IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 18, 2020, accepted June 2, 2020, date of publication June 8, 2020, date of current version June 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3000777

Map-Enhanced Ego-Lane Detection in the Missing Feature Scenarios

XIAOLIANG WANG, YEQIANG QIAN, (Graduate Student Member, IEEE), CHUNXIANG WANG, AND MING YANG, (Member, IEEE)
Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China
Corresponding author: Ming Yang (mingyang@sjtu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant U1764264/61873165.

ABSTRACT As one of the most important tasks in autonomous driving systems, ego-lane detection has been extensively studied and has achieved impressive results in many scenarios. However, ego-lane detection in the missing feature scenarios is still an unsolved problem. To address this problem, previous methods have been devoted to proposing more complicated feature extraction algorithms, but they are very time-consuming and cannot deal with extreme scenarios. Different from others, this paper exploits prior knowledge contained in digital maps, which has a strong capability to enhance the performance of detection algorithms. Specifically, we employ the road shape extracted from OpenStreetMap as lane model, which is highly consistent with the real lane shape and irrelevant to lane features. In this way, only a few lane features are needed to eliminate the position error between the road shape and the real lane, and a search-based optimization algorithm is proposed. Experiments show that the proposed method can be applied to various scenarios and can run in real-time at a frequency of 20 Hz. At the same time, we evaluated the proposed method on the public KITTI Lane dataset where it achieves state-of-the-art performance. Moreover, our code will be open source after publication.

INDEX TERMS Ego-lane detection, missing feature, OpenStreetMap, parameter estimation.

I. INTRODUCTION

With the development of artificial intelligence, autonomous driving systems have become research hot-spots in both academia and industry. As one of the essential modules, ego-lane detection allows the car to properly position itself within the road lanes, which is crucial for subsequent control and planning.

A typical ego-lane detection result in the KITTI Lane dataset is shown in Figure 1, where the ego-lane is labeled as green. It can be seen that there are three main tasks for ego-lane detection: left boundary detection, right boundary detection, and upper boundary detection. The upper boundary detection is mainly to detect the preceding vehicle, which has been studied by most scholars in recent years and has achieved encouraging results. Therefore, this paper focuses on the left and right boundary detection, that is, lane line detection and road curb detection in KITTI Lane dataset (the road in the KITTI Lane dataset is a two-way road and the vehicle is driving on the right lane).

FIGURE 1. A typical ego-lane detection result in the KITTI Lane dataset, where the ego-lane is labeled as green.

For lane line detection and road curb detection, one of the most challenging scenarios is missing feature, which may be caused by lane marking wear, lighting changes, and even no visible features. To tackle this challenge, previous methods [1]–[4] have been devoted to proposing more effective feature extraction methods to obtain as many features as possible, but they are very time-consuming and cannot deal with extreme scenarios. In addition, model fitting plays an important role when features are partially missing or other

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

objects are interpreted as features [5]. Therefore, this paper focuses on obtaining the compact high-level representation of lane boundaries through model fitting, thereby solving the missing feature problem.

In recent decades of research, various mathematical representation models have been used for model fitting, ranging from simple straight line models to complex spline models. Many researchers prefer model fitting using straight line [6], [7], which is a good approximation for the short range and is the most common case in highway scenarios. Although the straight line model is efficient and simple, it will fail in curved roads, so some researchers propose to use a circular arc as lane model [8], [9]. Furthermore, quadratic polynomials [10] and cubic polynomials [11] are also widely used for model fitting in curved situations. In recent years, more and more researchers prefer to use splines for model fitting such as cubic spline [12], Catmull-Rom spline [13], and B-Spline [14]. Although mathematical representation models have been widely used for model fitting, their performance is profoundly affected by the quality of lane features. When in some extreme scenarios, the overfitting issue will occur, therefore causing a large shape error between the fitted lane and the real lane.

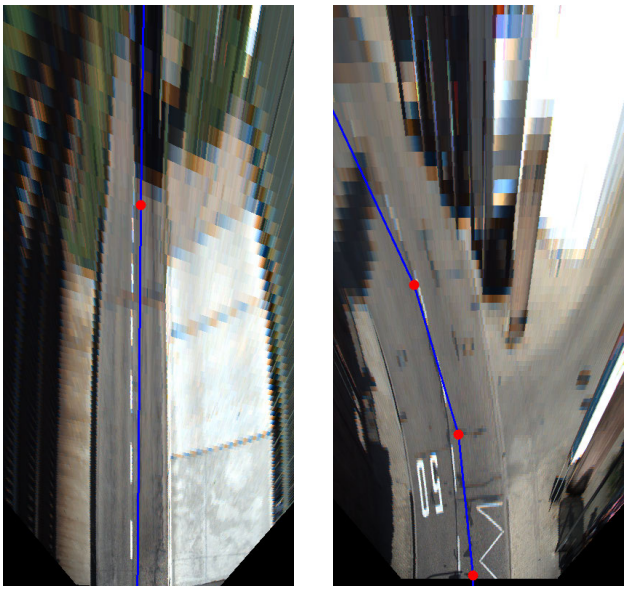
Nowadays, most autonomous driving systems have access to digital maps that contain rich geometric and semantic information about the environment. This prior information has been proven to have a strong capability to enhance the performance of algorithms in perception [15], prediction [16], and motion planning [17]. In this paper, we exploit OpenStreetMap (OSM) [18], a free online community-driven map to enhance our ego-lane detection algorithm. OSM data is structured using three basic geometric elements: nodes, ways, and relations [19]. Ways are geometric objects like roads, railways, rivers, *etc*. It includes a collection of nodes, where the number of nodes is determined by the complexity of the object. Taking the road as an example, a straight road may consist of only two or three points as shown in Figure 2(a), and a curved road may consist of dozens of points as shown in Figure 2(b), ensuring the consistency of the OSM road shape and the real lane. Therefore, we use OSM road shape as lane model, which is irrelevant to lane features and robust to a variety of missing feature scenarios.

However, the OSM data is provided by user contributions, so that it is coarse and rife with errors. At the same time, the localization system employed on the vehicle might be noisy. These two problems lead to position errors between the OSM data and the real lane. To eliminate these errors, we propose a search-based optimization method, which finds the optimal position offset parameters by minimizing the distance between the OSM data and the extracted features, thereby improving the detection accuracy of the algorithm.

In summary, this paper presents a novel map-enhanced ego-lane detection (MELD) approach to address the missing feature problem (shown in Figure 3). First, we project the 3D LiDAR point cloud onto a range image and perform ROI selection based on the horizontal slope feature and the vertical slope feature. Then, we use the ROI selection result to generate a mask on the bird's eye view of the image and use a gradient operator to detect lane features. Finally, we propose a search-based optimization method to employ the OSM road shape as lane model, further obtain a robust ego-lane detection result. The main contributions of this paper are as follows:

1) Exploit the OSM road shape as lane model, which is highly consistent with the real lane shape and irrelevant to lane features, thereby robust to the missing feature scenarios.
2) Propose a search-based optimization method to eliminate the position errors between the OSM data and the real lane, thereby improving the detection accuracy.
3) Propose an efficient ego-lane detection framework being able to run in real-time at a frequency of 20 Hz on a single CPU.

The remainder of this paper is organized as follows. Section II presents the related work of ego-lane detection. In Section III, the proposed MELD approach is presented in detail. Experimental results are presented in Section IV. Finally, we conclude the paper in Section V.



(a) Straight road (b) Curved road

FIGURE 2. The results of projecting OSM data onto the bird's eye view of the image, where node is displayed in red and road is displayed in blue. (a) is a straight road sample. (b) is a curved road sample.

II. RELATED WORK

The ego-lane detection methods can be categorized according to different criteria, of which the most common categories are: model-based and learning-based [20]. Model-based methods tend to build a shape model [6]–[14] to describe the lane. Learning-based methods employ either traditional classifiers [21]–[23] or deep neural networks [24]–[26] to estimate the category of each pixel. In recent years, some

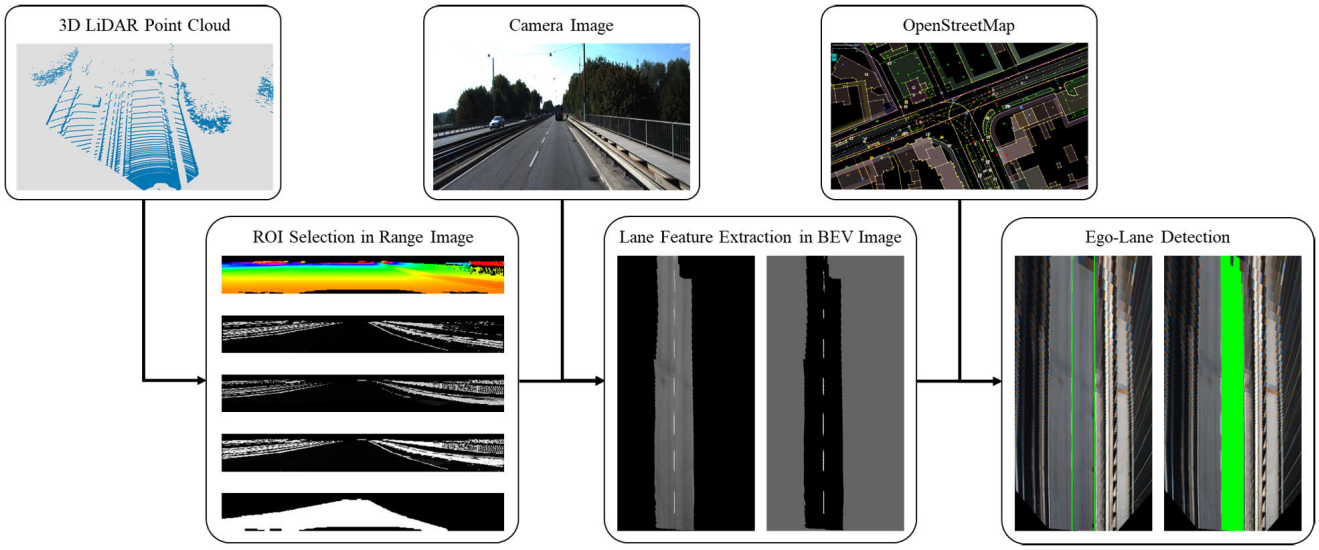


FIGURE 3. The framework of the map-enhanced ego-lane detection (MELD) approach. The input data includes 3D LiDAR point cloud, camera image, and OpenStreetMap. The main processing steps include ROI selection in range image, lane feature extraction in bird's eye view of image, and ego-lane detection.

methods based on LiDAR [27], [28] or fusion of LiDAR and camera [29], [30] have also been used for ego-lane detection. At the same time, since different sensors have different drawbacks, an online sensor reliability assessment and reliability-aware fusion method was proposed for ego-lane detection [31]. This paper focuses on solving the feature missing problem by using the road shape prior provided by OSM as the lane model. Therefore, the related work will mainly be carried out in two aspects: lane modeling and map using.

A. LANE MODELING

In recent years, lane modeling has played an important role in ego-lane detection, which refers to obtaining a compact high-level representation of road lane markings [32]. Different researchers have proposed different lane models. Some people only use simple straight lines, while others prefer to use more complex models, such as polynomial, clothoid, and spline.

The straight line model [6], [7] is the most commonly used geometric model. It is a good approximation for short distances and is the most common model in highway scenes. To increase the robustness of model fitting, several constraints have been applied additionally, such as parallelism [33] and road or lane width [34]. The straight line model is simple, but its applicability is limited, especially at long distances or curve road.

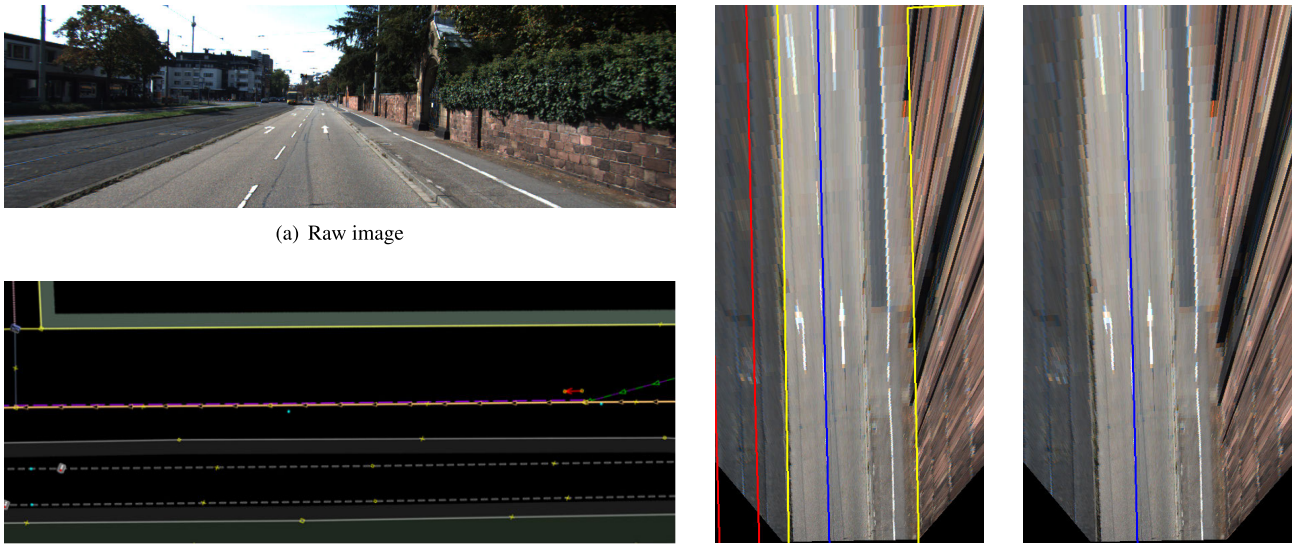
In [8], [9], curved roads are modeled in the bird's eye view using circular arc. Generally, the curvature of the road is small and continuous, so the circular arc is a conventional lane model on a ground plane [35]. However, the circular arc cannot handle more general curved roads.

Since performing well on more general curved roads, polynomials are also widely used for model fitting, including quadratic polynomial [10] and cubic polynomial [11]. But the fitting effect at the connection between a straight lane and a circular curve is limited [5].

Several researchers [36], [37] assume that the shape of the road as clothoid, which is defined by the initial curvature, the constant curvature change rate, and its total length. Clothoid can be approximated by a third-order polynomial and used to avoid abrupt changes in steering angle when driving from straight to circular roads.

Splines are smooth piecewise polynomial curves, which have been popular in previous studies [38]. Spline based lane model describes a wider range of lane structures, as it can form arbitrary shapes by a different set of control points [39]. Various spline representations have been proposed for lane modeling. In [12], a cubic spline with two to four control points is used for lane modeling. Wang *et al.* [13] presents lane modeling based on Catmull–Rom spline (also known as Overhauster spline), which is a local interpolating spline developed for computer graphics purposes. B-spline was introduced in [14], which can provide a local approximation of the contour with a small number of control points. Furthermore, nonuniform B-spline was used to construct the left and right lanes of the road [40]. Third-degree Bezier spline is also used to fit the left and right boundaries of the road surface [41]. The lane model was also improved to generate a B-snake model [42] or parallel-snake model [43]. Moreover, a spline-based particle filter is used to model the curvature of the lane [44].

Several combination models have also been proposed as lane models. In [45], the image is divided into multiple slices, and lanes in each slice are fitted with straight lines to form a piecewise linear model. Jung *et al.* [46] proposed a linear parabolic lane model consisting of a linear function in the near-range and a parabola in the far-range. The nearby



(a) Raw image

(b) Raw OSM data

(c) Transformed OSM data

(d) OSM road data

FIGURE 4. OSM data of a sample in the KITTI Lane dataset. (a) shows the raw image of a sample scenario in the KITTI Lane dataset. (b) shows the raw OSM data corresponding to the scenario, in which the red arrow indicates the position and direction of the ego vehicle. (c) shows the transformed OSM data where roads are shown in blue, railways are shown in red, and other geometric objects are shown in yellow. (d) shows the road that the ego vehicle is currently traveling.

straight line model provides the robustness of the model, and the parabola provides the flexibility of the model. Similar to [46], some researchers employ a clothoid model [47] or a hyperbola model [48] as the far-range model.

B. MAP USING

A map that contains rich geometric and semantic information about the environment is essential for autonomous driving systems. Impressive results have been achieved by introducing maps to perception [15], prediction [16], and motion planning [17]. Various map-based methods are also proposed for ego-lane detection.

In [49], the curvature of the road was first obtained from the GPS position and the digital map, and then it was used to determine whether it was driving on a straight road or a curved road. Different road regions use different lane detection modules, of which straight roads are fitted using linear models and curved roads are fitted using circular arc.

To enhance the performance and robustness of the lane detection system, Möhler *et al.* [50] proposed to extract lane width and curvature of upcoming road segments from a digital map to adapt certain configuration parameters. In addition, clothoid is used for model fitting.

It is worth mentioning that Godoy *et al.* [51] proposes an automatic program that extends the digital map definition to generate a better approximation of the real road shape. Specifically, the proposed algorithm replaces the straight line segments between nodes with cubic Bezier curves and automatically adjusts the control points for fitting road.

As described in Section I, all mathematical representation models have the overfitting issue when features are missing. The methods that using maps still use mathematical representation model as lane model, and the overfitting issue still exists. In this paper, we use the road shape in OSM data as lane model and transform the fitting problem into a search-based optimization problem. The advantage is that the prior knowledge provided by the map is effectively used, and the problem of missing feature is addressed.

III. EGO-LANE DETECTION

In this section, the proposed MELD approach will be described in detail. First, we describe the OSM data format and how to obtain the data needed for this paper. Next, we show the preprocessing step, which contains Region of Interest (ROI) selection and lane feature extraction. Finally, we explain how OSM data is used for ego-lane detection.

A. OpenStreetMap

In 2004, the OpenStreetMap project was started with the goal of creating a free to use and editable map of the world [18]. So far, the OSM project has been greatly developed, and more and more researchers prefer to employ the OSM to enhance their algorithms. The OSM data can be accessed via the corresponding website[1] by specifying a bounding box. Figure 4(b) shows the raw OSM data of the scenario corresponding to of Figure 4(a) (we add a red arrow to indicate the position and direction of the ego vehicle).

The OSM data is in the world coordinate system, but our ego-lane detection algorithm is performed in the road coordinate system. Therefore, the OSM data needs to be transformed to the road coordinate system first. Figure 4(c) shows the results of our coordinate transformation result.

[1] https://www.openstreetmap.org/

It should be noted that the data beyond the image view is clipped.

OSM data provides rich geometric information. However, for our purposes, the most useful information is the road that the ego vehicle is traveling on, so other geometric information is also clipped. Finally, the OSM data containing only the currently traveling road after being transformed and clipped is shown in Figure 4(d), which will be used as the lane model later.

B. PREPROCESSING

Before using the OSM road shape for ego-lane detection, lane line features and road curb features need to be extracted first. To improve the speed and accuracy of the algorithm, feature extraction is generally after ROI selection [5]. Therefore, we consider both ROI selection and lane feature extraction as preprocessing in this section.

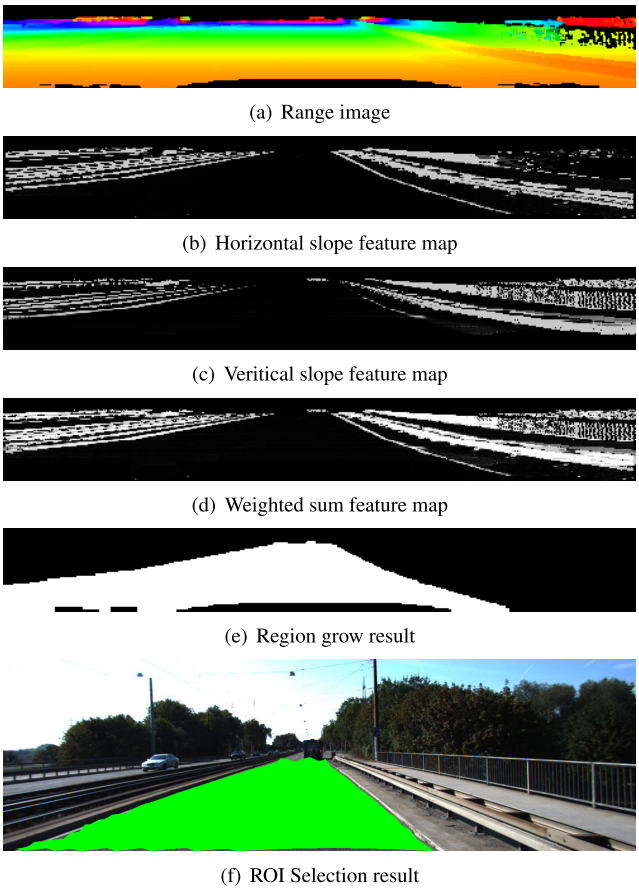
1) ROI SELECTION

Among all tasks in ego-lane detection, ROI selection is usually the first step performed in most of the previous studies [38]. The main reason for focusing on ROI selection is to increase the computation efficiency and detection accuracy. In this paper, we consider the drivable area to be the ROI. It contains all lane markers and road curbs for feature extraction, and trees, buildings and other objects outside the road can be ignored. Therefore, ROI selection can be redefined as road detection.

Camera is a light-sensitive sensor that is easily affected by illumination and shadows. Although many deep learning methods have greatly improved the performance of image processing in recent years, what has to be considered is computational efficiency, so it is not suitable for the preprocessing step. Unlike the camera, 3D LiDAR is unaffected by illumination and can provide accurate geometric information about the environment. Therefore, we use 3D LiDAR for ROI selection.

To meet the real-time requirements, we project the 3D point cloud data to a 2D range image, which can achieve data compression while retaining neighborhood information. The number of rows of the range image is defined by the number of laser beams of the 3D LiDAR. The KITTI dataset uses Velodyne HDL-64E, so the number of rows is 64. The number of columns of the range image is the horizontal resolution of the 3D LiDAR. We only use 90° field of view that coincides with the camera, so the number of columns is 500. In summary, the size of the range image is 64×500, and an example of a range image can be seen in Figure 5(a).

Based on the assumption that the road is flat and continuous, we do road detection on the range image using the region grow method. As the vehicle is traveling in the forward direction, the road is always located in front of the vehicle. Therefore, seed points are selected as points in front of the vehicle, which are located in the bottom center of the range image. The similarity between pixels is defined by the horizontal slope feature and the vertical slope feature.



(a) Range image

(b) Horizontal slope feature map

(c) Veritical slope feature map

(d) Weighted sum feature map

(e) Region grow result

(f) ROI Selection result

FIGURE 5. An illustration of the implementation process of the ROI selection. (a) shows a range image whose pixel value represents the distance from the point to LiDAR. (b) is an example of a horizontal feature map. (c) is an example of a vertical feature map. (d) is the corresponding weighted sum feature map. (e) shows the region grow result. (f) shows the ROI selection result.

For each pixel, the horizontal slope feature is calculated based on k neighborhood points in the same laser beam:

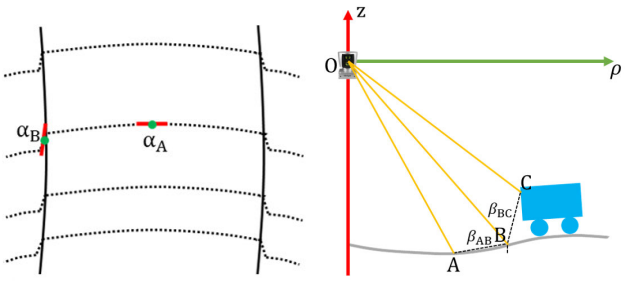
$$\alpha = \frac{\sum_{i=1}^{k}(x_i - \bar{X})(y_i - \bar{Y})}{\sum_{i=1}^{k}(x_i - \bar{X})^2} \tag{1}$$

where (x_i, y_i) is the position in the 3D LiDAR coordinate system of the pixel, and $\bar{X}$, $\bar{Y}$ are the average value of the k neighbors. As shown in Figure 6(a), the feature value α_A on the ground is close to 0, while the feature value α_B on the road curb is close to infinity, so the horizontal slope feature is used to detect the road curb. At the same time, the features were normalized using the logistic function, and the results are shown in Figure 5(b).

For each pixel, the vertical slope feature is calculated based on the points on two adjacent laser beams in the same ray direction:

$$\beta = \frac{z^{r+1} - z^r}{d^{r+1} - d^r} \tag{2}$$

where (d^r, z^r) is a point on the r laser beam, and (d^{r+1}, z^{r+1}) is a point on the $r+1$ laser beam, $d = \sqrt{x^{r2} + y^{r2}}$. As shown



(a) Horizontal slope feature (b) Veritical slope feature

FIGURE 6. An illustration of the two slope features. (a) is the horizontal slope feature, where α_A represents the ground and α_B represents the road curb. (b) is the vertical slope feature, where β_{AB} represents the ground and β_{BC} represents the obstacle.

in Figure 6(b), the feature value β_{AB} on the ground is close to 0, while the feature value β_{BC} on the obstacle is close to infinity, so the vertical slope feature is used to detect obstacles. At the same time, the features were normalized using the logistic function, and the results are shown in Figure 5(c).

After getting the two slope features, the weighted sum is finally calculated:

$$\gamma = a \cdot \alpha + b \cdot \beta \tag{3}$$

where a and b are coefficients of horizontal slope feature and vertical slope feature respectively. Figure 5(d) shows the weighted sum feature map and it can be seen that obstacles and road curbs are all detected. After obtaining the weighted sum feature map, we use horizontal and vertical region grow to obtain the road area, and the results are shown in Figure 5(e). Finally, we project road points onto the perspective image and use Delaunay Triangulation [52] to upsampling the sparse point cloud to obtain the ROI selection result. The ROI selection result is shown in Figure 5(f).

2) LANE FEATURE EXTRACTION

As described in Section I, lane features in the KITTI Lane dataset are mainly composed of two parts: lane line features and road curb features. In ROI selection, the horizontal slope feature has a good effect on detecting road curbs, so we directly use ROI selection results as road curb features.

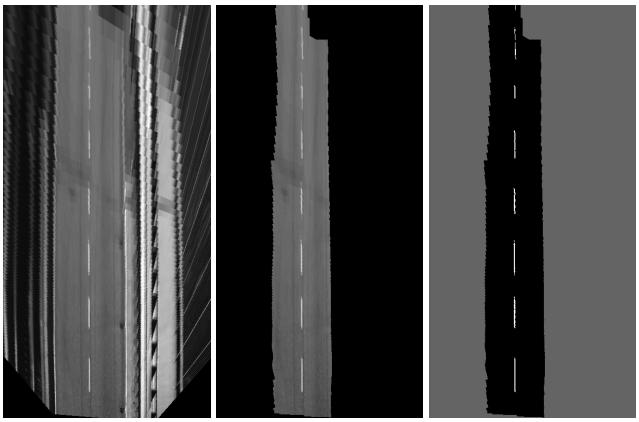
Lane line feature extraction aims to extract low-level features from images to support ego-lane detection, such as color, texture, edges [53]. Among them, edges are the most common feature used in ego-lane detection for structured roads [35]. An edge is mathematically defined by the gradient of the intensity function [54], so we define the gradient as:

$$G = \begin{bmatrix} -1 & -1 & -1 & 2 & 2 & 2 & -1 & -1 & -1 \end{bmatrix} * I \tag{4}$$

where I is the image and G is the calculated gradient.

Since the lane line width becomes smaller as the distance increases in the perspective image, we perform feature extraction on the bird's eye view image, so that the lane line width is constant and easy to detect. As the vehicle is traveling along the lane, the lane lines are distributed longitudinally, so we mainly extract the lateral gradient changes. At the same time, we found that the lane line width generally takes $2 \sim 4$ pixels on the bird's-eye view image. For these two reasons, and in order to increase the computational efficiency, we use a convolution with the size of 9×1. There is a sharp contrast between the road surface and painted lane lines, so the 3 elements in the middle of the convolution kernel are 2 and the others are -1. In this way, when there is no lane, the intensity values between pixels are similar, and the gradient is 0; when there is a lane line, the intensity of the three elements in the middle is high, the intensity of the two sides is low, and the gradient is relatively large.

Therefore, when the gradient $G(i, j)$ is greater than G_{th}, the pixel at the (i, j) position is marked as the lane. An example of the lane line feature extraction result can be seen in Figure 7(c). It should be noted that lane line feature extraction is performed on a gray-scale image (shown in Figure 7(a)), and pixels outside the ROI region are not considered (shown in Figure 7(b)).



(a) Raw image (b) ROI image (c) Feature image

FIGURE 7. An illustration of the implementation process of the lane line feature extraction. (a) shows the raw gray scale image on the bird's eye view. (b) projects the ROI selection result onto the image. (c) shows the results of lane line feature extraction.

C. EGO-LANE DETECTION

The main goal of this stage is to extract a compact high-level representation of the lane that can be used for decision making [53]. In most papers, mathematical representation models are used as compact high-level representations such as straight lines, polynomials, parabolas, and splines. In order to fit lane features to these mathematical representation models, Least Squares Method (LSM) and Random Sample Consensus (RANSAC) are widely used. Since mathematical representation models have the overfitting issue when features are missing, we exploit OSM data to enhance ego-lane detection.

As mentioned in the previous section, OSM data is provided by the volunteers, so it is very coarse and rife with errors, which is called OSM data error. At the same time,

when projecting the OSM data onto the image, the approximate vehicle pose estimation causes errors in the relative position of the OSM data with respect to the vehicle, which is called vehicle positioning error.

Since we perform ego-lane detection on the 2D image plane, the errors can be eliminated by the rotation parameter θ and the translation parameter x, y (the x-axis points to the vehicle's forward direction, while the y-axis is orthogonal to the x-axis and points the left of the vehicle). In real urban scenes, the radius of curvature of the road is relatively large, so the translation parameter x can be ignored. Therefore, we only need to consider the parameters y and θ, which represent the lateral offset and the heading offset, respectively. It should be noted that we do lane line detection and road curb detection simultaneously, so the lateral offset y consists of two parts: lane line lateral offset y_l and road curb lateral offset y_r. In summary, the parameters we need to estimate include heading offset θ, lane line lateral offset y_l, and road curb lateral offset y_r.

To estimate these three parameters, we minimize the distance from the detected lane features to the OSM data. Since the OSM road shape consists of a series of points and their connections, the distance from the feature point to the OSM data is equal to the distance from the feature point to its nearest connection:

$$d_i = \frac{|(y_j^Q - y_k^Q)x_i^P - (x_j^Q - x_k^Q)y_i^P + x_j^Q y_k^Q - y_j^Q x_k^Q|}{\sqrt{(y_j^Q - y_k^Q)^2 + (x_j^Q - x_k^Q)^2}} \quad (5)$$

where (x_i^P, y_i^P) is the i-th feature point. (x_j^Q, y_j^Q) and (x_k^Q, y_k^Q) are the two adjacent OSM points closest to the feature point. Therefore, the optimization function is:

$$\begin{aligned} &\min_{y_l, y_r, \theta} \sum_{i=1}^{m} d_i \\ &\text{s.t. } |y_l| \le y_{max}, \\ &\quad |y_r| \le y_{max}, \\ &\quad |\theta| \le \theta_{max}. \end{aligned} \quad (6)$$

where m is the number of feature points. y_{max} is the maximum value of lateral offset, and θ_{max} is the maximum value of heading offset.

The above optimization problem turns out to be very difficult to solve due to looking for the OSM line closest to the feature point. Therefore, we rely on a search-based algorithm to find the optimal approximate solutions. The basic idea is that we iterate through all possible values of these three parameters θ, y_l, and y_r. After iterating all parameters and obtaining all corresponding distances, we look for the optimal parameters that achieve the smallest distance. However, the time complexity of looping through these three parameters is $O(N^3)$, which is very time consuming and cannot meet the real-time requirements. Therefore, we optimize these three parameters separately, so that the time complexity is reduced to $O(3N)$.

Algorithm 1 Search-Based Parameters Optimization

Input: feature points $P \in R^{m\times 2}$, OSM points $Q \in R^{n\times 2}$
Output: optimization parameter λ^*

```
1:  d_min ← +∞
2:  λ* ← 0
3:  for λ = −λ_max to λ_max step δ_λ do
4:      d ← 0
5:      for p in P do
6:          for q in Q do                          ▷ descending order
7:              transform q to q'
8:              if (q'_x ≤ p_x) then
9:                  d += d_p                       ▷ d_p is calculated by (5)
10:                 break
11:             end if
12:         end for
13:     end for
14:     if (d ≤ d_min) then
15:         d_min ← d
16:         λ* ← λ
17:     end if
18: end for
```

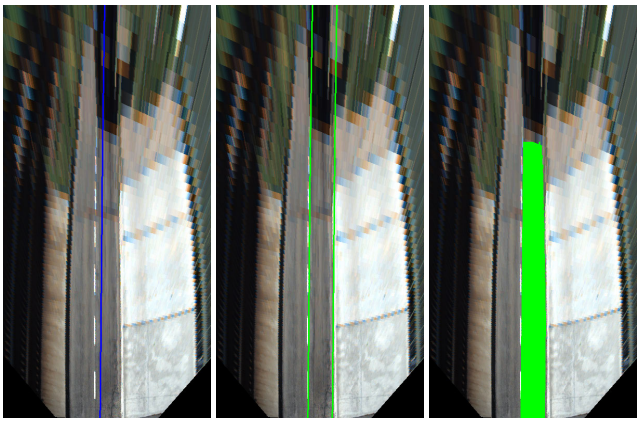
The proposed search-based optimization algorithm is presented in Algorithm 1. The inputs for the algorithm are the m features points P and n OSM points Q. The outputs from this algorithm are the optimization parameter λ^*, which separately represents θ^*, y_l^*, and y_r^* in each step of optimization. In line 3, all possible values are traversed by given the maximum value of these three parameters. From line 4 to line 13, the distance from the feature point to the OSM data is calculated. The optimal parameters that achieves the smallest distance is selected in line 14 to line 17.

After obtaining the optimization results of the left and right boundaries, we use the vertical slope feature (mentioned in the ROI selection subsection) to detect all obstacles between two boundaries, and take the point closest to the origin as the upper boundary. In this way, the result of ego-lane detection is the area surrounded by these three boundaries.

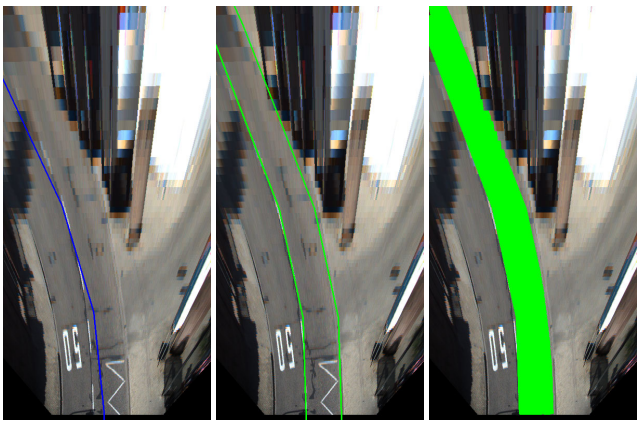
Figure 8 shows the ego-lane detection results of the scenarios corresponding to Figure 2. In (a), the significant lateral error is eliminated, and the OSM road shape perfectly coincides with the lane boundaries. It can be seen from (b) that the significant heading error is eliminated, except for some slight errors between the OSM road shape and the lane boundaries shape.

IV. EXPERIMENTAL EVALUATION

To evaluate the accuracy and real-time performance of MELD, we test it on the public KITTI Lane benchmark. All algorithms are implemented in C++, PCL (Point Cloud Library) and OpenCV (Open Source Computer Vision Library), running on a laptop computer with an Intel i5−8265U 1.66 GHz CPU with 8 GB main memory.



(a) Straight road



(b) Curved road

FIGURE 8. The result of ego-lane detection, where the left column is the raw OSM data, the middle column is the parameters optimization result, and the right column is the ego-lane detection result.

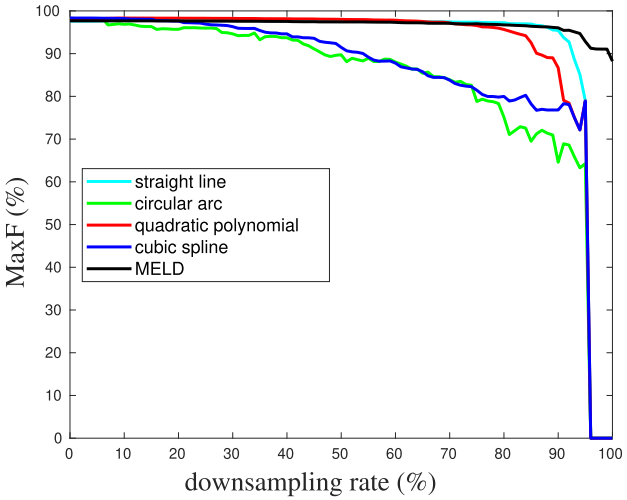


FIGURE 9. Robustness of different methods to missing feature.

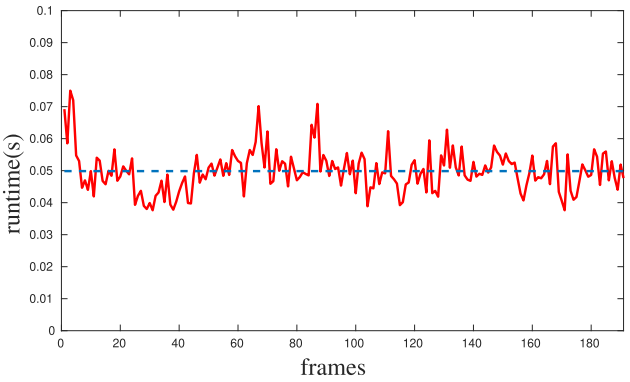


FIGURE 10. Runtime of MELD for both training and testing datasets.

A. EXPERIMENTAL SETUP

1) KITTI LANE BENCHMARK

The KITTI Lane benchmark [55] is a widely used benchmark for ego-lane detection. 95 training samples and 96 testing samples are collected in various urban scenes with marked lanes were included. The evaluation metrics include maximum F1-measure (MaxF), average precision (AP), precision (PRE), recall (REC), false positive rate (FPR), and false negative rate (FNR), where MaxF is used as the primary metric value for comparison between different methods.

2) EXPERIMENTS SETTING

For ROI selection, the neighborhood points size k for computing horizontal slope feature is 7, the weighting coefficient of the horizontal slope feature a is 0.5, and the weighting coefficient of the vertical slope feature b is 0.5.

For lane feature extraction, the gradient threshold G_{th} is 200.

For ego-lane detection, the maximum lateral error y_{max} is 100 pixels and the step size δ_y is 5 pixels; the maximum heading error θ_{max} is 0.1 radians and the step size δ_θ is 0.005 radians.

B. PERFORMANCE EVALUATION

We tested MELD on the KITTI Lane benchmark and compared it with other state-of-the-art methods, including SCRFFPFHGSP [21], SPlane + BL [22], SPRAY [23], UpConv-Poly [24], RBNet [25], MANLDF, RoadNet3 [26], and NVLaneNet. All results are evaluated on the KITTI evaluation server,[2] and the performance of the algorithms is shown in Table 1.

The results show that MELD achieved 93.56% in the MaxF score, which is 1.70% higher than the previous state-of-the-art method. The improvement of the MaxF score is mainly due to the fact that the PRE of MELD can reach 95.94%, and this is precisely because we use OSM road shape as the lane model, which can accurately detect lane boundaries and further achieve higher accuracy.

C. ROBUSTNESS TO MISSING FEATURE

To evaluate the robustness of MELD to the missing feature problem, we down-sample the lane features with the sampling rate from 0 % to 100 % and perform model comparison experiments on the training dataset. Contrast mathematical representation models include straight line, cir-

[2]http://www.cvlibs.net/datasets/kitti/eval_road.php

TABLE 1. Results of online evaluation on KITTI Lane benchmark. Best scores are highlighted in bold.

Method	MaxF	AP	PRE	REC	FPR	FNR	Runtime
SCRFFPFHGSP [21]	57.22 %	39.34 %	41.78 %	90.79 %	22.28 %	9.21 %	5 s / CPU
SPlane + BL [22]	69.63 %	73.78 %	80.01 %	61.63 %	2.71 %	38.37 %	2 s / CPU
SPRAY [23]	83.42 %	86.84 %	84.76 %	82.12 %	2.60 %	17.88 %	0.045 s / GPU
Up-Conv-Poly [24]	89.88 %	87.52 %	92.01 %	87.84 %	1.34 %	12.16 %	0.08 s / GPU
RBNet [25]	90.54 %	82.03 %	94.92 %	86.56 %	0.82 %	13.44 %	0.18 s / GPU
MANLDF	91.37 %	91.40 %	93.08 %	89.71 %	1.17 %	10.29 %	0.05 s / GPU
RoadNet3 [26]	91.47 %	91.01 %	91.78 %	91.17 %	1.44 %	8.83 %	0.3 s / GPU
NVLaneNet	91.86 %	**91.42 %**	90.89 %	**92.85** %	1.64 %	**7.15 %**	0.08 s / GPU
MELD	**93.56 %**	88.58 %	**95.94 %**	91.30 %	**0.68 %**	8.70 %	0.05 s / CPU

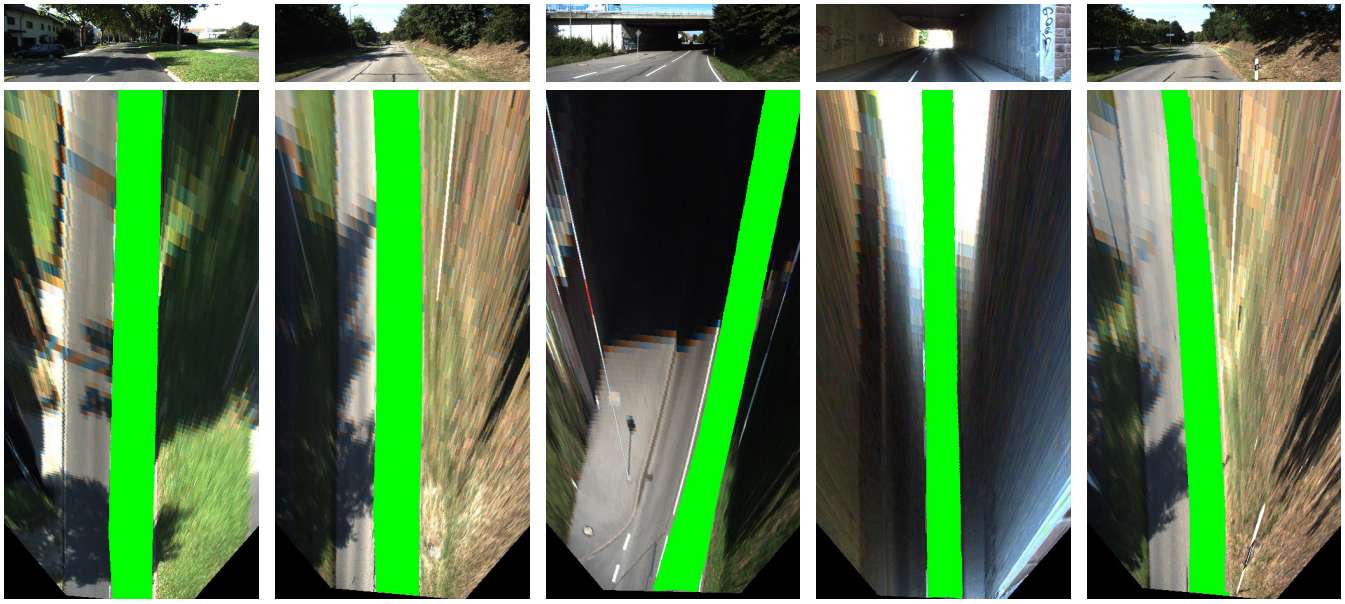


FIGURE 11. Qualitative results of MELD. Each column represents a typical missing feature scenario, where the top row is the raw image and the bottom row is the detection result (shown as green) on the bird's eye view of image.

cular arc, quadratic polynomial, and cubic spline. It should be noted that we use the left and right boundary points from the annotation results in the KTTTI Lane training dataset as lane features, which can avoid the interference of different detection algorithms and noise, therefore ensuring the fairness of the experiment. The evaluation metric uses MaxF, and the experimental results are shown in Figure 9.

It can be seen that the fitting results of all mathematical representation models become worse as the number of features decreases. However, since the OSM road shape is used as the lane model, MELD is very robust to missing features. Even if the number of features decreases, the effect remains unchanged. At the same time, in some extreme scenarios, such as no visible features, we directly use OSM road shape as the lane boundary, and the MaxF can reach 88.23 %, while other mathematical representation models cannot handle this scenario.

D. RUNTIME

Since MELD is to be used on autonomous driving systems, the less runtime of the algorithm allows systems to get information about the surrounding environment earlier, thereby ensuring the safety of the systems. As shown in Figure 10, the runtime of MELD on both training and testing datasets averages around 50 ms. This is twice as fast as the rotation rate of the 3D LIDAR, so MELD can be used safely on autonomous driving systems.

E. QUALITATIVE RESULTS

Some detection results of MELD are shown in Figure 11. For the first two columns, the lane line is heavily worn and blocked by shadows or other objects. For the third and fourth columns, the vehicle is going to pass through the tunnel, and the lane line in the far-range is covered by black or white. For the last column, there are no visible lane features can be seen in the picture. All these scenarios have the missing feature problem, but MELD can stably detect the ego-lane.

V. CONCLUSION

In this study, we employ the OSM road shape as lane model to enhance our ego-lane detection algorithm, which is robust to the challenging scenarios of missing feature. At the same time, to eliminate the position error between the OSM data

and the real lane, a search-based optimization algorithm is proposed to improve the accuracy of the algorithm. We validate the proposed algorithm on the well-known KITTI Lane benchmark, which achieved state-of-the-art performance in terms of accuracy and real-time performance. However, the proposed method has only been validated on the two-way road in which the vehicle is driving on the right lane like KITTI Lane dataset. In future work, we will expand our work to more general roads like multi-lane, intersection, and even no centerline. At the same time, to obtain more accurate ego-lane detection results, the OSM road shape error will also be eliminated.

REFERENCES

[1] H. Yoo, U. Yang, and K. Sohn, "Gradient-enhancing conversion for illumination-robust lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1083–1094, Sep. 2013.

[2] J. Niu, J. Lu, M. Xu, P. Lv, and X. Zhao, "Robust lane detection using two-stage feature extraction with curve fitting," *Pattern Recognit.*, vol. 59, pp. 225–233, Nov. 2016.

[3] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 286–291.

[4] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial cnn for traffic scene understanding," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018.

[5] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, "A review of recent advances in lane detection and departure warning system," *Pattern Recognit.*, vol. 73, pp. 216–234, Jan. 2018.

[6] A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with ransac and Kalman filter," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 3261–3264.

[7] H. Kong, J.-Y. Audibert, and J. Ponce, "Vanishing point detection for road detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 96–103.

[8] M. Nieto, L. Salgado, F. Jaureguizar, and J. Arróspide, "Robust multiple lane road modeling based on perspective analysis," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 2396–2399.

[9] F. Samadzadegan, A. Sarafraz, and M. Tabibi, "Automatic lane detection in image sequences for vision-based navigation purposes," in *Proc. ISPRS Image Eng. Vis. Metrol.*, 2006, pp. 251–257.

[10] W. Van Gansbeke, B. De Brabandere, D. Neven, M. Proesmans, and L. Van Gool, "End-to-end lane detection through differentiable least-squares fitting," 2019, *arXiv:1902.00293*. [Online]. Available: http://arxiv.org/abs/1902.00293

[11] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 286–291.

[12] Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 16–26, Mar. 2008.

[13] Y. Wang, D. Shen, and E. K. Teoh, "Lane detection using catmull-rom spline," in *Proc. IEEE Int. Conf. Intell. Vehicles*, vol. 1, Oct. 1998, pp. 51–57.

[14] J. Deng, J. Kim, H. Sin, and Y. Han, "Fast lane detection based on the b-spline fitting," *Int. J. Res. Eng. Technol*, vol. 2, no. 4, pp. 134–137, 2013.

[15] B. Yang, M. Liang, and R. Urtasun, "HDNET: Exploiting HD maps for 3D object detection," in *Proc. Conf. Robot Learn.*, 2018, pp. 146–155.

[16] S. Casas, W. Luo, and R. Urtasun, "IntentNet: Learning to predict intention from raw sensor data," in *Proc. Conf. Robot Learn.*, 2018, pp. 947–956.

[17] Y. F. Chen, S.-Y. Liu, M. Liu, J. Miller, and J. P. How, "Motion planning with diffusion maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 1423–1430.

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

[19] M. Hentschel and B. Wagner, "Autonomous robot navigation based on OpenStreetMap geodata," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 1645–1650.

[20] D. Vajak, M. Vranješ, R. Grbić, and D. Vranješ, "Recent advances in vision-based lane detection solutions for automotive applications," in *Proc. Int. Symp. ELMAR*, Sep. 2019, pp. 45–50.

[21] I. V. Gheorghe, "Semantic segmentation of terrain and road terrain for advanced driver assistance systems," Ph.D. dissertation, Dept. Eng. Comput., Coventry Univ., Coventry, U.K., 2015.

[22] N. Einecke and J. Eggert, "Block-matching stereo with relaxed fronto-parallel assumption," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 700–705.

[23] T. Kühnl, F. Kummert, and J. Fritsch, "Spatial ray features for real-time ego-lane extraction," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 288–293.

[24] G. Oliveira, W. Burgard, and T. Brox, "Eifficient deep methods for monocular road segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2016, pp. 4885–4891.

[25] Z. Chen and Z. Chen, "RBNet: A deep neural network for unified road and road boundary detection," in *Proc. Int. Conf. Neural Inf. Process.* Zürich, Switzerland: Springer, 2017, pp. 677–687.

[26] Y. Lyu, L. Bai, and X. Huang, "Road segmentation using CNN and distributed LSTM," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.

[27] M. Thuy and F. León, "Lane detection and tracking based on lidar data," *Metrol. Meas. Syst.*, vol. 17, no. 3, pp. 311–321, Jan. 2010.

[28] L. D. P. Veronese, A. Ismail, V. Narayan, and M. Schulze, "An accurate and computational efficient system for detecting and classifying ego and sides lanes using LiDAR," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1476–1483.

[29] A. S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller, "Finding multiple lanes in urban road networks with vision and lidar," *Auto. Robots*, vol. 26, nos. 2–3, pp. 103–122, Apr. 2009.

[30] S. Shin, I. Shim, and I. So Kweon, "Combinatorial approach for lane detection using image and LIDAR reflectance," in *Proc. 12th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, Oct. 2015, pp. 485–487.

[31] T. T. Nguyen, J. Spehr, J. Xiong, M. Baum, S. Zug, and R. Kruse, "Online reliability assessment and reliability-aware fusion for ego-lane detection using influence diagram and bayes filter," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Nov. 2017, pp. 7–14.

[32] Đ. Obradović, Z. Konjović, E. Pap, and I. J. Rudas, "Linear fuzzy space based road lane model and detection," *Knowl.-Based Syst.*, vol. 38, pp. 37–47, Jan. 2013.

[33] E. Adachi, H. Inayoshi, and T. Kurita, "Estimation of lane state from car-mounted camera using multiple-model particle filter based on voting result for one-dimensional parameter space," in *Proc. MVA*, 2007, pp. 323–326.

[34] R. Wang, Y. Xu, B. Li, and Y. Zhao, "A vision-based road edge detection algorithm," in *Proc. Intell. Vehicle Symp.*, 2002, pp. 141–147.

[35] S. Yenikaya, G. Yenikaya, and E. Düven, "Keeping the vehicle on the road: A survey on on-road lane detection systems," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 1–43, Oct. 2013.

[36] H. Loose, U. Franke, and C. Stiller, "Kalman particle filter for lane recognition on rural roads," in *Proc. IEEE Intell. Vehicles Symp.*, 2009, pp. 60–65.

[37] C. Gackstatter, P. Heinemann, S. Thomas, and G. Klinker, "Stable road lane model based on clothoids," in *Advanced Microsystems for Automotive Applications 2010*. Berlin, Germany: Springer, 2010, pp. 133–143.

[38] Y. Xing, C. Lv, L. Chen, H. Wang, H. Wang, D. Cao, E. Velenis, and F.-Y. Wang, "Advances in vision-based lane detection: Algorithms, integration, assessment, and perspectives on ACP-based parallel vision," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 3, pp. 645–661, May 2018.

[39] K. Zhao, M. Meuter, C. Nunn, D. Muller, S. Muller-Schneiders, and J. Pauli, "A novel multi-lane detection and tracking system," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 1084–1089.

[40] Q.-B. Truong and B.-R. Lee, "New lane detection algorithm for autonomous vehicles using computer vision," in *Proc. Int. Conf. Control, Autom. Syst.*, Oct. 2008, pp. 1208–1213.

[41] Q. Wen, Z. Yang, Y. Song, and P. Jia, "Road boundary detection in complex urban environment based on low-resolution vision," in *Proc. 11th Joint Int. Conf. Inf. Sci.*, 2008, pp. 208–214.

[42] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-Snake," *Image Vis. Comput.*, vol. 22, no. 4, pp. 269–280, Apr. 2004.

[43] X. Li, X. Fang, C. Wang, and W. Zhang, "Lane detection and tracking using a parallel-snake approach," *J. Intell. Robotic Syst.*, vol. 77, nos. 3–4, pp. 597–609, Mar. 2015.

[44] R. F. Berriel, E. de Aguiar, A. F. de Souza, and T. Oliveira-Santos, "Ego-lane analysis system (ELAS): Dataset and algorithms," *Image Vis. Comput.*, vol. 68, pp. 64–75, Dec. 2017.
[45] H.-Y. Cheng, B.-S. Jeng, P.-T. Tseng, and K.-C. Fan, "Lane detection with moving vehicles in the traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 571–582, Dec. 2006.
[46] C. R. Jung and C. R. Kelber, "Lane following and lane departure using a linear-parabolic model," *Image Vis. Comput.*, vol. 23, no. 13, pp. 1192–1202, Nov. 2005.
[47] R. Danescu, S. Nedevschi, and T.-B. To, "A stereovision-based lane detector for marked and non-marked urban roads," in *Proc. IEEE Int. Conf. Intell. Comput. Commun. Process.*, Sep. 2007, pp. 81–88.
[48] Y. Xing, C. Lv, H. Wang, D. Cao, and E. Velenis, "Dynamic integration and online evaluation of vision-based lane detection algorithms," *IET Intell. Transp. Syst.*, vol. 13, no. 1, pp. 55–62, Jan. 2019.
[49] Y. Jiang, F. Gao, and G. Xu, "Computer vision-based multiple-lane detection on straight road and in a curve," in *Proc. Int. Conf. Image Anal. Signal Process.*, 2010, pp. 114–117.
[50] N. Möhler, D. John, and M. Voigtländer, "Lane detection for a situation adaptive lane keeping support system, the safelane system," in *Advanced Microsystems for Automotive Applications 2006*. Berlin, Germany: Springer, 2006, pp. 485–500.
[51] J. Godoy, A. Artuñedo, and J. Villagra, "Self-generated OSM-based driving corridors," *IEEE Access*, vol. 7, pp. 20113–20125, 2019.
[52] D. T. Lee and B. J. Schachter, "Two algorithms for constructing a delaunay triangulation," *Int. J. Comput. Inf. Sci.*, vol. 9, no. 3, pp. 219–242, Jun. 1980.
[53] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: A survey," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 727–745, Apr. 2014.
[54] B. Ma, S. Lakshmanan, and A. Hero, "A robust Bayesian multisensor fusion algorithm for joint lane and pavement boundary detection," in *Proc. Int. Conf. Image Process.*, vol. 1, 2001, pp. 762–765.
[55] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

XIAOLIANG WANG received the B.S. degree of automation from the Dalian University of Technology, Dalian, China, in 2018. He is currently pursuing the M.S. degree with Shanghai Jiao Tong University, Shanghai, China.

His current research interests include ego-lane detection, object detection, object tracking, and multi-sensor fusion.

YEQIANG QIAN (Graduate Student Member, IEEE) received the B.S. degree in automation from Hunan University, Changsha, Hunan, China, in 2015. He is currently pursuing the Ph.D. degree in control science and engineering with Shanghai Jiao Tong University, Shanghai, China.

His main research interests include computer vision, pattern recognition, machine learning, and their applications in intelligent transportation systems.

CHUNXIANG WANG received the Ph.D. degree in mechanical engineering from the Harbin Institute of Technology, Harbin, China, in 1999.

She is currently an Associate Professor with the Department of Automation, Shanghai Jiao Tong University, Shanghai, China. Her research interests include robotic technology and electromechanical integration.

MING YANG (Member, IEEE) received the master's and Ph.D. degrees from Tsinghua University, Beijing, China, in 1999 and 2003, respectively.

He is currently the Full Tenure Professor with Shanghai Jiao Tong University and the Deputy Director of the Innovation Center of Intelligent Connected Vehicles. He has been working in the field of intelligent vehicles for more than 20 years. He has participated in several related research projects, such as the THMR-V project (first intelligent vehicle in China), European CyberCars and CyberMove projects, CyberC3 project, CyberCars-2 project, ITER transfer cask project, AGV, and so on.