm was used to filter out obstacle points, which would filter out curb points when road was curved In [18] a regression filter was introduced to make detection robust to occlusions. Both of [13] and [18] could only remove obstacle points inside road, however, they cannot deal with obstacles outside road, such as the situation in which multiple roads are distributed in parallel. Chen *et al.* [19] filtered candidate points by Gaussian Process, which depended heavily on the accuracy of the selected seed points. In [14], candidate points were extracted based on local normal saliency, then the distance to trajectory was used to filter candidate points which is not suitable for practical application.

All the aforementioned methods cannot effectively filter out obstacles inside and outside road area, especially for scenarios in which there are obstacles with the same feature as road curbs, such as parallel roads, railway tracks.

In order to solve the above problems, we propose a new robust method to extract road curbs from point cloud which is not affected by obstacles and road curvature. The main contributions of this paper are as follows:

- We propose a multi-feature loose-threshold layered method for candidate points extraction which avoids miss detections and too many false detections due to single feature.
- We present a density-based clustering method for classifying left and right candidate points which can correctly classify left and right candidate points under various road curvature.
- We design a two-step filtering method to remove false points caused by obstacles inside and outside road.

The remainder of this paper is organized as follows. Section II describes the overview of our method for road curb detection and tracking. Section 3-4 describes each step of our method in detail. Section 8 evaluates the proposed method through comprehensive experiments. Section 9 summarizes the contributions of this paper.

II. METHOD OVERVIEW

As shown in Fig. 2, the proposed method consists of five main steps. It inputs a frame of point cloud acquired from 3D-LiDAR and outputs curb points and curb curves.

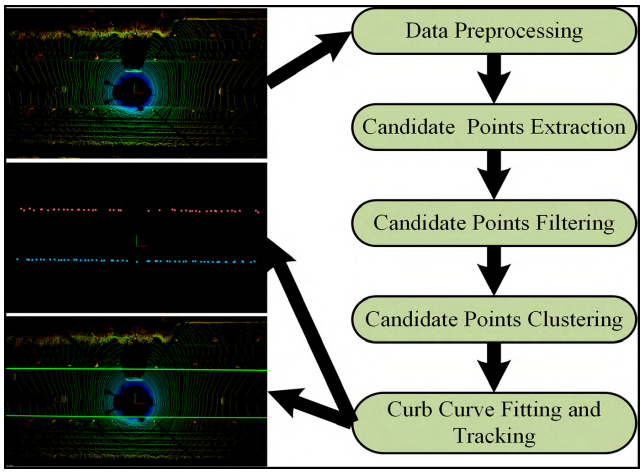

FIGURE 2. The pipeline of road curb detection and tracking method.

A. DATA PREPROCESSING

Because of vehicle motion and the principle of LiDAR, the raw point cloud is distorted. Thus, GPS/IMU is used to correct point cloud based on linear interpolation In order to remove obstacles above ground and reduce computational burden, a piecewise plane-based segmentation method is used to distinguish on-ground and off-ground points Only on-ground points are used for curb point extraction.

B. CANDIDATE POINTS EXTRACTION

A multi-feature loose-threshold layered method is proposed to increase the robustness of candidate points extraction. Firstly, four spatial features are defined based on road model. Then, the points that satisfy all spatial features are extracted.

C. CANDIDATE POINTS CLUSTERING

It is an important and difficult step to classify left and right candidate points for road curb detection, especially in the case of road with large curvature Thus, a density-based clustering method is proposed to classify left and right candidate points.

D. CANDIDATE POINTS FILTERING

There are still many false points in candidate points, including vehicles, pedestrians on road and buildings, railway tracks, adjacent roads. In order to filter out these false points, a two-step candidate points filtering method is proposed which consists of distance filter and RANSAC filter.

E. CURB POINTS FITTING AND TRACKING

In this step, least square method is used to fit quadratic curve based on curb points. In order to estimate road curb curve smoothly and robustly, an Amplitude-Limiting Kalman filter is used to smooth the fluctuation of road curb curve.

III. DATA PREPROCESSING

The proposed method uses 3D-LiDAR Velodyne HDL-64E sensor which features up to 64 lasers vertically aligned from $+2°$ to $-24.8°$, and its rotating delivers a $360°$ horizontal field of view. It generates a point cloud of 1000,000 points per second with a range of 120 m and typical accuracy of 5 cm [24].

In this paper, the Velodyne sensor is mounted on the top of the AV. The cartesian coordinate system we define is shown in Fig 3 Taking the center of the LiDAR as origin point, the x axis points to the direction of driving, and the z axis is vertically upward. The y axis points to the left side of the direction of the AV forward direction.

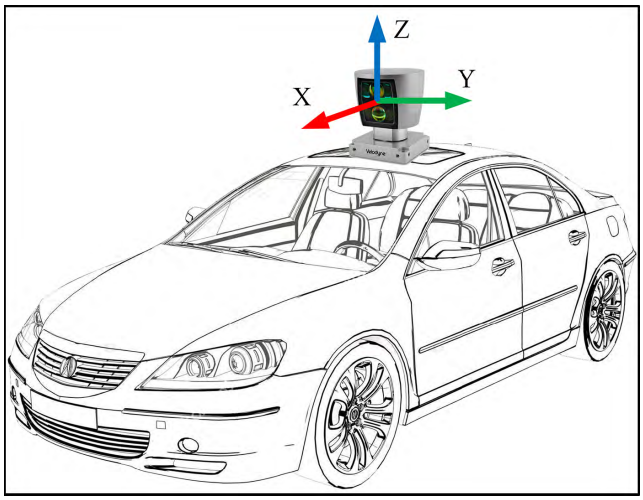

FIGURE 3. The cartesian coordinate system of LiDAR.

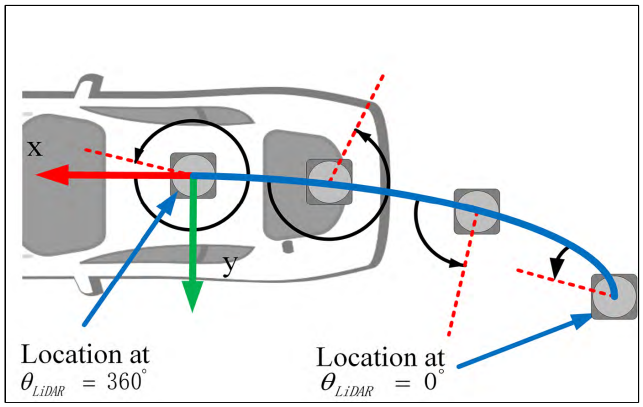

FIGURE 4. The car motion is added to the laser rotation. Each point in the data is not in the same coordinate system.

A. DISTORTION CORRECTION

According to the principle of LiDAR, it takes a certain period of time for LiDAR swiping environment a circle. During a scan period C , a mounted LiDAR can move, resulting in a non-single viewpoint measurement. As shown in Fig 4, the vehicle motion can be added to laser rotation causing the LiDAR measurement to be distorted [25]. So, if the motion of vehicle is not considered, the distorted 3D point cloud will affect curb points extraction To solve this problem, we transform all points into ending position in a frame.

Suppose that a scan period C is very short and the position and pose of vehicle changes linearly in a scan period.

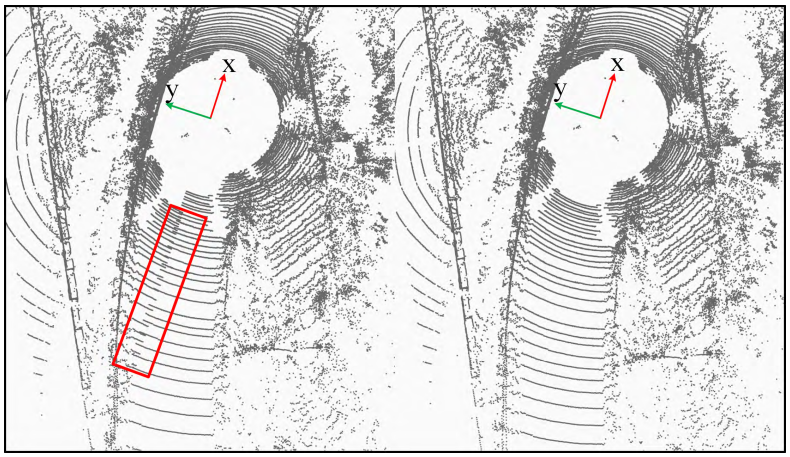

FIGURE 5. An example of point cloud distortion correction. (a) The point cloud before correction. (b) The point cloud after correction.

So, the pose and position of the vehicle at every moment can be obtained by the interpolation of the position and pose of between adjacent two frames. Assuming that the ending horizontal rotation angle of current frame is θ_1, the time required for LiDAR rotation from p_i to ending position is:

$$t_i = C \times (\theta_1 - \theta_2)/2\pi \tag{1}$$

where, θ_2 is the horizontal rotation angle of p_i

Besides, the position and pose changes of each frame can be obtained through GPS/IMU navigation system, including roll α_{frame}, pitch β_{frame}, yaw γ_{frame} displacement $x_{frame}y_{frame}$ z_{frame} Thus, the displacement matrix and pose matrix from p_i to ending position of current frame are calculated by linear interpolation as follows:

$$T_i = \frac{t_i}{C}\left[\, x_{frame} \;\; y_{frame} \;\; z_{frame} \,\right]^T \tag{2}$$

$$\left[\, \alpha_{\theta_2} \;\; \beta_{\theta_2} \;\; \gamma_{\theta_2} \,\right] = \frac{t_i}{C}\left[\, \alpha_{frame} \;\; \beta_{frame} \;\; \gamma_{frame} \,\right] \tag{3}$$

Then, the point p_i correction process is as follows:

$$R_i = R_z(\gamma_{\theta_2})R_y(\beta_{\theta_2})R_x(\alpha_{\theta_2}) \tag{4}$$

$$p_i^{'} = R_i p_i + T_i \tag{5}$$

where R_x, R_y, R_z represent rotation matrices around x axis, y axis and z axis, respectively. $p_i^{'}$ is the point corrected. As it can be seen in the red box in Fig. 5(a), laser line is duplicated due to the rotation of vehicle. After distortion correction, the laser line is not duplicated anymore (Fig 4(b))

For convenience of description, let P denote the undistorted point cloud and each point be $p_{li} = [\, x_{li} \;\; y_{li} \;\; z_{li} \,]$ where l is the number of corresponding laser line.

B. GROUND SEGMENTATION

The proposed method in this paper focuses on road curb extraction from point cloud. Since road curb is a part of ground, we use ground segmentation to extract on-ground points and remove off-ground points which usually consist of trees, buildings and other objects.

Here we use a piecewise plane-based segmentation method to distinguish on-ground and off-ground points. First, point cloud P is divided into several segments based on x coordinate. Then, RANSAC (Random sample consensus) method is applied to fit plane in each segment [26]. By dividing point cloud into segments, the method can basically deal with slope road scenario. The extraction of on-ground points does not require high precision because the curb detection method will be further applied.

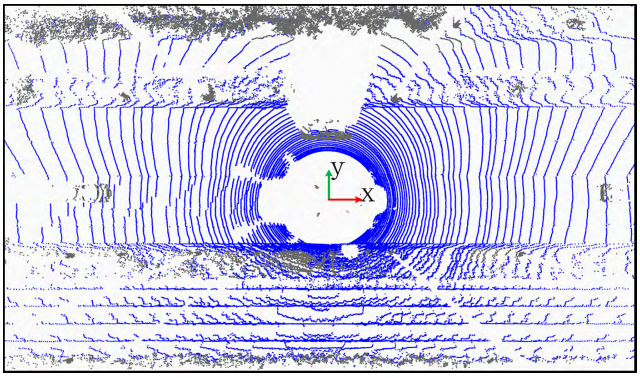

FIGURE 6. An example of ground segmentation. Where, on-ground points are shown in blue, off-ground points are shown in gray.

In our method, the ''rough'' on-ground points provide enough data for candidate points extraction in the next step and reduce most of outliers simultaneously. In Fig 6 is an example of ground segmentation method where on-ground points shown in blue are used to extract curbs and off-ground points are removed to reduce the cost of computation. For convenience, let P_{on} denote on-ground points and P_{off} denote off-ground points

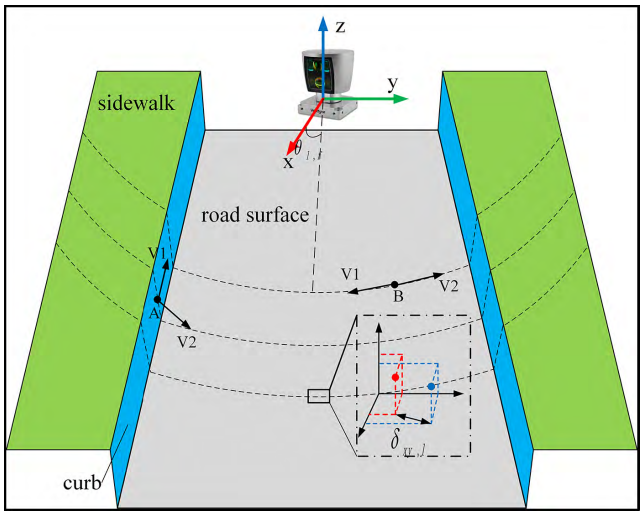

FIGURE 7. Road model.

IV. CANDIDATE POINTS EXTRACTION

A. ROAD CURB MODEL

In urban environments, it is clear that road surface is flat and sidewalks are higher on both sides. Road curbs are about 10 to 30 cm 10 cm higher than road surface. An ideal road model is shown in Fig. 7, in which the blue vertical planes represent the left and right road curb, the green horizontal planes represent sidewalks, and the gray horizontal plane represents road surface.

B. CANDIDATE POINTS EXTRACTION

HDL-64E has 64 lasers, scanning surrounding area in 360 degrees horizontally, each of which has its own vertical azimuth. Based on this, on-ground points P_{on} are divided into 64 scanning layers to extract candidate points.

The algorithm of candidate points extraction is mainly inspired by [12] and [23]. However, road curb is not always regular and the effects of various lasers are different because of varying scan ranges. When only a single feature is applied, miss and false detections are inevitable, particularly when road conditions are complex. Given that, a multi-feature loose-threshold layered method is proposed to extract candidate points In our work four spatial features are defined and each has a loose threshold The points that satisfy all spatial feature are extracted for each scanning layer The features applied in this paper are described in the following sections.

1) HEIGHT DIFFERENCE BETWEEN NEIGHBORING POINTS

As shown in Fig 7, there is always a height difference between road curb and road surface and the height of curb points of same laser layer will obviously increase. So, height difference and height standard deviation are selected as a feature of curb points. The maximum and minimum values of z coordinates of all points in the neighbors of p_{li} are represented as Z_{max} and Z_{min} respectively. Thus, the height difference feature is defined as:

$$T_{height1} \leq Z_{max} - Z_{min} \leq T_{height2}$$
$$\sqrt{\frac{\sum (z_{li} - \mu)^2}{n_{height}}} \geq T_{height3}$$

where $\mu = \frac{\sum z_{li}}{n_{height}}$, n_{height} is the number of neighbors, z_{li} is z values of each neighbors and l is the number of the corresponding laser line, $T_{height1}$, $T_{height2} and T_{height3}$ are the thresholds of height difference feature.

2) SMOOTHNESS OF NEIGHBORING POINTS

The smoothing feature proposed by [27] is used to describe the smoothness in the neighborhood of a point. For any point p_{li}, We use S to represent neighbors continuously collected on both sides of p_{li}. Thus, the smoothness feature is defined as:

$$s = \frac{1}{|S| \, \|P_{li}\|} \cdot \left\| \sum_{\substack{P_j \in S \\ j \neq i}} (p_{li} - p_{lj}) \right\|$$
$$s \geq T_{smoothness}$$

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

3) TANGENT VECTOR OF NEIGHBORING POINTS

The tangent vector feature proposed by [23] describes the angle between two vectors originating from the same point p_{li}. The angle θ_{li} is defined as:

$$\theta_{li} = cos^{-1} \frac{V_1 \cdot V_2}{|V_1| \cdot |V_2|}$$
$$\theta_{li} \leq T_{tangent}$$
$$V_1 = \left[\sum_{k=1}^{n} (x_{l,i-k}, x_{li}), \sum_{k=1}^{n} (y_{l,i-k}, y_{li}) \right]$$
$$V_2 = \left[\sum_{k=1}^{n} (x_{l,i+k}, x_{i}), \sum_{k=1}^{n} (y_{l,i+k}, y_{li}) \right]$$

where, V_1 and V_2 are the vectors originating from point p_{li}, n is the number of neighbors $T_{tangent}$ is threshold of tangent vector feature. The angle between two vectors on road curb is less than the angle between two vectors on road surface. An example is shown in Fig 7, where A is a curb point, B is a road surface point, there is an obvious difference between the tangent vector angles of point A and point B.

4) HORIZONTAL DISTANCE OF ADJACENT POINTS

The horizontal distance feature proposed by [23] represents the horizontal distance between two adjacent points in the same laser line, as show in Fig 7. It sets the horizontal distance threshold $\delta_{xy,l}$ when the point p_{li} is on the flat road surface.

It is defined by

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

where, H_s is the absolute value of the height of the scanning point $p_{li}\theta_{l,f}$ is the vertical azimuth of scanning line, θ_a is the horizontal angular resolution of LiDAR If p_{li} is selected as candidate curb point, the horizontal distance between p_{li} and its adjacent point should be larger than $\delta_{xy,l}$.

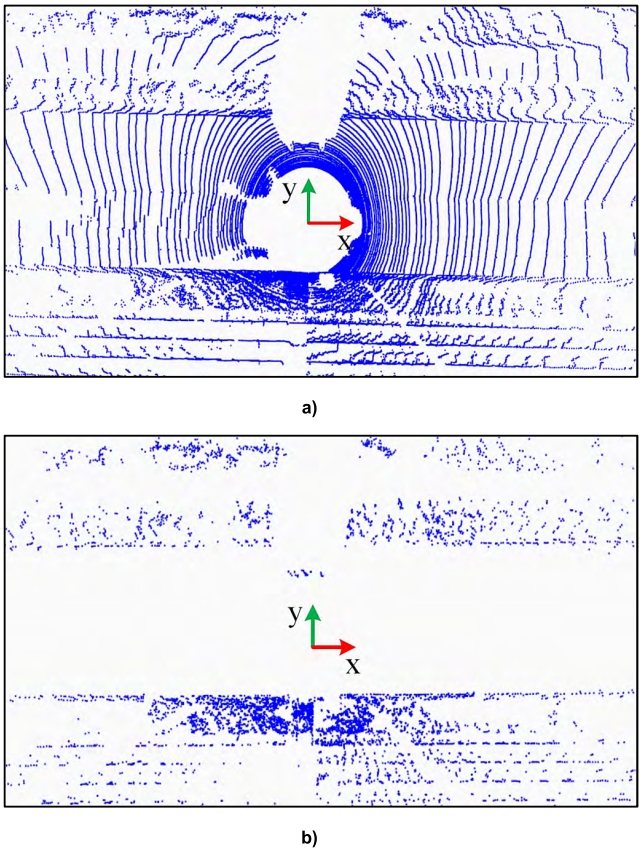

FIGURE 8. An example of candidate curb point extraction. a) On-ground points. b) Candidate points.

Then all of the features are tested in the experiment and loose thresholds are determined Since multiple features with loose thresholds are used to extract candidate points, miss detection can be well avoided and road surface points can be removed very well even on slope and curved road. After candidate points extraction, the extracted candidate points $P_{candidate}$ are obtained, as shown in Fig 8

V. CANDIDATE POINTS CLUSTERING

In order to filter out false points in $P_{candidate}$, candidate points first need to be clustered into two classes, representing left and right candidate points respectively. As far as I know, there are very few effective methods to distinguish left and right candidate points, especially for curved road. Most of literatures simply classify left and right candidate points according to lateral coordinates.

In order to solve the problem, a density-based clustering method is proposed for classifying left and right candidate points which is inspired by the DPCA (Density Peaks Clustering Algorithm) [28] and DBSCAN (Density-based spatial clustering of applications with noise) [29] The DPCA method cannot effectively cluster candidate points into left and right candidate points. Because each point is assigned to the same cluster as its nearest neighbor of higher density [28], when candidate points on one side are sparse and road width is small, it is likely that the clustering will fail Besides, the DBSCAN cannot determine cluster center and number of clusters. The clustering algorithm proposed in this paper combines the advantages of these two methods and overcomes their disadvantages at the same time which can effectively cluster candidate points into left and right candidate points.

A. CLUSTERING ALGORITHM EXPLAIN

The clustering algorithm consists of two steps: determination of cluster centers and assigning each point to clusters. The first step is inspired by DPCA and has basis in the assumptions that cluster centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any points with a higher local density [28]. For convenience we make two definitions as follows:

Definition 1: (local density) The *local density* ρ_i of point i is computed as

$$\rho_i = \sum_j \chi(d_{ij} - d_c)$$

where χ (x) = 1 if x<0 and χ (x) = 0 otherwise. Basically, ρ_i is equal to the number of points that are closer than d_c to point i d_{ij} is the distance between any two points, d_c is neighborhood threshold

Definition 2: (nearest higher density distance) The *nearest higher density distance* of point i is measured by computing the minimum distance between the point i and any other point with higher density:

$$\delta_i = \min_{j:\rho_j > \rho_i}(d_{ij})$$

For the point with highest density, we conventionally take $\delta_i = max_j(d_{ij})$.

The points that has both large *local density* and *nearest higher density distance* are selected as cluster centers. An example is shown in Fig.9 point 1 and point 9 are selected as cluster centers.

The second step is inspired by DBSCAN which classified points based on *density-reachable* principle. For convenience, we make three definitions:

Definition 3: (neighbors of a point) The *neighbors* of point i denoted by N_i is defined as follows:

$$N_i = \{j \in D | d_{ij} \leq d_c\}$$

where D is input dataset and $|N_i| = \rho_i$

Definition 4 (Directly Density-Reachable): The point i is *directly density-reachable* from point j and point j is core point if $i \in N_j$ and $\rho_j \geq n_{min}$, n_{min} is core point threshold.

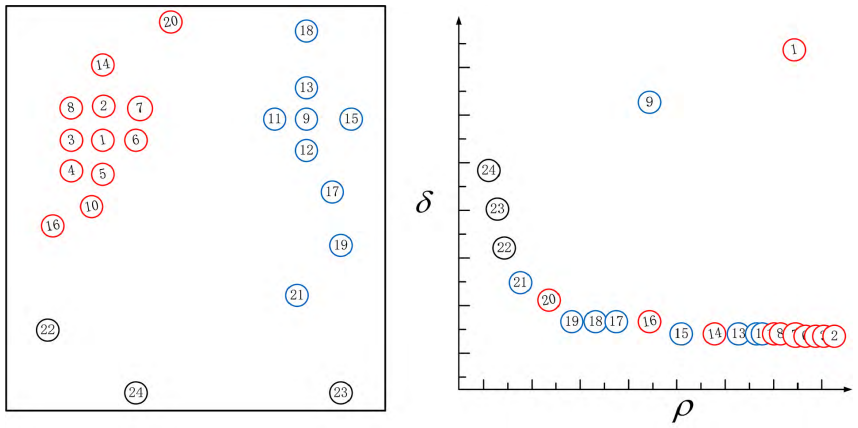

FIGURE 9. a) Point distribution. The number represents the local density order of each point. b) Decision graph for the data. (Different colors correspond to different clusters).

Definition 5 (Density-Reachable): The point i is *density-reachable* from point j if there is a chain of points $i_1, \cdots, i_n, i_1 = j, i_n = i$ such that i_{k+1} is directly density-reachable from i_k

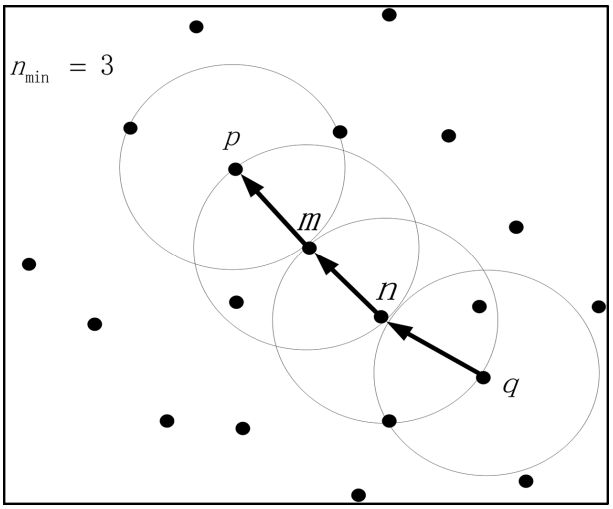

FIGURE 10. Density-reachable and directly density-reachable.

As shown in Fig. 10, point p is *directly density-reachable* from m, and *density-reachable* from n and q. After the cluster center is determined, all the points which is density-reachable from the same cluster center are grouped into one class.

B. CURB CLUSTERING ALGORITHM PROCESS

Input:

the candidate points $P_{candidate}$, neighborhood threshold d_c, core point thresholdc n_{min}

Output:

left candidate points $P_{candidate,l}$ and right candidate points $P_{candidate,r}$.

Process:

1. Determination of cluster centers.

a) For each point i, compute *local density* ρ_i based on neighborhood threshold d_c.

b) For each point i, compute *nearest higher density distance* δ_i.

c) For each point, compute decision value $\gamma_i = \rho_i \delta_i$, select two points $p_{center,l}$,$p_{center,r}$ with the largest γ_i as the cluster centers for left and right candidate points, where the sign of y coordinates of these two centers are opposite.

2. Assign each point to cluster.

a) Select all the points that $\rho_i \geq n_{min}$ as core points to form core point set P_{core}. According to step 1, the two cluster centers have large *local density* so add them to the core point set P_{core}.

b) In core point set P_{core}, compute *neighbors* $N_{core,i}$ for each point i based on threshold d_c.

c) In candidate point set $P_{candidate}$, compute *neighbors* $N_{candidate,i}$ for each core point i.

d) Choose $p_{center,l}$ as seed point, retrieve all points in P_{core} that are *density-reachable* from the seed to obtain the left core point set $P_{core,l}$ based on $N_{core,i}$ for each core point i.

e) For each core point i in $P_{core,l}$, first add core point i to $P_{candidate,l}$, then add each point j in $N_{candidate,i}$ to $P_{candidate,l}$ if point j is not clustered.

f) Choose $p_{center,r}$ as seed point, repeat step **d** and **e** to obtain $P_{candidate,r}$.

The parameter for clustering algorithm is neighborhood threshold d_c and core point threshold n_{min} The threshold d_c should be less than half of the width of road To prevent missing isolated curb points, the threshold n_{min} should be chosen as small as possible.

C. CLUSTERING EXAMPLES

The clustering algorithm selects the cluster centers by using the idea of DPCA, and overcomes the defect of assigning

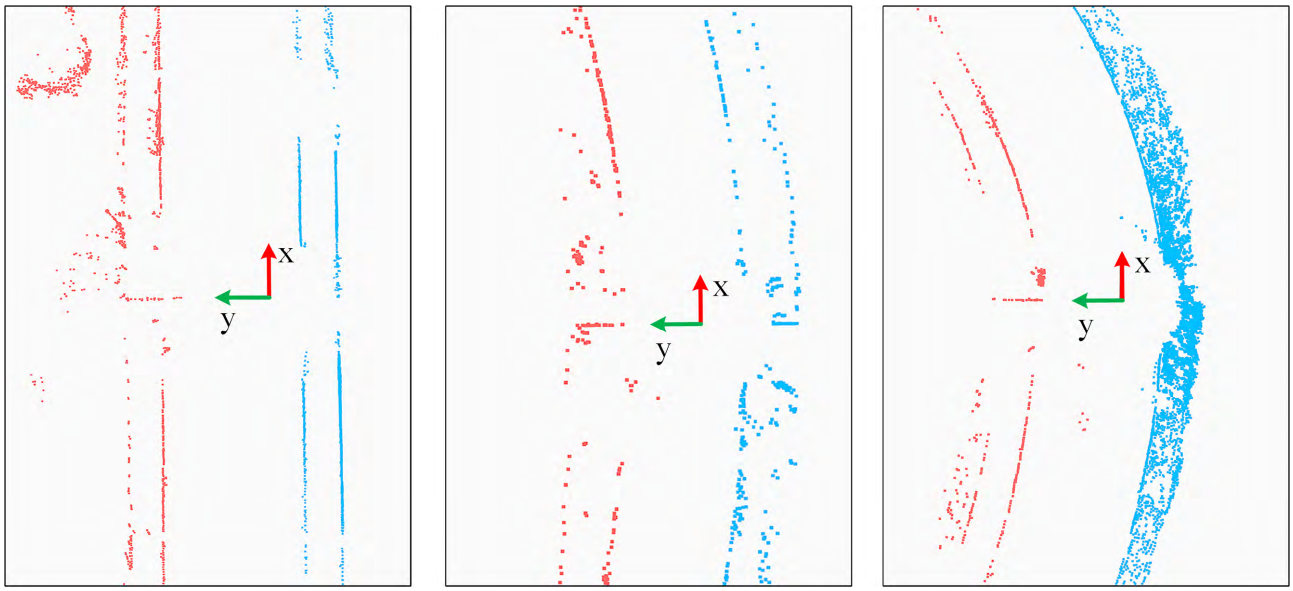

FIGURE 11. Results of candidate points clustering on three different road curvatures.

point to the same cluster as its nearest neighbor of higher density by using the idea of DBSCAN.

Results of our clustering algorithm about candidate points clustering on three different curvature road are shown in Fig. 11. The red points are left candidate points and the blue points are right candidate points. As we can see, the algorithm proposed in this paper can correctly classify left and right road candidate points under different road curvature conditions.

VI. CANDIDATE POINTS FILTERING

After candidate points extraction, there are also a variety of false points, including vehicles, pedestrians on road and buildings, railway tracks, adjacent roads, as shown in Fig 8. So, a two-step filtering method is proposed to filter out false points inside and outside road, which consists of distance filter and RANSAC filter.

A. DISTANCE FILTER

The distance filter is used to remove false points caused by objects outside road whose feature is similar to curbs (e.g. buildings, railway tracks and parallel roads). Based on the fact that curbs are usually the nearest obstacles of AV on road, the distance filter searches the lateral nearest points to AV. The first step of distance filter is to divide candidate points of same laser line into two quadrants based on x value. For each quadrant the closest point relative to longitudinal axis (x) is preserved. The following equation represents the distance filter applied in each laser's quadrant:

$$f_{dist}\left(q_{i,j}\right) = \underset{p_{i,j}}{argmin} \left|y_{i,j}\right|$$

where $q_{i,j}$ is the j-th quadrant of i-th laser and $y_{i,j}$ is y value of each candidate points. For each laser quadrant one nearest point is preserved. The result of distance filter for $P_{candidate,l}$ and $P_{candidate,r}$ is $P_{filter_{dis},l}$ and $P_{filter_{dis},r}$, as shown in Fig 12.

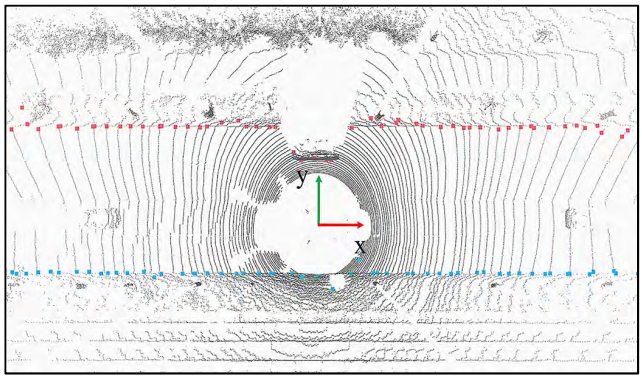

FIGURE 12. An example of candidate points filtered by distance filter (left is red, right is blue).

B. RANSAC FILTER

When obstacles as pedestrians and cars are present in road, the distance filter will possibly detect them as curbs. These obstacles can cause occlusion to sensor and make difficult to identify actual curbs. The RANSAC filter is introduced to remove points that located inside road. The RANSAC algorithm based on the idea of random sampling consistency is used to estimate the quadratic polynomial model from candidate points filtered by distance filter and iterates continuously until the fitted model satisfies as many points as

possible [26]. After RANSAC filter, all the points whose distances from fitted model are less than threshold are preserved. The result of distance filter for $P_{filter_{dis},l}$ and $P_{filter_{dis},l}$ is $P_{curb,l}$ and $P_{curb,r}$, as shown in Fig 13, which is final curb points.

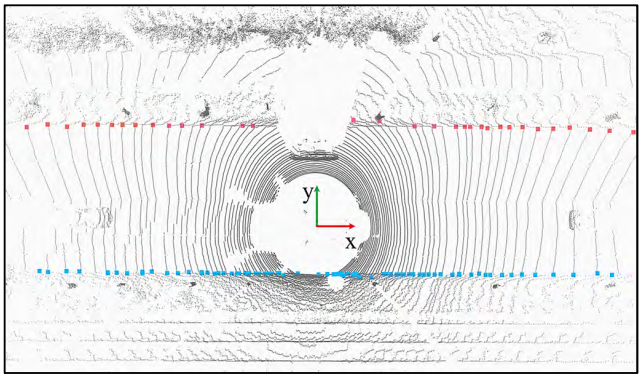

FIGURE 13. An example of curb points obtained by RANSAC filter (left is red, right is blue.

VII. ROAD CURB CURVE FITTING AND TRACKING

A. CURB CURVE FITTING

After curb points filtering, least square method is used to fit quadratic curve based on $P_{curb,l}$ and $P_{curb,r}$. The result is in the following form:

$$y = ax^2 + bx + c$$

The fitted road curb curve is represented as $L_l and L_r$. As show in Fig 14, road curb curves are in green.

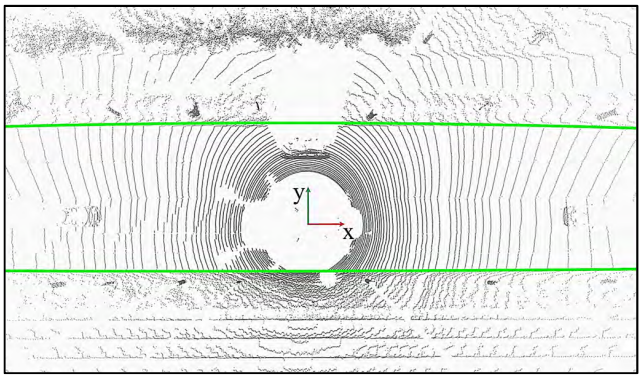

FIGURE 14. Result of curb curve fitting.

B. CURB CURVE TRACKING

The road curb curve obtained by least square method only reflects current road condition, which is not smooth. So, it is necessary to use Kalman filter to smooth the fluctuation of curb curve. We build a curb curve prediction model and curb curve measurement model based on accurate vehicle state data from GPS/IMU system.

1) AMPLITUDE-LIMITING FILTER

Because false detections will have serious consequences for the decision-making of AV, the Amplitude-Limiting filter in [13] is used to remove false detections. Only if the result of detection meets threshold requirement of Amplitude-Limiting filter, the current curb curve is considered to be effective and can be input to the update step of Kalman filtering. Otherwise, the prediction of the previous frame is used as final detection result. The requirement of Amplitude-Limiting filter is as follows:

$$\begin{aligned} \left|a_{previous} - a_{current}\right| &\le T_a \\ \left|b_{previous} - b_{current}\right| &\le T_b \\ \left|c_{previous} - c_{current}\right| &\le T_c \end{aligned}$$

where, $a_{previous}, b_{previous} and c_{previous}$ represent the road curb curves coefficients detected in previous frame. $a_{current}, b_{current} and c_{current}$ represent road curb curves coefficients detected in current frame.

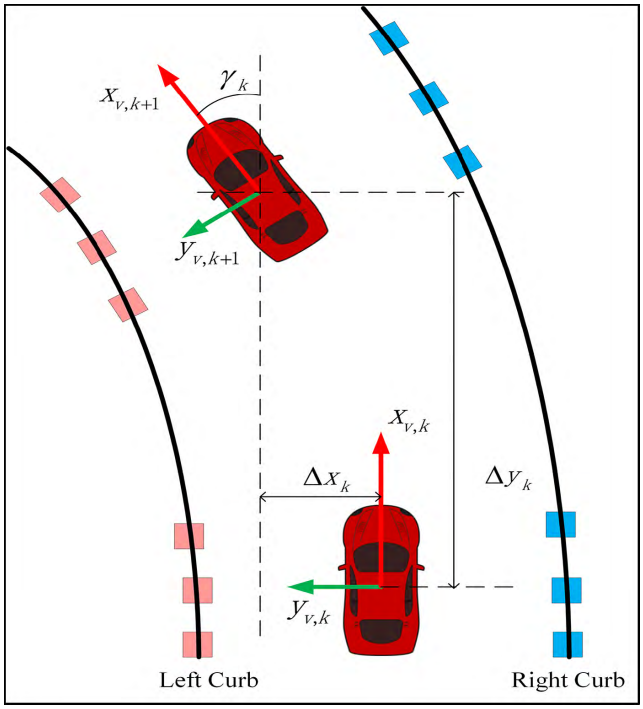

FIGURE 15. The coordinate frames of curb tracking.

2) CURB CURVE PREDICTION MODEL

The curb curve prediction model proposed in [30] is used for tracking. As shown in Fig 15, the two coordinate frames $(x_{v,k}y_{v,k})$ and $(x_{v,k+1}y_{v,k+1})$ represent vehicle coordinate at time k and $k+1$ respectively for derivation of the curb curve tracking algorithm. In order to track the fitted curb curve, we choose three different points on the fitted curve L_l and L_r to track which form points D_l and D_r. The three points have constant x value.

The curb curve prediction model is defined as follows:

$$\mathbf{x}(k+1) = \mathbf{A}(k)\,\mathbf{x}(k) + \mathbf{B}(k)\,\mathbf{u}(\mathbf{k}) + \boldsymbol{\upsilon}(\mathbf{k}) \qquad (6)$$

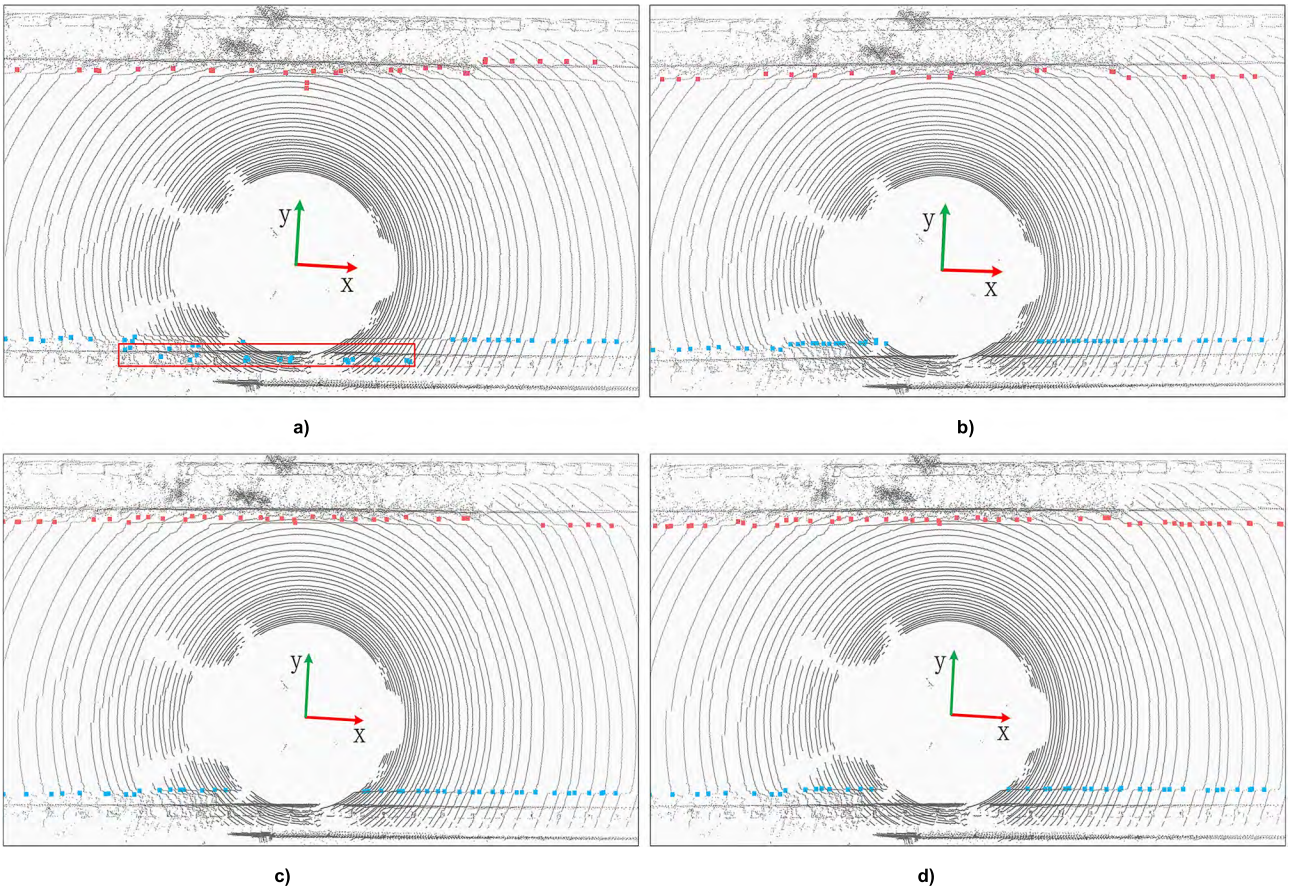

FIGURE 16. The results of four methods on straight road scenario (left is red, right is blue). In the red box is false points. a) Kang. b) Zhang. c) Zai. d) Proposed.

where

$$\mathbf{x}\,(\mathrm{k}) = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix}$$

$$\mathbf{u}\,(\mathrm{k}) = \begin{bmatrix} \Delta x(k) \\ \Delta y(k) \end{bmatrix}$$

$$\mathbf{A}\,(\mathrm{k}) = \begin{bmatrix} -\cos(\gamma_k) & \sin(\gamma_k) \\ -\sin(\gamma_k) & -con(\gamma_k) \end{bmatrix}$$

$$\mathbf{B}\,(\mathrm{k}) = \begin{bmatrix} -\cos(\gamma_k) & \sin(\gamma_k) \\ -\sin(\gamma_k) & -con(\gamma_k) \end{bmatrix}$$

x (k) and y(k) are coordinates of points in D_l and D_r, $\Delta x\,(k)$ *and* $\Delta y(k)$ are moving distance between two frames of x-axis and y-axis, γ_k is yaw angle of AV Then the predicted point position can be obtained by Equation (6) which relies on the precision of GPS/IMU system.

3) CURB MEASUREMENT MODEL

After the points D_l and D_r are obtained the measurement model is represented by

$$\mathbf{z}\,(\mathrm{k}) = \mathbf{x}\,(\mathrm{k}) + \boldsymbol{\nu}(\mathrm{k}) \tag{7}$$

where $\mathrm{z}\,(\mathrm{k}) = \left[\, z_x(k)\ z_y(k) \,\right]^T$ is the measurement vector and the measurement noise $\boldsymbol{\nu}(\boldsymbol{k})$ is white Gaussian noise with a covariance matrix $\mathbf{R}$

4) TRACKING ALGORITHM PROCEDURE

Based on the prediction and measurement model above, a Kalman filter is used to track curb curve to improve the smoothness of curbs. The tracking algorithm is carried out by the following steps:

1) Predict state and error covariance matrix at time k + 1 based on result of time k

$$\mathbf{x}(\mathbf{k}+\mathbf{1}|\mathbf{k}) = \mathbf{A}\,(\mathrm{k})\,\mathbf{x}\,(\mathrm{k}) + \mathbf{B}\,(\mathrm{k})\,\mathbf{u}\,(\mathrm{k}) + \boldsymbol{\omega}(\mathrm{k}) \tag{8}$$

$$\mathbf{P}\,(\mathrm{k}+1\,|\,\mathrm{k}) = \mathbf{A}\,(\mathrm{k})\,\mathbf{P}\,(\mathrm{k})\,\mathbf{A}^T\,(\mathrm{k}) + \mathbf{Q} \tag{9}$$

where, $\mathbf{x}(\mathbf{k}+\mathbf{1}|\mathbf{k})$ is the prediction of state vector and $\mathbf{x}(\mathrm{k})$ is the current state vector. $\mathbf{P}\,(\mathrm{k}+1\,|\,\mathrm{k})$ is the prediction of error covariance and $\mathbf{P}(\mathrm{k})$ is the error covariance of current state. The process noise $\boldsymbol{\omega}(\mathrm{k})$ is white Gaussian noise with covariance matrix $\mathbf{Q}$

2) If the detection result does not meet the threshold requirements of Amplitude-Limiting filter, $\mathbf{x}\,(\mathrm{k}+1|\mathrm{k})$ is the final result of the tracking algorithm. Otherwise, update the

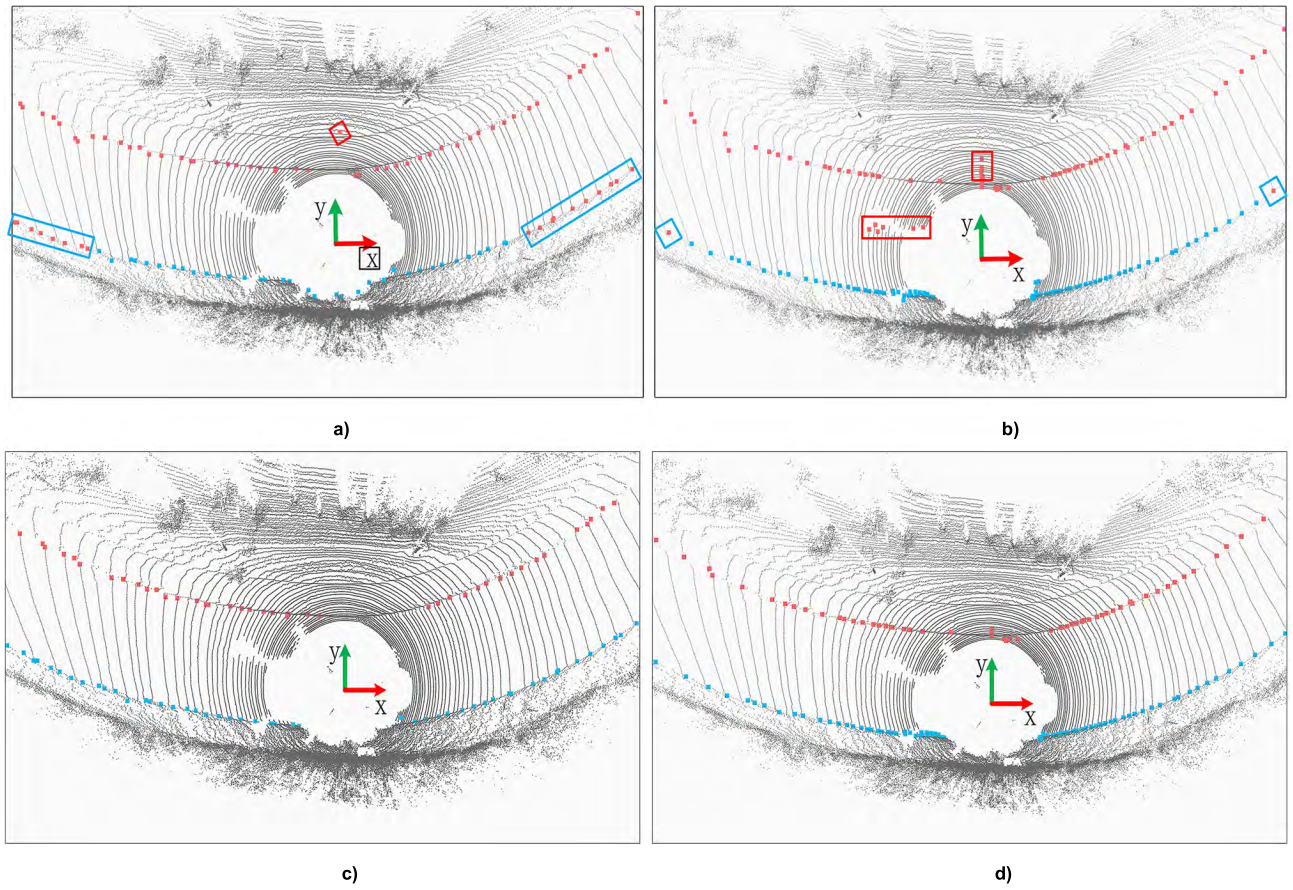

FIGURE 17. The results of four methods on curved road scenario (left is red, right is blue). In the red box is false points and in the blue box is the point that has been misclassified. a) Kang. b) Zhang. c) Zai. d) Proposed.

state and the error covariance matrix as follows:

$$\mathbf{G}_K\,(k+1) = \mathbf{P}\,(\mathrm{k}+1\,|\,\mathrm{k}) \cdot [\mathbf{P}\,(\mathrm{k}+1\,|\,\mathrm{k}) + \mathbf{R}]^{-1} \tag{10}$$

$$\mathbf{x}\,(\mathrm{k}+1) = \mathbf{x}\,(\mathrm{k}+1\,|\,\mathrm{k}) + \mathbf{G}_K\,(\mathrm{k}+1) \times (\mathbf{z}\,(\mathrm{k}+1) - \mathbf{x}(\mathbf{k}+\mathbf{1}|\mathbf{k})) \tag{11}$$

$$\mathbf{P}\,(\mathrm{k}+1) = \mathbf{P}\,(\mathrm{k}+1\,|\,\mathrm{k}) - \mathbf{G}_K\,(k+1)\,\mathbf{P}\,(\mathrm{k}+1\,|\,\mathrm{k}) \tag{12}$$

where, $\mathbf{G}_K\,(\mathrm{k}+1)$ is the Kalman filter gain, $\mathbf{x}\,(\mathrm{k}+1)$ is the final output of the tracking algorithm. After all the points among D_l and D_r are tracked using Equations (8)-(12), the road curb curves of two sides are obtained.

VIII. EXPERIMENT RESULTS

To test the robustness and the effectiveness of the proposed method, experiments were conducted for both qualitative and quantitative evaluation. The KITTI and Udacity dataset [31] and [32] are used to evaluate the proposed algorithm. These two datasets mainly cover urban and highway scenarios. However, the datasets have no ground truth for road curbs We have manually labeled curbs in each frame. The proposed method is tested and evaluated on four typical scenarios (straight road, curved road, obstacle inside road, obstacle outside road). The algorithm is implemented by C + + and PCL on Ubuntu 16.04. All experiments were performed on a 3.70 GHz Intel Core i7-8700K processor with 16 GB of RAM.

In order to demonstrate the effectiveness of the proposed method, the following state-of-the-art methods are employed for comparison:

Kang *et al.* [5]: The Hough transformation method.

Zhang *et al.* [23]: The sliding-beam and feature based search method.

Zai *et al.* [15]: The supervoxel generation and graph-cut method.

A. QUALITATIVE EVALUATION OF THE ROAD CURB EXTRACTION ALGORITHM

The value of parameters in our curb detection and tracking algorithm are defined in Table 1.

In straight road scenario (Fig.16), the results of all methods are similar because of the simplicity of the scenario except that Kang's has false points. Besides, we have less miss detection because of multi-feature loose-threshold method. In curved road scenario (Fig.17), we can see that Kang's could not distinguish left and right curb points accurately just by lateral coordinates. Zhang's has fewer points of error classification, it also cannot work well when the road

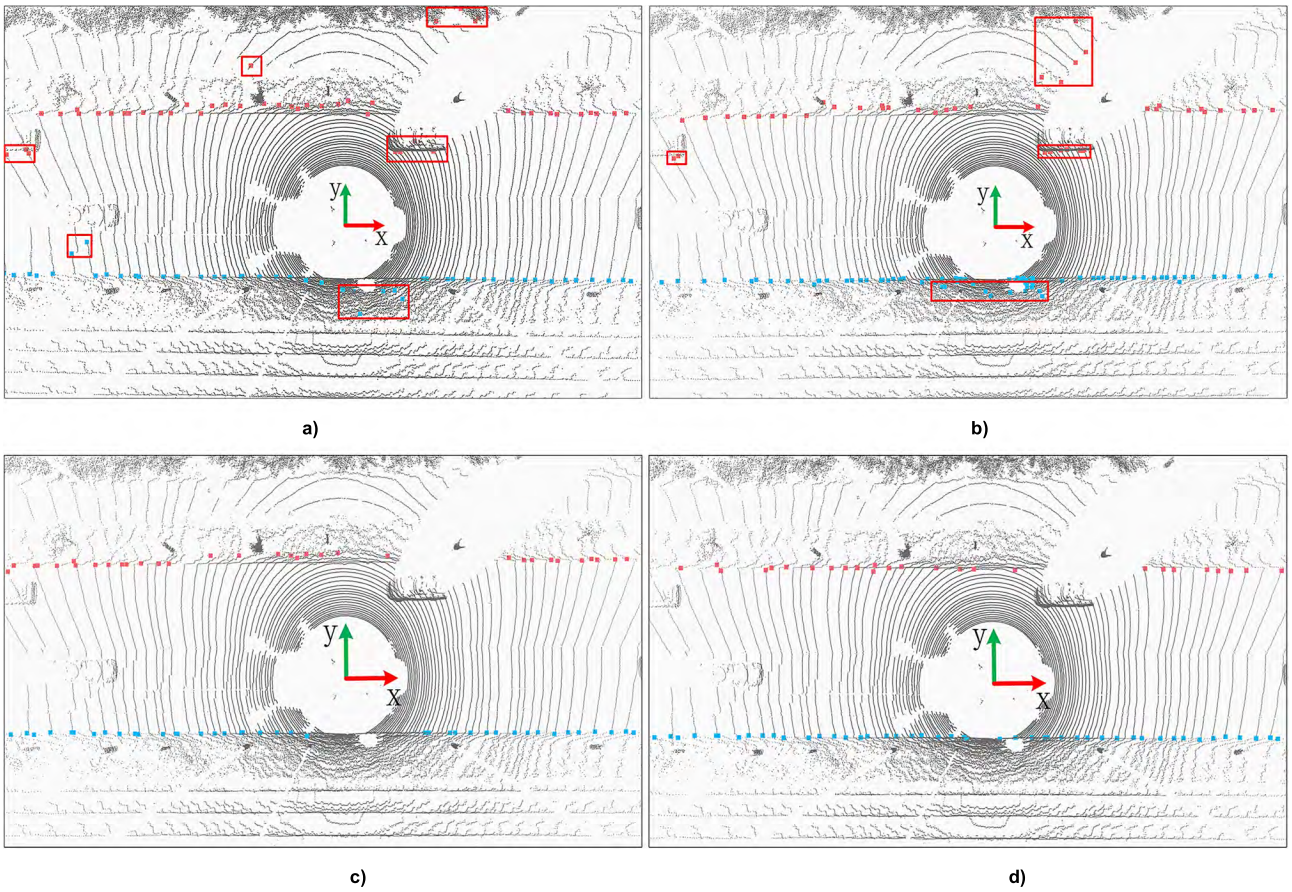

FIGURE 18. The results of four methods on obstacle inside road scenario (left is red, right is blue). In the red box is false points. a) Kang. b) Zhang. c) Zai. d) Proposed.

TABLE 1. Parameters for experiment.

Parameters	Value
$T_{height1}$	0.03
$T_{height2}$	0.30
$T_{height3}$	0.01
$T_{smoothness}$	0.005
$T_{tangent}$	170
d_c	2.5
n_{min}	1
T_a	0.005
T_b	0.5
T_c	0.8

curvature increases further. And Kang's and Zhang's have false points. Zai's and our method can correctly classify left and right curb points. In obstacle inside and outside road scenario (Fig. 18 and Fig.19) we can see that Zhang's and Kang's method cannot remove false points caused by obstacles inside and outside road. By contrast, our method has performed better because of curb points filtering proposed in this paper. Zai's method also achieves accurate detection result.

The qualitative evaluation experiments illustrate that the proposed method is accurate and robust for various road scenarios. Kang's and Zhang's methods cannot deal with false points because of the lack of filtering process. Besides, Kang's method is poor for distinguish left and right curb points just by lateral coordinates. Compared with Zai's method, we have less miss detection and Zai's method need trajectory data which is not suitable for practical application.

B. QUANTITATIVE EVALUATION OF THE ROAD CURB EXTRACTION ALGORITHM

Four different methods are quantificationally evaluated on two datasets, and 500 typical frames are selected for each scenario. In order to quantificationally evaluate our algorithm, three quantitative metrics in [33] and [34] are introduced for a comprehensive evaluation. For convenience, the distance threshold of true detection is set to 5 cm owing to labeling error.

1) ***Precision***, which denotes the proportion of curbs detected correctly in the curbs detected on one side. The *Precision* of one frame is the average of the *Precision* values of two sides.

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

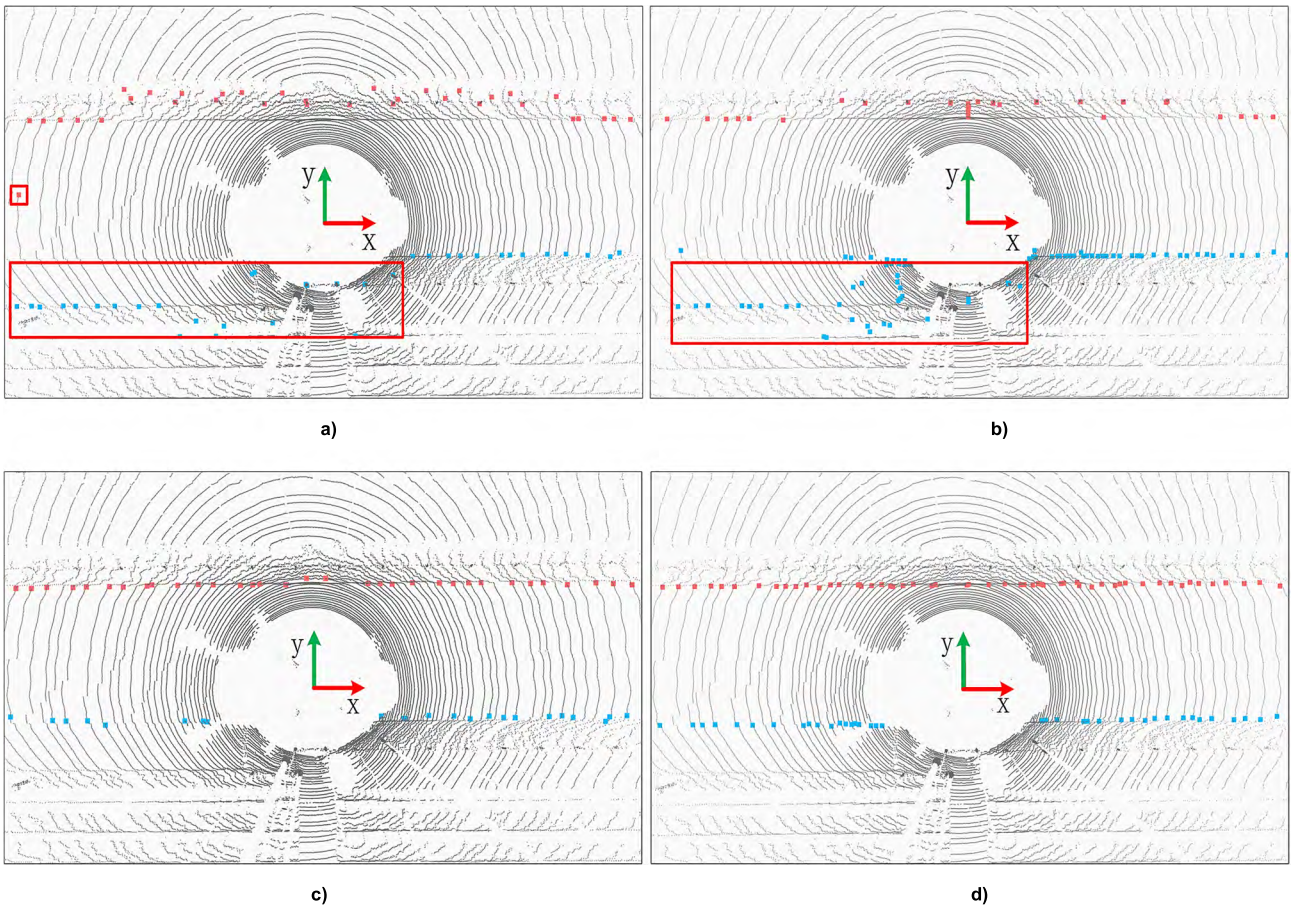

FIGURE 19. The results of four methods on obstacle outside road scenario (left is red, right is blue), In the red box is false points. a) Kang. b) Zhang. c) Zai. d) Proposed.

TABLE 2. Quantitative evaluation results in kitti.

Mean±Std.	Methods	Straight road	Curved road	Obstacle in road	Obstacle out road
Precision	proposed	**0.9214±0.0251**	**0.9203±0.0372**	**0.9146±0.0673**	**0.9238±0.0532**
	Kang [5]	0.7574±0.0271	0.5394±0.0431	0.5193±0.0481	0.6072±0.0691
	Zhang [23]	0.8594±0.0697	0.7864±0.0390	0.5394±0.0616	0.6373±0.0637
	Zai [15]	0.9090±0.0428	0.9014±0.1052	0.9012±0.0728	0.9109±0.0361
Recall	proposed	0.8524±0.0451	**0.7695±0.0376**	0.7519±0.0471	**0.7445±0.0548**
	Kang [5]	0.7726±0.0366	0.6290±0.0468	0.6994±0.0534	0.6534±0.0661
	Zhang [23]	**0.8628±0.0275**	0.7616±0.0550	0.7334±0.0562	0.6734±0.0308
	Zai [15]	0.8501±0.0506	0.7594±0.0245	**0.7637±0.0629**	0.7392±0.0924
F_1	proposed	**0.8856±0.0460**	**0.8382±0.0362**	0.8253±0.0550	**0.8245±0.0308**
	Kang [5]	0.7649±0.0540	0.5808±0.0335	0.5960±0.0543	0.6295±0.0634
	Zhang [23]	0.8611±0.0061	0.7738±0.0167	0.6916±0.0253	0.6549±0.0390
	Zai [15]	0.8786±0.0440	0.8243±0.0488	**0.8268±0.0655**	0.8161±0.0251

where TP is the true positive numbers and FP is the false positive numbers (false alarm).

2) ***Recall***, which denotes the proportion of the curbs detected correctly in the labeled curbs on one side. The *Recall* of one frame is the average of the *Recall* values of two sides.

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

where FN is false negative (missed detection).

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

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

The quantitative evaluation and comparison are given in Table 2 and 3 It is clear that our proposed method is robust and achieves the best performance in the two datasets Zai's method also obtains good result, however it needs trajectory

TABLE 3. Quantitative evaluation results in udacity.

Mean±Std.	Methods	Straight road	Curved road	Obstacle in road	Obstacle out road
Precision	proposed	**0.9094±0.0467**	**0.8973±0.0460**	**0.8946±0.0458**	**0.8838±0.0239**
	Kang [5]	0.7283±0.0356	0.5327±0.0335	0.5597±0.0285	0.6768±0.0476
	Zhang [23]	0.8842±0.0636	0.8796±0.0167	0.7596±0.0851	0.7737±0.0952
	Zai [15]	0.8965±0.0974	0.8832±0.0251	0.8712±0.0759	0.8791±0.0187
Recall	proposed	0.8427±0.0261	0.8197±0.0325	**0.8029±0.0274**	**0.7895±0.0627**
	Kang [5]	0.7846±0.0362	0.6590±0.0458	0.6394±0.0837	0.6184±0.0258
	Zhang [23]	**0.8538±0.0253**	0.7216±0.0543	0.7164±0.0308	0.7338±0.0249
	Zai [15]	0.8401±0.0748	**0.8234±0.0247**	0.7937±0.0390	0.7798±0.0374
F_1	proposed	**0.8748±0.0285**	**0.8567±0.0390**	**0.8463±0.0275**	**0.8340±0.0268**
	Kang [5]	0.7554±0.0253	0.5892±0.0353	0.5969±0.0506	0.6463±0.0366
	Zhang [23]	0.8687±0.0673	0.7928±0.0655	0.7374±0.0634	0.7532±0.0467
	Zai [15]	0.8674±0.0747	0.8523±0.0461	0.8306±0.0472	0.8265±0.0488

data as input. In our experiment, we do not have the future trajectory of vehicle.

In addition, Zhang's and Kang's method achieve poorer performance for obstacle inside and outside road scenario than straight road scenario, because the influence of obstacles inside and outside road is not considered, so the robustness is poor. Besides, Kang's achieves the worst performance, because it cannot also correctly classify left and right curb points in curved road scenario We can conclude that our proposed method can handle curb detection problem at complex scenarios and is more robust and achieves better performance.

IX. CONCLUSION

This paper presents a robust method for curb detection and tracking in structured environment. A multi-feature loose-threshold layered method is proposed for candidate points extraction. Based on candidate points, a density-based method is proposed for classifying left and right candidate points. Then, a two-step candidate points filter is used to remove false points caused by obstacles inside and outside road. Besides, the Amplitude-Limiting Kalman filter is used to solve the problem of false detection and miss detection. Comprehensive experiment evaluations clearly demonstrate that our proposed method achieves better robust and accurate.

REFERENCES

[1] F. Oniga, S. Nedevschi, and M. M. Meinecke, "Curb detection based on a multi-frame persistence map for urban driving scenarios," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2008, pp. 67–72.
[2] J. Siegemund, "Curb reconstruction using conditional random fields," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 203–210.
[3] L. Wang, "Multi-cue road boundary detection using stereo vision," in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Jul. 2016, pp. 1–6.
[4] Y. Maalej, "VANETs meet autonomous vehicles: Multimodal surrounding recognition using manifold alignment," *IEEE Access*, vol. 6, pp. 29026–29040, 2018.
[5] Y. Kang, C. Roh, S.-B. Suh, and B. Song, "A lidar-based decision-making method for road boundary detection using multiple Kalman filters," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4360–4368, Nov. 2012.
[6] S. Thrun *et al.*, "Stanley: The robot that won the DARPA grand challenge," *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, 2006.
[7] K. Buehler, Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, vol. 56. Berlin, Germany: Springer, 2009.
[8] G. Seetharaman, A. Lakhotia, and E. P. Blasch, "Unmanned vehicles come of age: The DARPA grand challenge," *Computer*, vol. 39, no. 12, pp. 26–29, Dec. 2006.
[9] S. Verghese, "Self-driving cars and lidar," in *Proc. CLEO Appl. Technol. Opt. Soc. Amer.*, 2017, p. AM3A.1.
[10] J. Fang. (2018). "Simulating LIDAR point cloud for autonomous driving using real-world scenes and traffic flows." [Online]. Available: https://arxiv.org/abs/1811.07112
[11] 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. 12th Int. Conf. Inform. Control, Automat. Robot. (ICINCO)*, vol. 1, Jul. 2015, pp. 191–198.
[12] S. Peng *et al.*, "A robust detection algorithm for urban road boundaries based on 3D lidar," *J. Zhejiang Univ.*, vol. 52, no. 3, pp. 504–514, 2018.
[13] K. Hu, "Real-time extraction method of road boundary based on three-dimensional lidar," *J. Phys. Conf. Series*, vol. 1074, no. 1, p. 4258, 2018.
[14] H. Wang *et al.*, "Road boundaries detection based on local normal saliency from mobile laser scanning data," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 10, pp. 2085–2089, Oct. 2015.
[15] D. Zai *et al.*, "3-D road boundary extraction from mobile laser scanning data via supervoxels and graph cuts," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 802–813, Mar. 2018.
[16] W. Yao, Z.Deng, and L. Zhou, "Road curb detection using 3D lidar and integral laser points for intelligent vehicles," in *Proc. 6th Int. Conf. Soft Comput. Intell. Syst.*, Nov. 2012, pp. 100–105.
[17] 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 J. Photogramm. Remote Sens.*, vol. 79, pp. 80–93, May 2013.
[18] A. Hata and D. Wolf, "Feature detection for vehicle localization in urban environments using a multilayer LIDAR," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 420–429, Feb. 2016.
[19] T. Chen, D. Dai, D. Liu, J. Song, and Z. Liu, "Velodyne-based curb detection up to 50 meters away," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2015, pp. 241–248.
[20] S. Xu, R. Wang, and H. Zheng, "Road curb extraction from mobile lidar point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 996–1009, Feb. 2017.
[21] L. Z. T. Z. R. Mingwu, "Algorithm of real-time road boundary detection based on 3D lidar," (in Chinese), *J. Huazhong Univ. Sci. Technol. (Natural Sci. Ed.)*, vol. 39, Sup. 2, 2011.
[22] S. Mike, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, "Fisher discriminant analysis with kernels," in *Proc. IEEE Signal Process. Soc. Workshop*, Aug. 1999, pp. 1–6.
[23] Y. Zhang, J. Wang, X. Wang, and J. M. Dolan, "Road-segmentation-based curb detection method for self-driving via a 3D-LiDAR sensor," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3981–3991, Dec. 2018.
[24] *Velodyne. HDL-64El.* Accessed: Jan. 19, 2019. [Online]. Available: https://velodynelidar.com/hdl-64e.html
[25] P. Merriaux. (2018). "LiDAR point clouds correction acquired from a moving car based on CAN-bus data." [Online]. Available: https://arxiv.org/abs/1706.05886
[26] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[27] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," *Robot. Sci. Syst.*, vol. 2, p. 9, Jul. 2014.
[28] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.

[29] M. Ester, "Density-based spatial clustering of applications with noise," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 1996, p. 240.
[30] Y. Zhang, J. Wang, X. Wang, C. Li, and L. Wang, "A real-time curb detection and tracking method for UGVs by using a 3D-LIDAR sensor," in *Proc. IEEE Conf. Control Appl. (CCA)*, Sep. 2015, pp. 1020–1025.
[31] P. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[32] (2017). *Udacity. Public Driving Dataset*. Accessed: Mar. 7, 2017. [Online]. Available: https://github.com/udacity/self-driving-car/tree/master/datasets
[33] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.
[34] Y. Sasaki *et al.*, "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 pursuing the Ph.D. degree in vehicle engineering with Jilin University, Changchun, China. His current research interests include environment perception, behavior estimation, and prediction.

JIAN WU is currently a Professor with the College of Automotive Engineering, Jilin University. He has authored over 40 peer-reviewed papers in international journals and conferences, and has been in charge of numerous projects funded by national government and institutional organizations on vehicles. His research interests include vehicle control systems, electric vehicles, and intelligent 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. His research interests include vehicle control systems, electric vehicles, and intelligent vehicles.

SHUN YANG received the B.S. degree in thermal energy and power engineering from Harbin Engineering University, Harbin, China, in 2013. He is currently pursuing the Ph.D. degree in vehicle engineering with Jilin University, Changchun, China. He is a Visiting Scholar with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA, USA, and with the Department of Psychology, Stanford University, Stanford, CA, USA. His current research interests include CNN-based autonomous vehicle control, object detection, and deep reinforcement learning-based autonomous vehicle control.