Received May 23, 2017, accepted June 13, 2017, date of publication August 2, 2017, date of current version March 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2731804

Robust Lane-Mark Extraction for Autonomous Driving Under Complex Real Conditions

HANYU XUAN[1], HONGZHE LIU[1], JIAZHENG YUAN[2], AND QING LI[3]
[1]Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing 100101, China
[2]Information Institute of Beijing Union University, Beijing 100101, China
[3]Engineering Center of Beijing Union University, Beijing 100101, China

Corresponding author: Hongzhe Liu (liuhongzhe@buu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61571045 and Grant 61372148, in part by the Beijing Natural Science Foundation under Grant 4152016, and in part by the High-level Teacher Team Building Support Plan of Beijing municipal colleges and universities—High-level Innovative Team Building Plan under Grant IDHT20170511.

ABSTRACT Lane marks on roads are among the most important items of road scene information in the process of autonomous driving, and lane-mark extraction based on visual cognitive computing is one of the most important components of advanced driving assistance systems in intelligent transportation system. Onboard cameras mounted on the front of autonomous vehicles capture road scene images from which lane marks are extracted. This paper proposes a new lane-mark extraction algorithm with four major parts. First, this paper handles the road images captured from onboard cameras by grayscale and fast median filter. Then, we exploit the characteristics of lane marks in road images as constraints to propose lane-features filter based on multi-constraints used to extract lane marks. Then, a clustering algorithm based on the double point removal of a *p*-least squares algorithm is proposed to cluster features, and recursive dichotomy algorithm is used to fit the candidate lane marks. Finally, we carry out verification and optimization on candidate lane marks to obtain more accurate and stable extraction results. In our experiment, we divide the common complex road scenes into four categories. The results show that the proposed method can robustly extract lane marks under various complex real conditions. This paper also proposes forward a method to evaluate the results of lane-mark extraction, and partial test results are evaluated.

INDEX TERMS Autonomous driving, ADASITS, lane-mark extraction, lane-features filter, multi-constraints, cluster features, fit, candidate lane marks, verification and optimization, complex real conditions, robust.

I. INTRODUCTION

With the rapid development of intelligent sensing technology and Internet technology, autonomous driving has become a promising field. And the autonomous vehicle market has become a battlefield. The market share of vehicles is also rising. According to the United Nations, fatal accidents on the road that are not due to drunkenness are mainly caused by misjudgment and carelessness of drivers. Advanced driving assistance systems in intelligent transportation system (ADASITS) can alert the driver to dangerous situations, and can even take active measures when driving [1].

Both autonomous driving and ADASITS require the ability to analyze the road scene just as a human does. Drivers rely mostly upon their visual systems. Compared with active sensors such as LIDAR, a passive sensor such as an onboard camera is nonintrusive and has high resolution, low power — requirements, and low cost and it is easily integrated into the environment [2]. Lane-mark extraction is a basic task of road scene analysis. We can utilize extracted lane-mark information to localize, which determine the relative position between a vehicle and the road. Most existing methods based on computer vision lack robustness, and they are sensitive to complex driving environments and bad weather.

The objective of many studies [3]–[7] has been to extract lane marks using the specific color information and thresholds. These methods are effective only on structural roads under ideal conditions. Chao's method [8] used a simple K-means algorithm to segment the image. Ohashi's method [9] used the hill-climbing algorithm in the three-dimensional CIELab histogram of the image, and the K seeds for the K-means segmentation were automatically determined. When lane-mark edges are merged into shadows or

This work is licensed under a Creative Commons Attribution 3.0 License. For more information, see http://creativecommons.org/licenses/by/3.0/

dirt on the road, this method often performs unsatisfactorily. The authors of other studies [10]–[15] eliminated noise edges in complex scenarios in various ways. The objective was to retain the edges of the lane-marks and remove the residue edges of shadows. These methods are not robust when facing bad weather such as rain. Other researchers [16]–[22] remapped the road points in images coordinates into points in the same world coordinates by perspective transformation. It is necessary to obtain the parameters of onboard camera used to collect road images, including camera position, orientation, and optics. Most existing lane-mark extraction methods are in applicable in complex real conditions. Improving the robustness of lane-mark extraction is essential, complicated, and challenging.

In this paper, we propose a new lane-mark extraction algorithm consisting of four parts. To extract the features of a lane-mark more effectively, we use grayscale algorithm and fast median filter to preprocess a road image captured from an onboard camera. The lane-mark feature can be effectively extracted as one of the determinants of whether a lane-mark extraction algorithm has high robustness. Taking a series of lane-mark characteristics in road images as constraints, a lane-mark feature filter based on multi-constraint is used to extract lane-mark features. For the distribution characteristics of lane-mark features and the inevitable noise in feature point extraction, we propose a clustering algorithm based on the double removal of p-least squares algorithm. Then recursive dichotomy fits lane-mark features into the candidate lane-marks. These candidates are selected by a dynamical vanishing point. We optimize the lane's stability according to Image inter-frame correlation and Kalman filter, and we use tracking results of the Kalman filter to set a dynamic region of interest (ROI). This step avoids the waste of computing resources caused by operating on the whole road image and falsely detecting lane-mark features that can affect the final extraction results. In our experiment, we divide common complex road scenes into four categories, and we propose a method to evaluate the results of lane-mark extraction. The results show that the proposed method can extract lane marks robustly in various complex real conditions. The flowchart of our method is shown in Figure 1.

The remainder of this paper is organized as follows. Section 2 surveys some related work. Section 3 presents the preprocessing process. Section 4 explains lane feature filtering based on multi-constraints. Section 5 provides the details of the lane clustering and fitting algorithm. Section 6 explains verification and optimization of the candidate lane. Section 7 reports on our experimental results. Section 8 provides our conclusion.

II. RELATED WORK

Extracting lane marks based on computer vision is an important component of autonomous driving and ADASITS. Digital images are traditionally represented by a set of unrelated pixels. Therefore, valuable information is often buried in such unstructured data. Existing lane marks extraction methods based on computer vision are either region or feature-based.

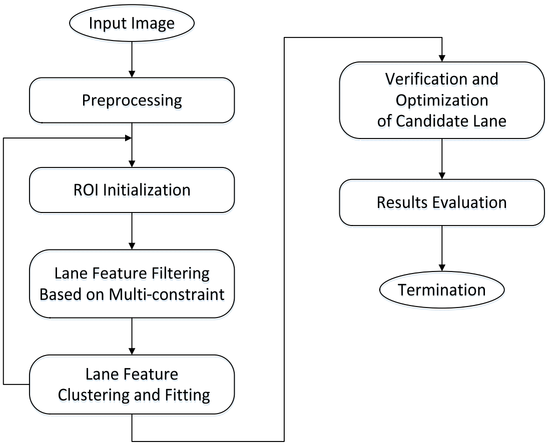

Input Image
Preprocessing
ROI Initialization
Lane Feature Filtering Based on Multi-constraint
Lane Feature Clustering and Fitting
Verification and Optimization of Candidate Lane
Results Evaluation
Termination

FIGURE 1. Flowchart of proposed method.

Region-based methods [3]–[7] consider lane-mark extraction as a problem of extracting the region of a lane-mark using specific color information and thresholds. This method is simple, and is only suitable for an ideal state such as a road with clear lane marks and no shadows and dirt etc. More adaptable and robust feature-based methods have been widely used in recent years. The algorithm proposed in this paper relies on a feature-based method. A feature-based method typically comprises four basic steps: image preprocessing, feature detection, fitting, and candidate validation and tracking, as shown in Figure 2.

Preprocessing involves removal of noise and preparation of the image for subsequent steps. Since objects are likely to exhibit certain invariant features in various real-world imaging conditions, image features that are stable across varying scales, rotations, illuminations, or viewpoints are desirable for recognition and indexing tasks.

Feature extraction is the extraction of lane-features in road images using various filters or statistical methods. Lane marks constitute a strong cue, since these they usually have clear edges and relatively high intensities. Algorithms such as edge distribution function [23], [24], directionally adjustable filter [25], canny filter [26]–[28], sobel filter [29], [30], etc. are often used in lane-mark detection.

The main ideal of fitting is to describe the lane-mark visually. The position of the lane-mark is displayed in the road image. Several common mathematical models such as straight line [31], quadratic curve [8], cubic curve [32] and recursive dichotomous [33] are used to fit lane marks. The results of fitting by common models are shown in Figure 2.

To achieve robust system performance, spatial and temporal domain constraints are widely used to validate features and obtain correct lane-mark locations. By using constraints of symmetry [34]–[36], continuity [36], [37], lane-mark orientation [38], [39], etc., the effect of noise caused by shadows, puddles, and tire skid marks on detection results can be alleviated to some extent. A lane tracking process uses

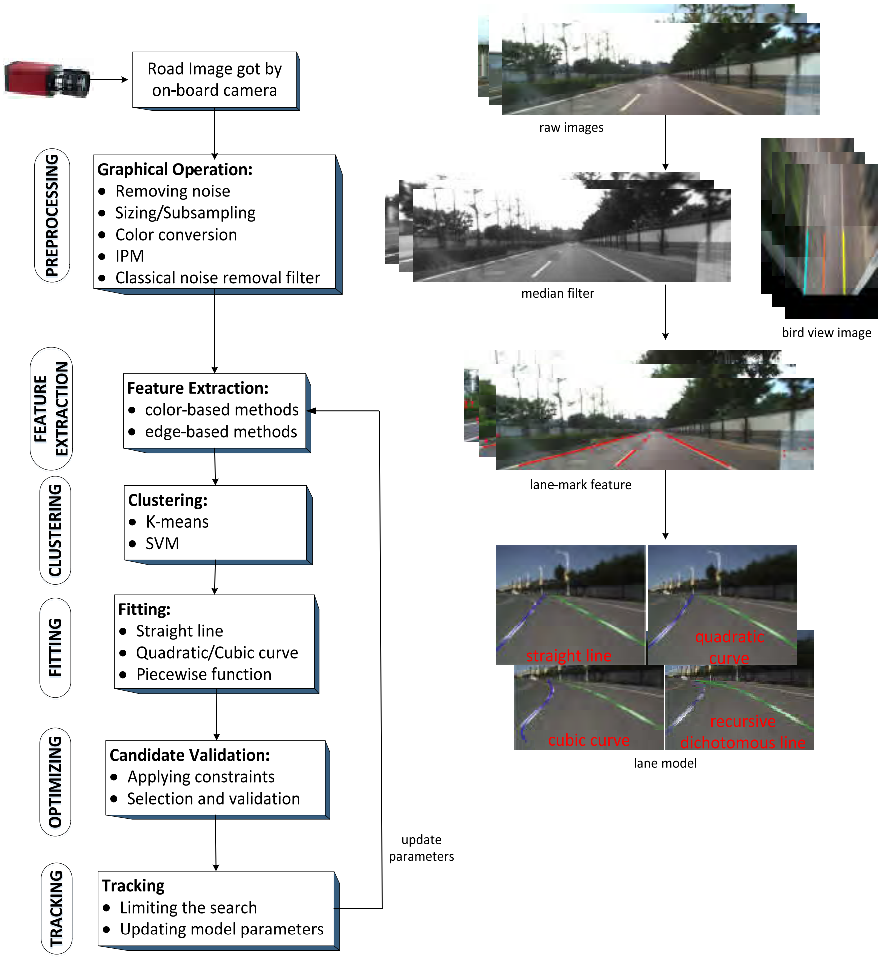


FIGURE 2. Basic steps of a feature-based method.

previous information to detect new features in an approaching image area [16], [36], and updates model parameters just in time.

III. PREPROCESSING

A. GRAYSCALE

Before performing the lane-mark detection process, we must convert the image from the RGB color space to a grayscale intensity image. On a real-road, a lane-mark is only painted in white and yellow, which form a strong visual contrast with the surface of the road. To enhance the character of lane information, we want to retain more white and yellow information. Therefore, we weaken the proportion of B channel component values. The equation is

$$G_{ray} = R \times 0.5 + G \times 0.5, \tag{1}$$

The following equation is usually used for grayscale:

$$G_{ray} = 0.30 \times R + 0.59 \times G + 0.11 \times B, \tag{2}$$

Different grayscale methods can affect the extraction of lane-mark features. We compared the effects of two different grayscale methods on a data set in this phase of lane-mark feature extraction. The test data set consists of 179 images of size $1,280 \times 1,024$ pixel. We counted the number of extracted lane-mark features using the two grayscale methods. The test results are shown in the following table.

TABLE 1. Number of extracted lane-mark features.

Grayscale method	Total number of features	Average number of features
Equation (2)	546,129	3,051
Equation (1)	570,473	3,187

In general, we can get more features after using Equation (1). Figure 3 shows extracted features after using Equation (2) and (1) respectively. The missing part of features are circled out in Figure 3. Our method can detect a subtle change of lane intensity values, especially when a white or yellow lane appears in the image. Therefore, it can enhance the accuracy of image processing.

B. FAST MEDIAN FILTER

Some methods [40] use a classical median filter to preprocess the image. A median filter is a typical nonlinear filter, which

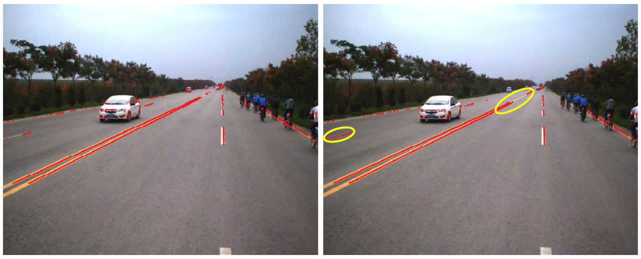


FIGURE 3. **Extracted features after using Equation (2) and (1).**

uses a unified method to deal with all statistics. It not only changes the value of the noise, it changes the value of the signal point, and spreads noise in the neighborhood.

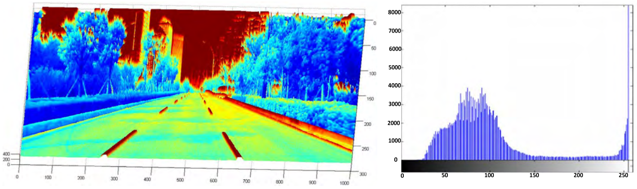


FIGURE 4. **Overall intensity distribution and histogram of lane gray value.**

As shown in Figure 4, when we analyze the overall intensity distribution and the histogram of lane gray values, it is not difficult to find that the gray distribution of road images is an approximately Gaussian distribution, and the gray value of the adjacent pixels has changed little.

The 3×3 field window used in this paper processes the signal and noise respectively, when using a fast median filter. The road surface portion that includes the lane-mark is preferentially processed. The window of the fast median filter starts from the bottom left of the road image and slides from left to right and bottom to top. We need not to calculate the median value of each pixel in the field window, just when the following condition holds

$$\{f_{N1}, f_{N2}, f_{N3}\} \neq \{f_{O1}, f_{O3}, f_{O3}\}, \tag{3}$$

We can then find the median f_m in the field window. In the above formula, f_{O1} f_{O2} and f_{O3} are sliding-out window pixel values, f_{N1} f_{N2} and f_{N3} are sliding-in pixel values. Figure 5 shows the direction of movement of the field window of the fast median filter.

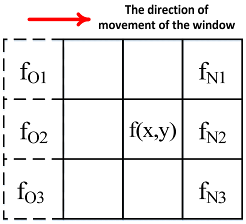


FIGURE 5. **Field window of fast median filter.**

Next, we must determine whether the central pixel in the field window is a signal or noise. If it is a signal, we do nothing and let $f_m(x, y) = f(x, y)$. If it is noise, we acquire the median. We judge noise and signals by the following function:

$$f(x, y) = \begin{cases} noise, & f == f_{\max} \,||\, f == f_{\min} \&\& |f - f_m| > T \\ signal, & else, \end{cases} \tag{4}$$

Where $f_{\max}$ is the maximum of the field window, $f_{\min}$ is the minimum, and T is the decision threshold. The results show that the filtering effect is better when T=30.

IV. LANE-FEATURE FILTER BASED ON MULTI-CONSTRAINTS

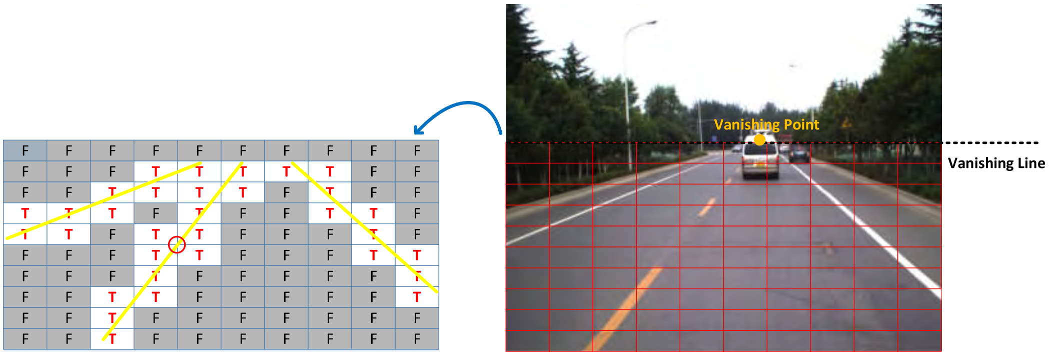
After image preprocessing, the traditional methods of lane-mark feature extraction [41], [42] perform well only when lane-mark edges are clear and pure. In method [43], Wu et al. used canny filter and opposite gradient orientation as a constraint condition to eliminate false-edges. However, it is not enough. Shadows, surface water, bumps, light, and other uncertainties increase the difficulty of lane-mark feature extraction in a real driving environment. These uncertain environmental factors will lead to false or missed detection, and affect the final result. Considering these factors, we decided to use adaptive threshold and lane feature filter based on multi-constraints to select and filter candidate features, and then obtain reliable lane-mark features.

A. ROI INITIALIZATION

The road area is generally located in front of the automobile and is not far from the vehicle as it moves forward [43]. This is why our ROI is selected on the bottom side of the image.

When the current input road image is the first frame or the Kalman filter is invalid, we conduct initialization operations of ROI. We set the field of lane-mark detection below 0.4H, where H is the height of the input road image. The purpose is to not only narrow the range of effective detection, but to exclude the interference of the background of the sky, trees and buildings on the road image. We use the cue that the location of the lane-mark may not shift too much between two adjacent image frames. As the onboard camera's fixed position changes or the sizes of the collected road images differ, the ROI can be adjusted accordingly. We divide squares of ROI into windows of interest (WOI), whose sizes adjusted adaptively with the change in size of the adaptive dynamic ROI. We make full use of the results of the Kalman filter and the location of the vanishing point to set an adaptive dynamic ROI, and use the vanishing line (the Y-axis of the vanishing point) as the border of the ROI.

We construct a table to label the blocks that are occupied or close to the lane-mark model from the result of the Kalman filter. Figure 5 shows the process of initializing adaptive dynamic ROI: yellow lines are the result of the Kalman filter calculated by previous frames. The windows enclosed by yellow lines are the possible WOIs for the present frame. The table plotted at the right of Figure 4 shows the status of the WOI. If the window is in the regions enclosed by yellow lines, then we will mark the window as T; otherwise, we will mark it as F.



F	F	F	F	F	F	F	F	F	F
F	F	F	T	T	T	T	T	F	F
F	F	T	T	T	T	F	T	F	F
T	T	T	F	T	F	F	T	T	F
T	T	F	T	T	F	F	F	T	F
F	F	F	T	T	F	F	F	T	T
F	F	F	T	F	F	F	F	F	T
F	F	T	T	F	F	F	F	F	T
F	F	T	F	F	F	F	F	F	F
F	F	T	F	F	F	F	F	F	F

FIGURE 6. Initialization process adaptive dynamic ROI.

When the lines pass through the intersection of the windows (the red circle), we set all the adjacent windows of this corner as T. Those WOIs that are set to T connect to the ROI we need. Next, we will extract the lane-mark features in the current frame of ROI.

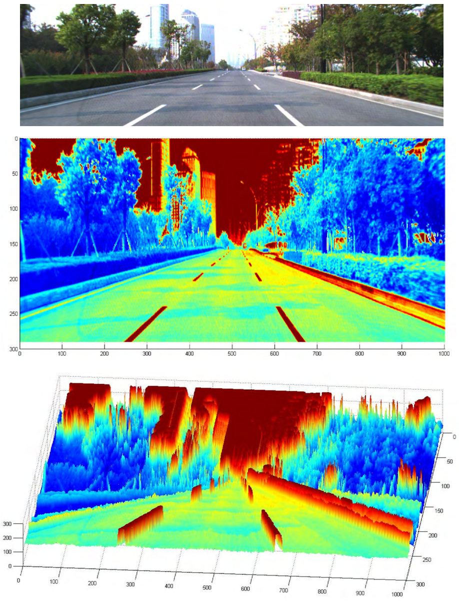


FIGURE 7. Overall gray intensity distribution.

B. PEAK-FEATURE CONSTRAINT

Figure 7 shows the overall intensity distribution of the road image where a darker color indicates a larger the gray value. As can be seen from the figure, the part of the lane-mark compared to the surrounding road surface has a higher brightness, and a greater magnitude of change. It is like the altitude distribution map, and it formulates a mountain-like lane area.

In the process of actual autonomous driving, an onboard camera will shake when the car hits bumps, resulting in the lens not being able to focus. Or, when an autonomous vehicle turns quickly, the rapid shift will cause blurred motion. In the overall intensity distribution of road images, these phenomena cause the lane-mark area to be fuzzy with an increased width. Due to the difference between pixels, multiple peaks will appear in the near range. As we show in Figure 8, it is necessary to combine the local adjacent peaks that satisfy certain conditions. If a peak has been detected in the vicinity (12-pixel field) of another detected peak and the value of the adjacent peak is closer ($\left|g_p - g_{p+1}\right| < 20$, where g_p and g_{p+1} are the respective gray values at the two peaks), we select the peak with the higher luminosity value as the reserve peak and all troughs between the adjacent peaks are deleted.

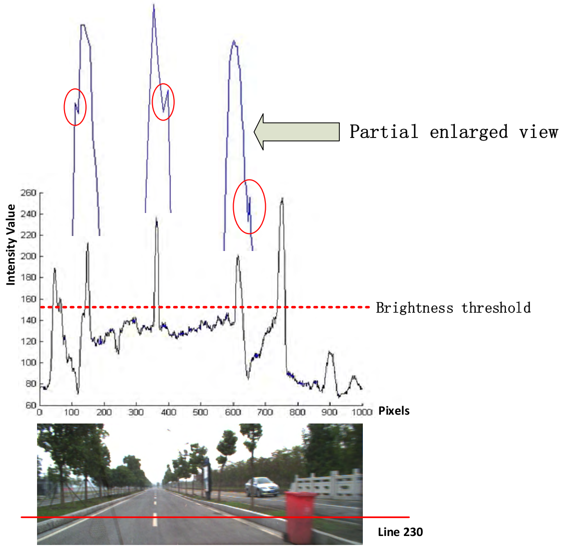


FIGURE 8. Line 230 of pixel gray scale change.

Account for the effects of light changes or distribution unevenness in peak-feature extraction, this paper designs a cosine function based on the proportion and translation transform to dynamically determine a brightness difference criterion threshold. The peak whose brightness difference is higher than the threshold is retained. The cosine function is expressed as follows:

$$T = \begin{cases} 10, & 0 \leq \mathrm{R}_i \leq 20 \\ 10 + \left[\cos\left(\frac{\mathrm{R}_i - 20}{160} \times \pi + \pi\right) + 1\right] \times 80, & 20 < \mathrm{R}_i \leq 180 \\ 40, & 180 < R_i \leq 255 \end{cases} \quad (5)$$

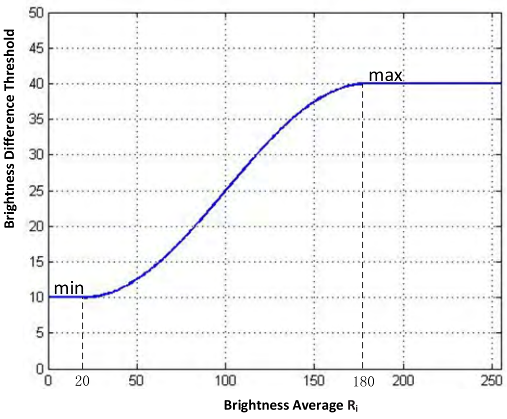


FIGURE 9. Brightness difference threshold function.

Red	$0 \le \theta < \frac{\pi}{2}$
Green	$\frac{\pi}{2} \le \theta < \pi$
Blue	$-\pi \le \theta < -\frac{\pi}{2}$
Yellow	$-\frac{\pi}{2} \le \theta < 0$

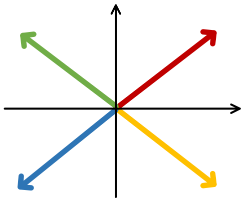


FIGURE 10. Color table.

Shadows are often formed by trees on both sides of the road. At the shadow junction, the brightness shows as dark - bright - dark. Therefore, the brightness value at the trough cannot be too low. In this paper, we only retain the trough that satisfies $g_p > 0.4 \times R_i$ corresponding to the peak, where R_i is the brightness difference threshold of line i.

C. GRADIENT ORIENTATION CONSTRAINT

The lane-marks consist of two parallel sideline. In the gray scale of the road image, the gray value appears to jump. As shown in Figure 7, it corresponds to the trough formed by both sides of the peak. Therefore, while we extract the peak characteristics, we also recode the troughs on the left and right sides of the peak. In other words, trough characteristics should emerge in pairs.

The two parallel sidelines of a lane-mark have opposite gradient orientations [44]. The gradient orientation $G(x, y)$ of an edge pixel can be represented as:

$$G(x, y) = \tan^{-1}(\frac{d_y}{d_x}), \tag{6}$$

Where d_y and d_x are the gradients in the vertical and horizontal directions, respectively.

We estimate the number of possible edge-orientations in four levels by using the four quadrants. The pixels with edge-orientation in the first, second, third, and fourth quadrants are plotted in red, green, blue, and yellow, respectively. The color table is shown in Figure 10.

D. LANE-MARK "LOCATION-WIDTH" CONSTRAINT FUNCTION

The lane-mark has vertical characteristics that extend forward and horizontal characteristics that have a certain width in the road image, which is the most important point [45]. According to national standard GB 5768.3-2009 [46], the horizontal width of a lane-mark is generally 10∼15cm in China.

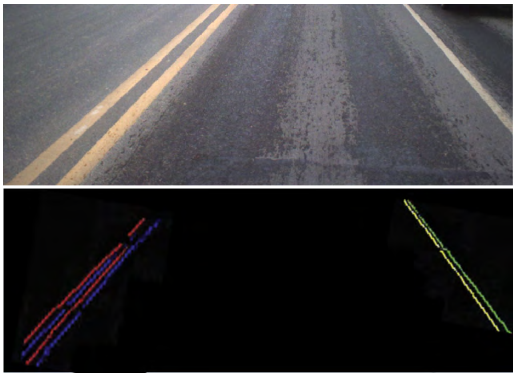


FIGURE 11. Sideline orientation color for ROI.

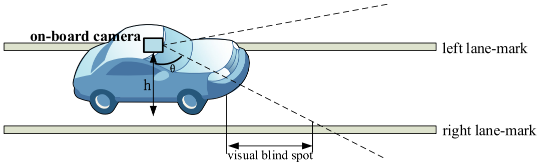


FIGURE 12. Picture capture diagram of onboard camera.

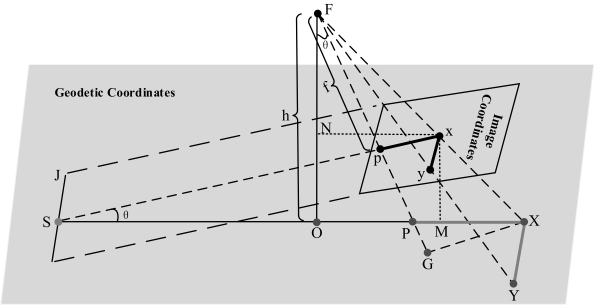


FIGURE 13. Diagrammatic sketch for road images captured by onboard camera.

An onboard camera mounted behind an autonomous vehicle windshield has a certain height. Its line of sight presents an overlooking state from top to bottom, and it has a certain inclination. Figure 12 shows the schematic illustration of an onboard camera capturing road images, where H is the height of the camera from the ground.

The mathematical model for Figure 12 is abstracted as follows:

As Figure 12 shows, we establish the geodetic coordinate OXY, where the point O is the origin, OX is the x-axis, and the XY direction is the y-axis direction. F is the focus of the onboard camera. Line FO is perpendicular to plane OXY and the foot of a perpendicular is point O. In the image coordinate, *pxy* is the image plane. Connect point F to point Y and point F to point X. The segment FY and FX intersect the image plane at y and x, respectively.

According to the spatial geometry and the triangular similarity principle, it is easy to arrive at the following conclusions: any two parallel straight lines mutually mapped to the image coordinate are bound to intersect at one point. We refer to the intersection as the vanishing point of the lane-mark. The vanishing point can be expressed as:

$$(x_{vp}, y_{vp}) = (\frac{f}{\tan\theta}, \frac{kf}{\sin\theta}), \tag{7}$$

where f is the focal distance, θ is the inclination of sight, and k is the linear slope in the geodetic coordinate. Therefore, the road images captured by an onboard camera always have a strong perspective effect. The lane-marks in road images have the characteristics that the closer the distance, the larger the width. That is lane-marks at the bottom of the image look wider. With increased distance from the bottom of the image, the lane-marks look narrower.

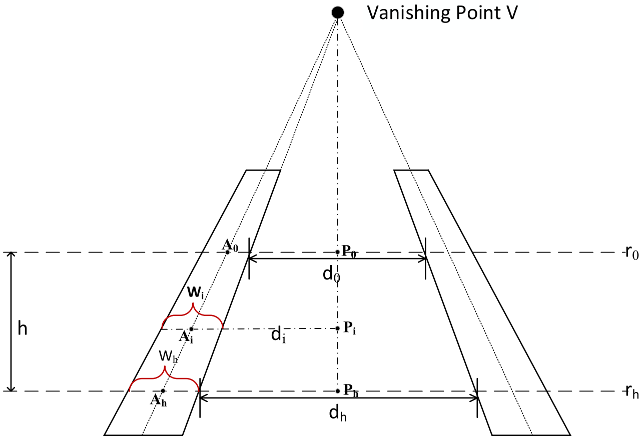


FIGURE 14. Lane-mark "position-width" function.

We researched on the perspective effect of lane marks in the road image. The corresponding mathematical model is established as shown in Figure 14. We use the spatial geometric relationship to get the lane-mark position-width function:

$$W_i = (A_iP_i - d_i) \times 2, \tag{8}$$

where $A_iP_i = \frac{VP_0+i}{VP_0+h} \times (W_h + d_h) \times \frac{1}{2}$, $d_i = \frac{VP_0+i}{VP_0} \times d_0$, $VP_0 = \frac{d_0}{d_h-d_0} \times h$.

V. LANE FEATURE CLUSTERING AND FITTING

A. FEATURE CLUSTERING

1) REMOVAL OF OUTLIERS

The Hough transform (HT) and least squares (LS) are two common algorithms used to obtain straight lines. In the HT, the problem of straight-line detection is transformed into the peak problem in the Hough space [28], [35], [40]. By the constraints of the algorithm itself, HT has some limitations [47]. When the HT is applied to noise-containing features, the number of units in the coarse resolution will be distributed to several smaller discrete parameter units under fine resolution. If the resolution $\Delta\rho$ is too small, the accumulator cannot accumulate enough votes to cause the detection to fail. Therefore, the HT can only achieve limited detection accuracy due to the influence of noise. LS obtains an exact line of the given data set in the sense of mean square error [27], and LS is extremely sensitive to outliers that deviate from the regression line. The introduction of strong noises improves the standard deviation of the entire data set, which causes the noise points to be easily mistaken for normal data points, so that these outliers cannot be removed.

Algorithm 1 Algorithm for Removal of Outliers

```
Input:  Hough transform parameter list (delta-Rho,
        delta-Theta etc.)
        Extracted feature points set P={$P_1$, $P_2$, ..., $P_n$}
        Distance error limit $d$, mean error threshold $\varepsilon$
Output: Parameters of the straight line and all the points
        on the line
Step 1: Initialize the straight lines set L = {$L_1$,$L_2$,...,$L_l$}
        by using progressive probabilistic Hough
        transform
Step 2: For every line $L_i \in$ L, do:
            Find the feature points in feature points set P
        that are not greater than distance d from line $L_i$.
        These feature points constitute set $S_i$
        = {$P_1$, $P_2$, ..., $P_m$}
Step 3: Use the least squares method to determine the
        regression line parameters $k$ and $b$, and the
        mean square error $e$ of the set S
Step 3: For every point $P_j(x_j, y_j) \in S_i$, do:
    Step 3.1: All the feature points that satisfy condition
              $kx_j + b > y_j$ form a subset $S_{pos}$
              All the feature points that satisfy condition
    $kx_j + b < y_j$ form a subset $S_{neg}$.
    Step 3.2: Select point $P_p$ and $P_n$ from set $S_{pos}$ and
              $S_{neg}$ respectively.
              $P_p = \arg\max_{P_p \in S_{pos}} d(P_p)$,
              $P_n = \arg\max_{P_n \in S_{neg}} d(P_n)$.
              where $d(P)$ represents the distance from
              the point $P$ to the regression line
    Step 3.3: if ($e > \varepsilon$)
                  Remove point $P_p$ and $P_n$, update set $S_i$
                  and feature points set P
```

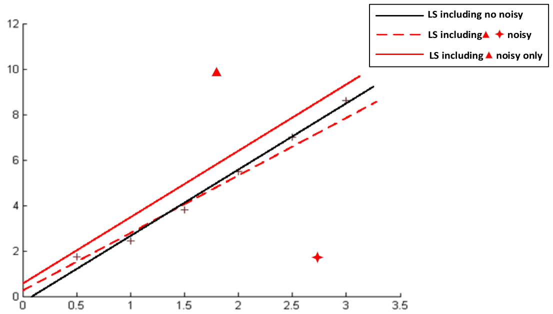


FIGURE 15. Fitting noise-containing data by LS.

As shown in Figure 15, when these data points are relatively concentrated but include some noise points that are

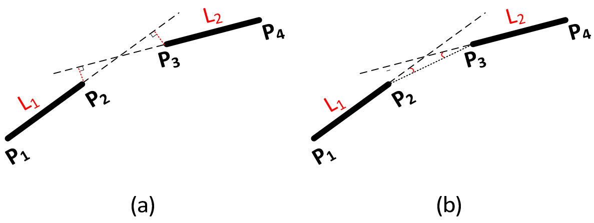


FIGURE 16. (a) Distance similarity (b) Direction similarity.

equal in intensity and distributed on both sides, the new regression line will be more biased in the opposite direction after removing noise from one side. Therefore, it is more difficult to remove the noise points in the opposite direction. To ensure a final regression line that mainly includes normal data points with error near zero, we always remove noise points of error maximum not only on a single side, but simultaneously on both sides of the regression line.

In complex real conditions, the lane-mark will inevitably introduce noise in the feature-extraction stage. We primarily consider how to cluster effectively and determine the location of the lane-mark when the data points contain noise. Considering the high efficiency and drawbacks of HT and LS, we propose a new straight-line detection algorithm that combines with the advantages of the two algorithms and can eliminate outliers.

2) SIMILARITY MEASUREMENT

To determine the affiliation of these straight lines, two similarity measures called distance similarity and direction similarity are introduced. Figure 15 is a schematic representation of these two kinds of similarity measure.

$P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ are the two end points of line L_1. $P_3(x_3, y_3)$ and $P_4(x_4, y_4)$ are the two end points of line L_2. The angles θ_1 and θ_2 are the respective inclination angles of line $L_1$and line L_2. The inclination angle of a straight line defined by points P_2 and P_3 is θ. We use the following formula to measure the approximate consistency of two straight lines in distance and direction.

$$\begin{aligned} dis &= |(x_3 - x_2)\sin\theta_1 - (y_3 - y_2)\cos\theta_1| \qquad (9)\\ &\quad + |(x_3 - x_2)\sin\theta_2 - (y_3 - y_2)\cos\theta_2|, \\ dir &= |\theta_1 - \theta| + |\theta_2 - \theta|, \qquad (10) \end{aligned}$$

These straight lines with approximate consistency in distance and direction are considered part of the same lane-mark. Those points on the lines that belong to the same lane-mark are clustered into the same category. Next, we must describe lane-mark and determine its location by using the recursive dichotomy method, which fits those feature points that belong to the same category.

B. LANE FITTING

The accuracy of a simple mathematical model used to describe a lane-mark is usually low, but a complex mathematical model is terrible in real time. Considering the accuracy and real-time performance comprehensively, a new lane-mark description method named recursive dichotomy is adopted in this paper. A horizontal straight line divides the ROI of the road image into two sub-regions. We use a straight line to fit each sub-region, so we can use polylines consisting of straight lines to describe the lane-mark. We use a rationality index (denoted by R) as a basis for whether this region should be re-subdivided, and whether it is reasonable to select the designated dividing line. We then use every sub-region as an input region and recursively implement this method.

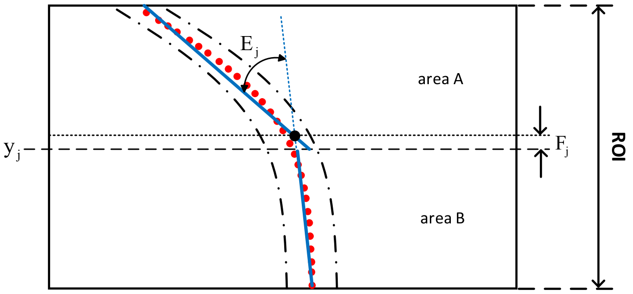


FIGURE 17. Rationality index of recursive dichotomy.

As shown in Figure 17, when the division of time j is carried out, the line y_j divides the ROI into two sub-regions (regions A and B), and straight-line equations are obtained by the least squares algorithm in the two sub-regions.

$$\begin{cases} y = k_a x + b_a \\ y = k_b x + b_b \end{cases} \qquad (11)$$

where the point $P_j(p_x, p_y)$ is the intersection of two straight lines, F_j represents the longitudinal distance from the intersection to line y_j , i.e., $F_j = |y_j - p_y|$; and E_j represents the slope difference between the lines in the two sub-regions, i.e., $E_j = |k_a - k_b|$. When the angle between the two lines is the largest after division and the intersection is on the corresponding dividing line y_j, that is to say, these data points in the region is most suitable for polyline to express. Then, we obtain the index R by the equation

$$R_j = \frac{E_j}{\mathrm{E}} \times \frac{\mathrm{F}}{F_j}, \qquad (12)$$

The larger the value of R_j, the more ideal the subdivision of the corresponding sub-regional division. E and F are the normalized factors: $E = \max(E_j)$ and $F = \min(F_j)$.

VI. VERIFICATION AND OPTIMIZATION OF CANDIDATE LANE

To improve the reliability of the algorithm and obtain more robust extraction results, we must verify and optimize its lane-mark detection result.

A. SELECTION BY DYNAMIC VANISHING POINT

The vanishing point reflects the direction information of parallel lines in a three-dimensional scene. It is an important clue to understand the three-dimensional scene from a two-dimensional image space. Therefore, using the vanishing point as a constraint condition can to some extent, select lane marks and filter out the part that is not. It can also obtain more accurate extraction results. Figure 18 shows the schematic of vanishing point selection.

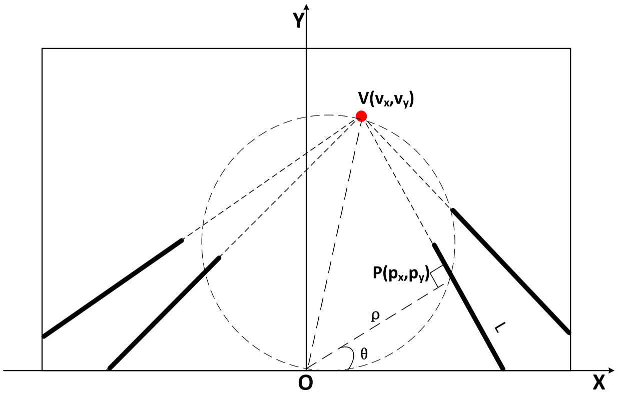


FIGURE 18. Schematic of vanishing point selection.

As shown in Figure 18, we can establish the relationship between the line-marks of a road image and the vanishing point in the coordinate system OXY, where the point O is the midpoint of the width of the road image. The vanishing point coordinates of the current frame are $V(v_x, v_y)$. The line L is the candidate lane-mark. Draw a line perpendicular to line L through the origin o. The coordinate of the foot of the perpendicular is $P(p_x, p_y)$, the length of the perpendicular line is ρ, and the inclination angle is θ. According to the basic geometric properties of circle, the foot of perpendicular P must be on the circle whose diameter is the line determined by the origin o and vanishing point V. Thus, the following equations can be obtained:

$$\begin{cases} x^2 + y^2 - (xp_x + yp_y) = 0 \\ p_x \cos\theta + p_y \sin\theta - \rho = 0, \end{cases} \tag{13}$$

Obviously, the vanishing point V is one of the solutions of this equation group. The objective function is constructed as follows

$$\Delta\rho = \left| v_x \cos\theta_i + v_y \sin\theta_i - \rho_i \right|, \tag{14}$$

where the parameters θ_i and ρ_i of the line L_i must be determined. We can solve for $\Delta\rho$ though the objective function. When $\Delta\rho$ is in a small range, the corresponding candidate lane-mark is effective. We apply the property of the vanishing point for a constraint condition. It filters remarkably well, especially for scattered interference lines.

B. LANE STABILITY OPTIMIZATION

1) IMAGE INTER-FRAME CORRELATION

In the actual acquisition system of autonomous driving, an onboard camera directly obtains the video-stream information. Therefore, it has significant redundancy between two adjacent frames in the video stream [37]. The motion of a vehicle has continuity in time and space. Because the sampling frequency of the onboard camera is large (about 100 fps), the vehicle only moves a short distance during a sampling period, and the change of road scene is tiny between adjacent frames. Thus, the previous frame provides strong location information on lane-marks for the next frame. Thus, the correlation between adjacent frames called inter-frame correlation is introduced in this paper.

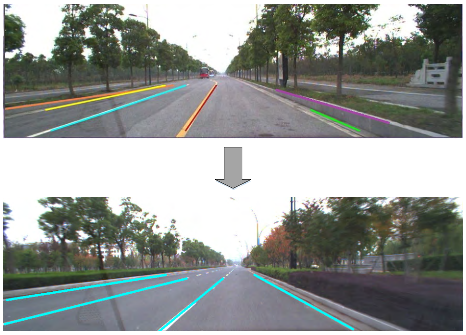


FIGURE 19. Function of filtering by vanishing point.

The number of lane marks detected in the current frame is m, and they are expressed in a set $L = \{L_1, L_2, \cdots, L_m\}$. The number of lane marks saved in the historical frame is n, and they are expressed in a set $E = \{E_1, E_2, \cdots, E_n\}$. We first establish a matrix $C = m \times n$, where element c_{ij} of matrix C represents the distance Δd_{ij} between the ith line L_i in the current frame and the jth line E_j in the historical frame. Δd_{ij} is calculated as follows:

$$\Delta d_{ij} = [|x_i^{L_A} - x_j^{E_A}|, |x_i^{L_B} - x_j^{E_B}|]^T \in R^2, \tag{15}$$

where A and B represent the ends of lines L_i and E_j, respectively.

Then we count the number e_i in the ith row that satisfy $\Delta d_{ij} < T$ in the matrix C. If e_i is less than 1, then the current lane-mark does not associate with the lane-marks of the previous frame. Therefore, the current lane-mark is used as a new lane-mark to update the information of the historical frame of the next frame. If e_i is equal to 1, then L_i of the current image is correlated with lane-mark E_j of the previous frame. If e_i is greater than 1, then the vector V_i is used to record the position of the lane-mark satisfying the conditions in the ith line of the current frame, namely:

$$V_i = \{v_{i1}, \cdots, v_{ij}\}, v_{ij} = \begin{cases} 0, & \Delta d_{ij} > T \\ \Delta d_{ij}, & other, \end{cases} \tag{16}$$

In vector V_i, we count the V_j in the jth column with nonzero elements, and determine the smallest element of V_i:

$$(\Delta d_{ij})_{\min} = \min\{V_j\}\ (V_j \neq 0), \tag{17}$$

When $\exists(\Delta d_{ij})_{\min} > 0$, we consider that lane-mark L_i of the current image matches lane-mark E_j of the historical frame.

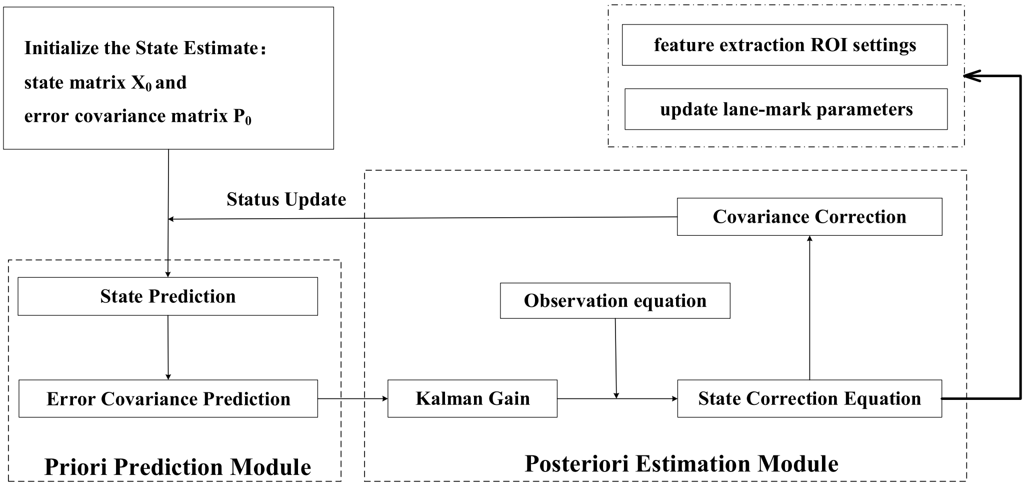

Initialize the State Estimate:
state matrix X_0 and
error covariance matrix P_0
feature extraction ROI settings
update lane-mark parameters
Status Update
Covariance Correction
State Prediction
Observation equation
Error Covariance Prediction
Kalman Gain
State Correction Equation
Priori Prediction Module
Posteriori Estimation Module

FIGURE 20. Kalman filter module diagram.

When the lane-mark detected by the current frame is in accord with the inter-frame correlation constraint, we consider that the current and the previous lane-marks are the same, and the position of the current lane-mark is displayed. Otherwise, the lane-mark detected by the current frame will be abandoned. If the cumulative abandonment instance of the inter-frame correlation constraint exceed T ($T = 3$), the parameter of the lane-mark of the corresponding historical frames will be updated.

2) KALMAN TRACKING

In the process of real-time acquisition of road images, the motion of the lane-mark between adjacent frames is slow, and we regard it as approximatively uniform linear motion.

We rebuild the state of the system through the value of observation and have recursion in order of "forecast-observation-correction" to eliminate random interference from systematic observations. Then we recover the natural characteristics of the original signal from the disturbed signal through observation to obtain a reasonable prediction of the location of the lane-mark in the next frame. We express the state vector of Kalman filter as

$$x_k = [x(k), y(k), v_x(k), v_y(k)]^T, \tag{18}$$

where $x(k)$ and $y(k)$ represent the center coordinates of the objective lane-mark, and $v_x(k)$ and $v_y(k)$ represent the moving speed of the point in the horizontal and vertical direction respectively. Kalman filter consists of two basic modules shown in Figure 20.

The prior prediction module establishes an priori estimate of the current system state using a time-renewal equation. The values of the current state variables and the error covariance estimates are calculated forward in time so as to construct a priori estimates for the state at the next moment. The posterior correction module is primarily responsible for feedback. We use the state estimation equation to make an optimal posteriori estimate of the current system with the a priori estimates obtained in the priori prediction module and the observed values of the current system. We update the parameters of the current system state and return to the next time.

VII. EXPERIMENT

A. PLATFORM INTRODUCTION

To verify the robustness and effectiveness of our proposed algorithm, the detection effects in the complex real-road environment are validated based on the platform of an autonomous vehicle. Figure 21 shows the autonomous vehicle Jing-Long I of Beijing Union University.

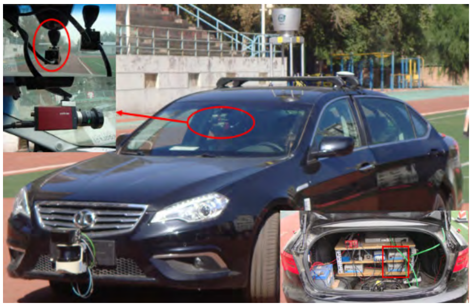
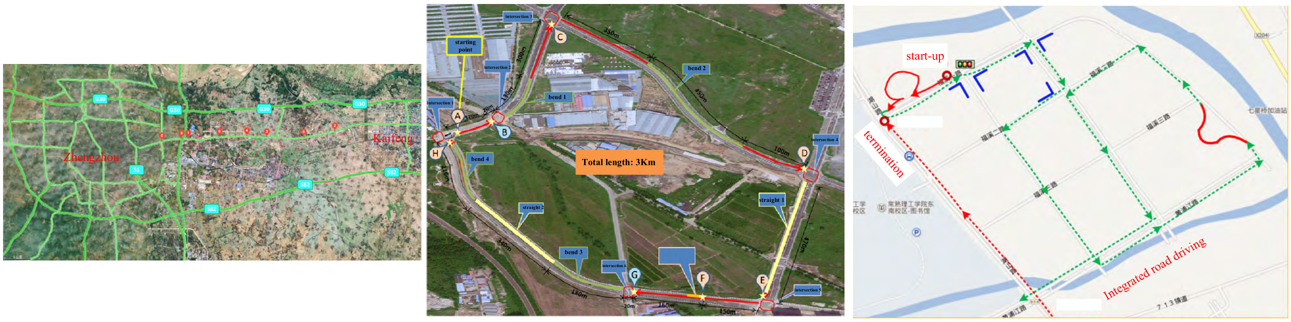


FIGURE 21. Experiment platform of autonomous vehicle.

In this platform, the Pike F100C industrial digital onboard camera is 1 meter from the ground and mounted in the central position just below the windshield of the autonomous vehicle. The optical axis of the camera is parallel to the plane of the vehicle chassis and is straight ahead of the moving vehicle. The size of the collected image is $1,000 \times 290$ pixels. The collection period is 10ms. We use an industrial computer of the GEMOTECH series with an i7 processor, 2.67GHz CPU and 8G memory. It resists high temperature and humidity, seismic, activity, and effects of dust, and has electromagnetic compatibility.

TABLE 2. Classification of main factors of influence in a real driving environment.

Index	Category	Included Factors	Supplement
1	Interference of Pavement Signs	common road traffic signs	zebra crossing, stop line, turn arrows
		speed limit mark	
		deceleration zone	
2	Shelter	pedestrian shelter	
		vehicle shelter	front vehicles shelter the lane-mark
		water droplets shelter	water droplets on the windshield when driving in rain.
		wiper shelter	
		others	cones shelter the lane-mark
3	Light Factor	evening driving	the light is dark
		night driving	vehicle lights up
		backlight	
		strong exposure	driving out of an overpass hole.
		strong light condition	when the light is too strong, the lane-mark is shiny
		counter vehicle lighting	driving in fog, counter vehicle has open headlamp
4	The Interruption of Other Traces on Pavement.	road damage or fouling	cracks, obsolete abandoned lane marks, mud and so on
		shadow	shadow of trees on both sides of the road.
		water stains	after rain
		small amount of snow	light accumulation of snow on the road forms ruts



(a) (b) (c)

FIGURE 22. Data collection scene.

The collected images of different road scenes in a real driving environment are divided into four categories, as shown in Table 2:

The data sets are mainly from four scenes: (1) The avenue of Zhengkai, which links Zhengzhou and Kaifeng in Henan province in China, a total length of 39.2km. These images

TABLE 3. Statistics of the data sets collected at different categories of different scenes.

scene	The index of influenced factors	The quantity of data set	Total number of frame
The avenue of Zhengkai	1	6	20078
	2	9	34125
	3	8	25100
	4	5	18700
The test area of Garden Expo Park	1	12	48792
	2	5	16254
	3	11	42630
	4	6	20421
The competition site	1	5	18036
	2	3	13230
	3	6	22780
	4	4	15427
traffic scene database TSD-MAX	1	25	2355
	2	16	1450
	3	18	1626
	4	13	1237

are 1, 000 × 1, 000pixel. (2) The test area of Garden Expo Park in Fengtai district of Beijing, a total length of 3km. The size of these images is 1, 000 × 290pixel. (3) The competition site of the 7th China Autonomous Driving Future Challenge (2015), in Changshu, Jiangsu Province. These images are 1, 000 × 290pixel. (4) The TSD-MAX traffic-scene database provided by Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. TSD-MAX provides a training data set for the offline test of the cognitive basic ability in a visual information environment in the 8th China Autonomous Driving Future Challenge (2016). These images are 1, 280 × 1, 024pixel. The data collection scenes are shown in Figure 22.

In the test area of Garden Expo Park, we spent three years, 2015 to 2017, collecting data from different seasons, weather, and times in the actual test tasks. These data contain almost all complex real driving conditions. There is also a small amount of experimental data from the daily travel section, such as the highway sections from the school to the Garden Expo Park and from the school to the Beijing International Airport.

We count the data sets collected by different categories of different scenes, and the statistical results are listed in Table 3.

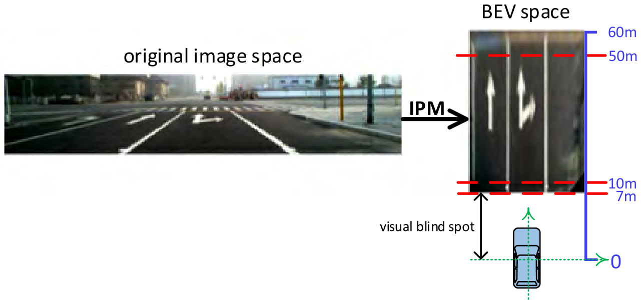


FIGURE 23. Conversion to BEV space.

B. EVALUATION STANDARDS

The position of a lane-mark is expressed by ordered points under the image coordinate system. Adjacent points are connected to form polylines as the position information of extracted lane marks. Since lane marks in road images often have a perspective effect, we perform an inverse perspective transformation (IPM) on the original image and converse coordinate system of the original image into the bird view space (BEV). The conversion process is shown in Figure 23.

To ensure a reasonable evaluation, we ignore the area where the vehicle position is too close to control the vehicle

TABLE 4. Evaluation score of partial extraction results.

Data sets	Marked true value	Number of detected lane-marks	Positive check TP	False check FP	Missed check FN	Score S
TSD-Lane-00000	546	452	271, 786	1, 304	10, 831	0. 95
TSD-Lane-00001	539	506	179, 881	2, 257	10, 676	0. 93
TSD-Lane-00002	539	540	254, 350	3, 767	10, 544	0. 94
TSD-Lane-00003	540	532	270, 290	4, 525	16, 022	0. 93
TSD-Lane-00004	495	417	89, 528	6, 388	8, 273	0. 86
TSD-Lane-00005	613	456	239, 607	1, 439	5, 813	0. 97
TSD-Lane-00006	365	221	49, 770	1, 129	2, 178	0. 93
TSD-Lane-00007	423	299	68, 105	1, 826	3, 918	0. 92
TSD-Lane-00008	431	188	37, 780	2, 853	218	0. 92
TSD-Lane-00009	659	464	87, 419	654	832	0. 98
TSD-Lane-00010	523	366	70, 846	2, 415	4, 435	0. 91

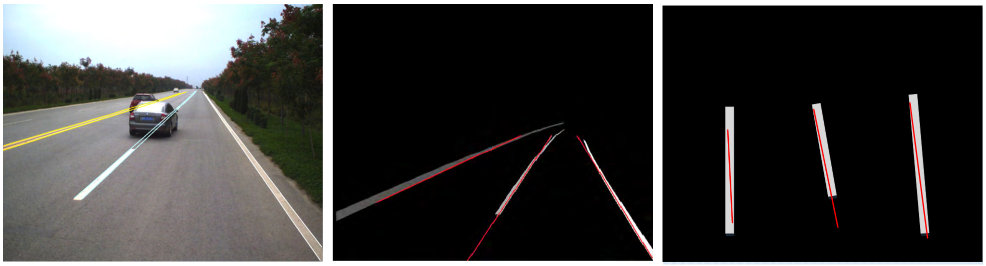


FIGURE 24. Standard for evaluation of lane-mark location.

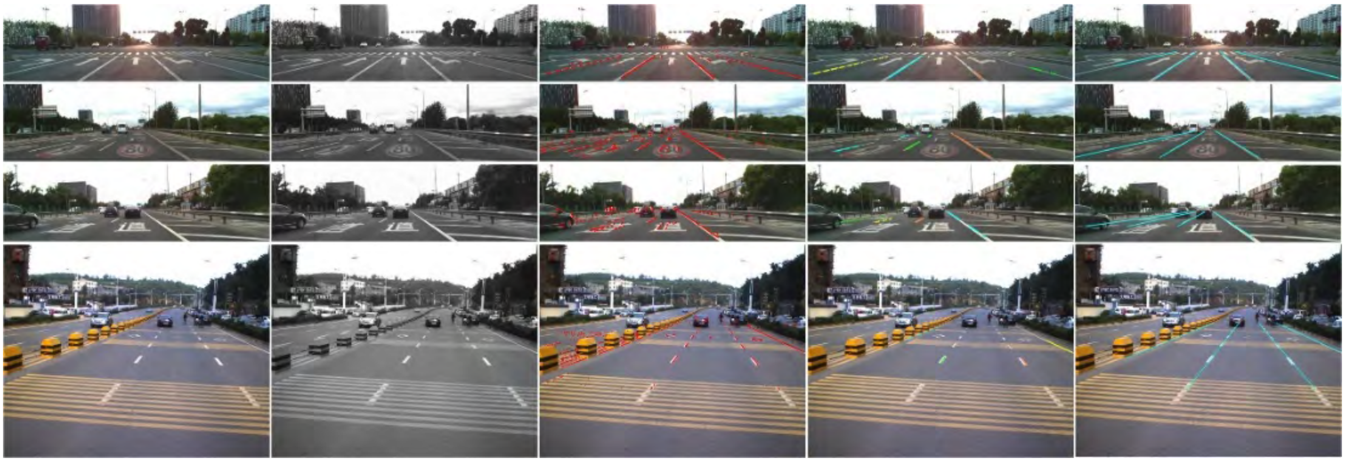


FIGURE 25. Interference of Pavement Signs.

and the area where the vehicle position is far enough away that it can easily lead to greater error. Thus, the stipulated scope of the evaluation is 10 m to 50 m from the front of the current vehicle.

According to Chinese National Standard GB5768.3 – 2009, the width of each lane-mark is usually 15~20cm. The total width of a double-yellow solid lane-mark is usually 40~60cm (two yellow solid lines with a width of 15 cm and 10~30cm interval). One pixel in the original image corresponds to a horizontal extent of about 5 cm in the BEV space. During the evaluation, the position of the lane-mark is characterized by its mid-axis. At the above threshold, it tolerates

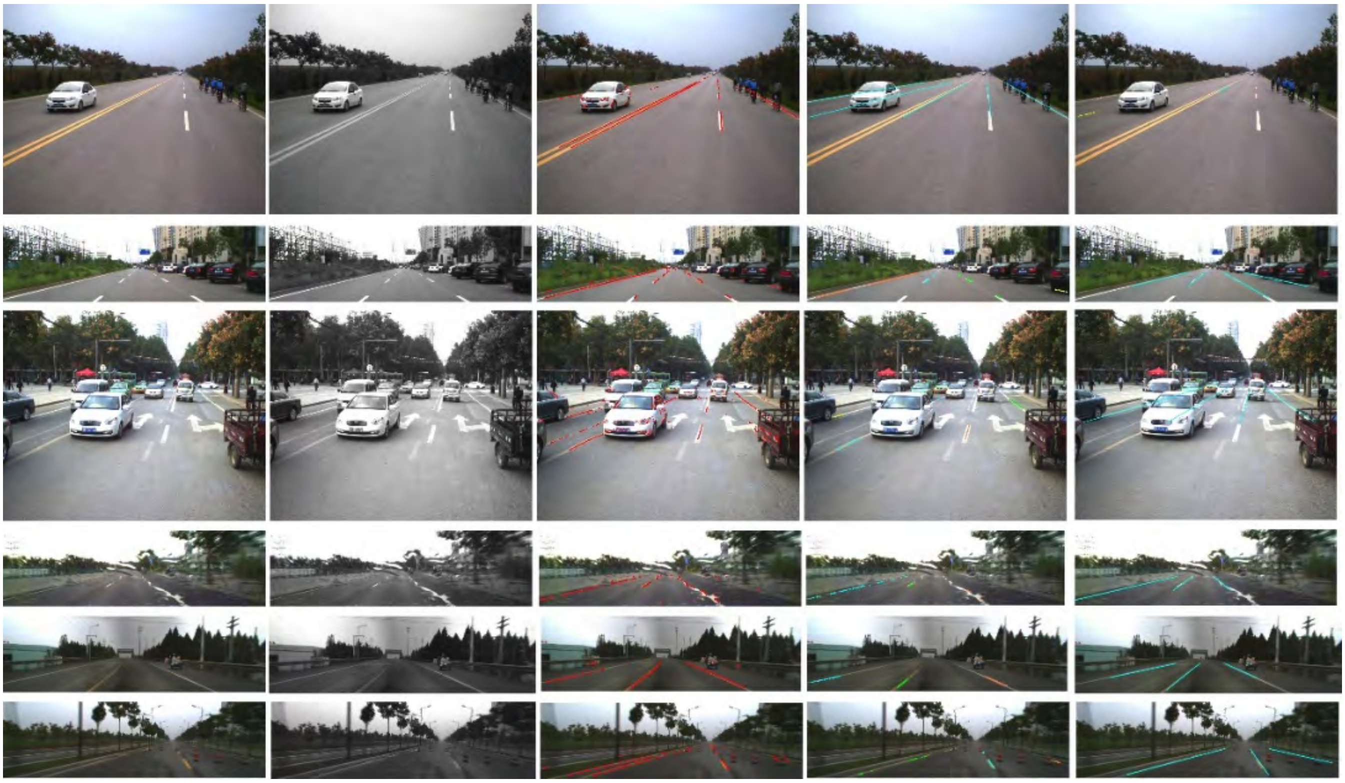


FIGURE 26. Shelter.

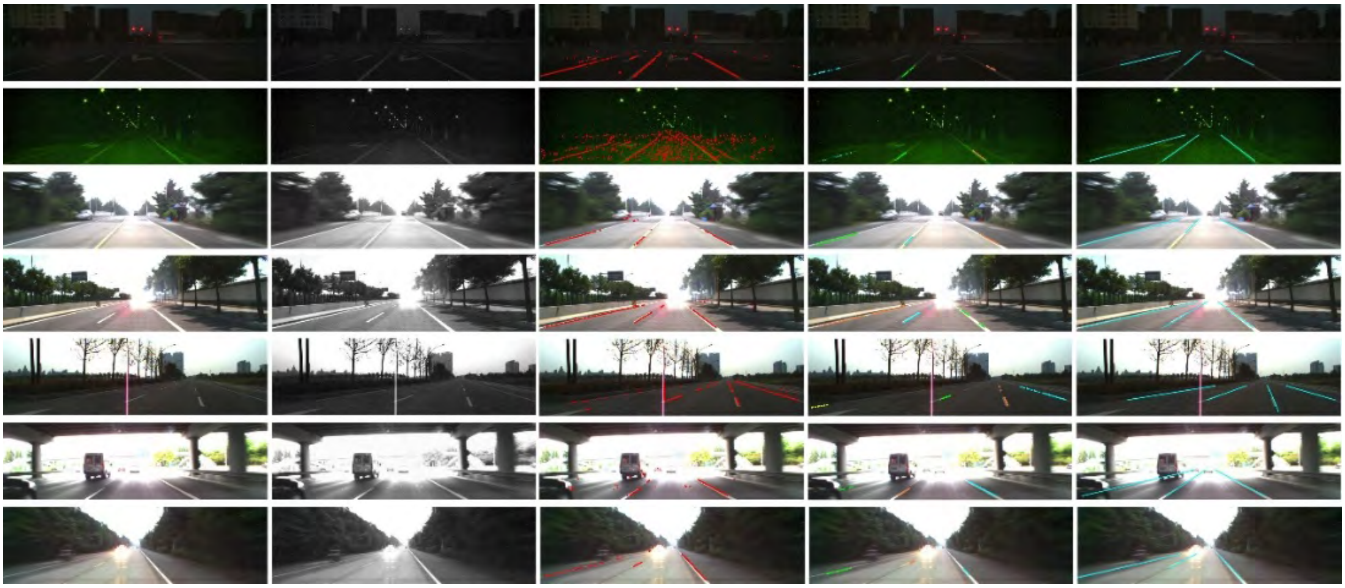


FIGURE 27. Light factor.

1∼3 pixel deviation between the distal end of the detection position of the lane line and the actual edge in the original image.

We refer to the scoring rules of the Traffic Scene Cognitive Basis Ability Offline Test and correspondingly formulate our evaluation standards of lane-mark detection results. In our evaluation standards, we remove the judgment on lane-mark type (yellow solid lane, yellow dotted lane, white solid lane and white dotted lane). The standard for an evaluation lane-mark location is defined as follows. First, extracted lane-marks and corresponding marked real values will be rasterized into a set of points. The number of points for which the

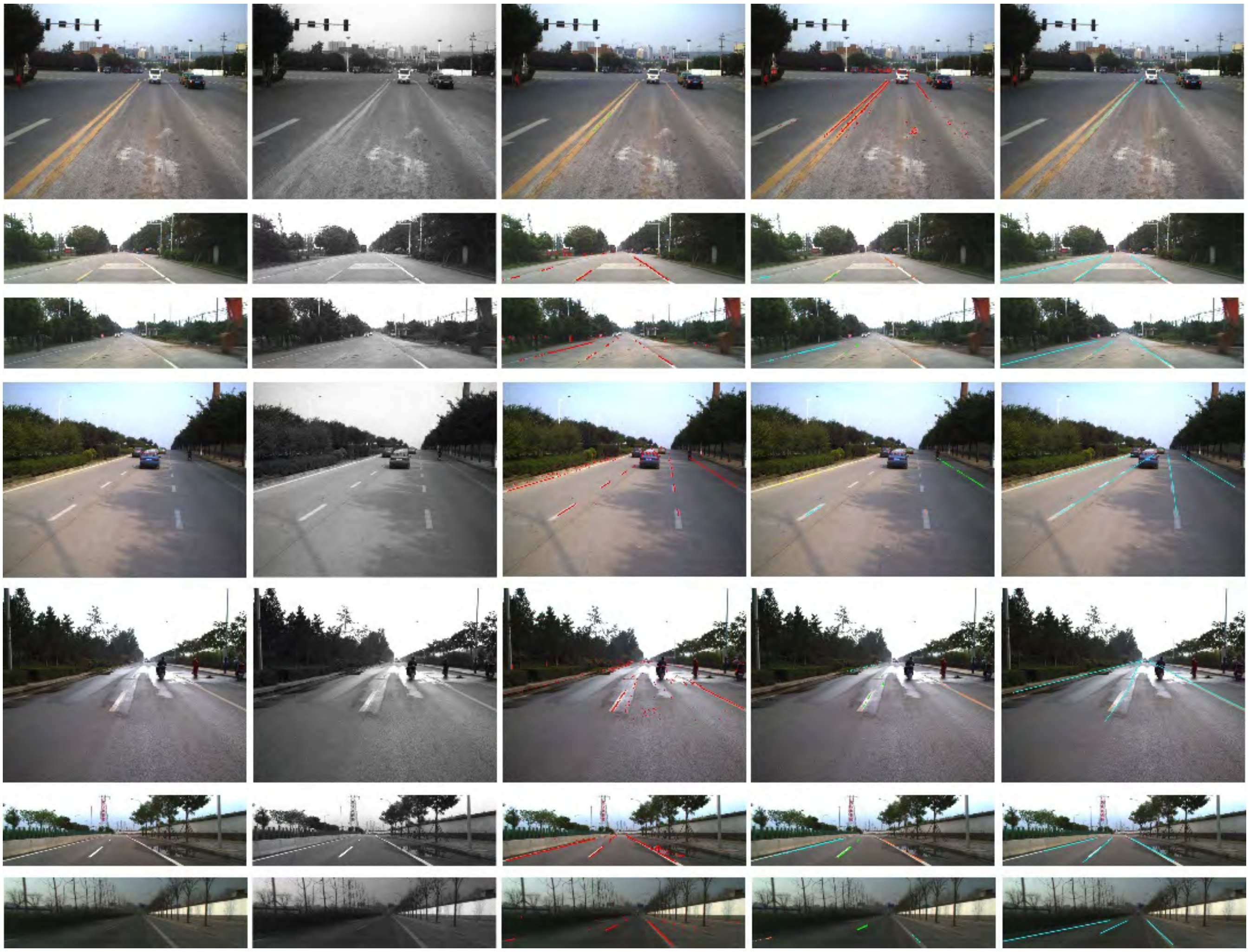


FIGURE 28. Interruption of other traces on the pavement.

distance is less than 20 cm from the marked real value points set to detected rasterized points is taken as the number of matching points. If the marked lane-mark is double yellow, the above distance threshold is set to 37.5 cm. Figure 23 (a) shows the real values that are marked manually. The results of lane marks extracted by the algorithm in this paper are indicated by red lines in Figure 23 (b). The non-lane part is painted in black. Figure 23 (c) is the image of the BEV space obtained by the IPM.

The extraction results are compared with the pre-marked real value. The score S is used to evaluate the effect of lane-mark extraction and is calculated as follows:

$$S = \frac{TP}{TP + FP + FN}, \tag{19}$$

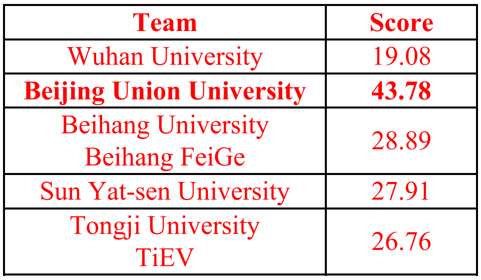
where the input discrete point is regarded as the positive checkpoint in the true value area; TP is the total length of the positive lane-mark; FN is the total length of the missed check, with value equal to the total marked true value minus TP; FN is the total length of false checks, with value equal to all extracted lane-mark points minus TP.

C. PERFORMANCE ANALYSIS

In the actual driving process of an autonomous vehicle, the phenomenon of false extraction and missed extraction often appears due to the complexity and variety of the driving environment. Our proposed lane-mark extraction algorithm can better shield these effects of uncertain factors on lane-mark extraction results in a complex and variable driving environment. Figure 25 $\sim$ Figure 28 show the experiment results for four different scenarios driving scenarios. The first column of each figure shows the original images collected by the onboard camera. The second column shows results after preprocessing. The third column shows the features of lane marks filtered by lane feature filtering based on multi-constraints. The fourth column shows clustering results by the clustering algorithm proposed in the fifth part. The fifth column shows the final result.

The first subfigure of Figure 25 contains lane-marks, zebra crossings and indicator arrows, and the illumination conditions are not ideal. There are speed limit marks in the second subfigure, and road speed limit messages in the third subfigure. Specifically, the lane-marks are submerged in the yellow

TABLE 5. Lane-mark detection test scores.



Team	Score
Wuhan University	19.08
Beijing Union University	**43.78**
Beihang University Beihang FeiGe	28.89
Sun Yat-sen University	27.91
Tongji University TiEV	26.76

deceleration zone, but our algorithm can still determine the position of lane-marks clearly.

Pedestrians block the right lane-mark by nearly 50% in the first subfigure of Figure 26. In the second subfigure, parking on the right side of the road obscures most of the lane-mark. A moving vehicle blocks part of the lane-mark in the third subfigure. In the fourth subfigure, when we do not operate the wipers on rainy days, raindrops slide down the windshield and block the target lane-mark. However, our algorithm can still extract the lane-mark robustly. The residual shadows formed by the movement of the wipers partially shields the partial lane-mark in fifth the subfigure, while the cones placed on the road hide the lane-mark in the sixth subfigure.

The first subfigure of Figure 27 is captured at 5∼6 p.m. The second is captured when the vehicle is driving at night. In the third subfigure, the light is strong and the target lane-mark is shiny. There are light spots on the windshield when driving in reverse light in the fourth and fifth subfigures. In the sixth subfigure, exposure conditions are not ideal when the vehicle drives out of the bridge hole. The opposing vehicle turns on its headlights due to foggy weather in the seventh subfigure. Our algorithm has strong robustness to driving scenes under abnormal light.

The lane-marks are blurred due to the interference of lime and mud in the first subfigure of Figure 28. There are repairs and cracks on the pavement in the second subfigure, and mud ruts in the third subfigure. There are shadows caused by trees on the roadside in the fourth subfigure, and lane marks are submerged in water in the fifth subfigure. Furthermore, there is water in the low-lying area of the roadside in the sixth subfigure, and ruts in snow in the seventh subfigure. However, our method can still extract the lane-marks accurately.

The results of the proposed algorithm are evaluated using the partial data sets from TSD-MAX. The results of evaluation are shown in Table 4.

The Traffic Scene Cognitive Basis Ability Offline Test of Intelligent Vehicle Future Challenge 2016 was held in China's Changshu City, Jiangsu Province, in November 2016. This offline test is funded by the Major Research Plan of the National Natural Science Foundation of China "Cognitive Computing of Visual and Auditory Information." It consists of three parts including traffic signal detection, front vehicle detection and lane-mark detection. There were five teams participating in the test of lane-mark detection. The test scores are shown in the table below.

VIII. CONCLUSIONS

In this paper, a lane-mark extraction algorithm based on autonomous driving with onboard vision is proposed to deal with complex real conditions. The four major parts of the proposed method are setting the preprocessing, lane-feature filtering based on multi-constraints, lane-feature clustering and fitting and verification and optimization of candidate lane. The proposed algorithm is evaluated for complex real conditions captured in an autonomous vehicle test, such as variable illumination conditions in different weather, occlusions such as pedestrians or other vehicles on the road, and other interference grown on the road caused by human factors intentionally or not.

The method is not fully applicable in extreme conditions, such as when the lane-mark is completely submerged by water, covered by snow, or obscured by a vehicle. Our proposed algorithm can be improved with information from multiple vehicular sensors, which we can combine according to some optimization criteria, enabling us to produce a consistent interpretation and description of the observed environment. Furthermore, the proposed algorithm could be improved by parallel implementation because several designed functions are independent.

REFERENCES

[1] G. N. Bifulco, L. Pariota, M. Brackstione, and M. Mcdonald, "Driving behaviour models enabling the simulation of advanced driving assistance systems: Revisiting the action point paradigm," *Transp. Res. C, Emerg. Technol.*, vol. 36, pp. 352–366, Nov. 2013.

[2] 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, p. 2, 2013.

[3] K.-Y. Chiu and S.-F. Lin, "Lane detection using color-based segmentation," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2005, pp. 706–711.

[4] C. D'Cruz and J. J. Zou, "Lane detection for driver assistance and intelligent vehicle applications," in *Proc. Int. Symp. Commun. Inf. Technol.*, Oct. 2007, pp. 1291–1296.

[5] P. Chanawangsa and C. W. Chen, "A new color-based lane detection via Gaussian radial basis function networks," in *Proc. Int. Conf. Connected Vehicles Expo (ICCVE)*, Dec. 2012, pp. 166–171.

[6] C. Jang *et al.*, "Curved lane detection for embedded system with adaptive threshold value based on Gaussian mixture model," in *Proc. Korea Auton. Soc. Autumn Conf. Exhib.*, 2015, p. 661.

[7] Y. Yang and H. E. Ying, "Lane detection and identification algorithm based on RGB space," *Comput. Modernization*, no. 2, pp. 86–90, 2014.

[8] C. Ma and M. Xie, "A method for lane detection based on color clustering," in *Proc. 3rd Int. Conf. Knowl. Discovery Data Mining*, Jan. 2010, pp. 200–203.

[9] T. Ohashi, Z. Aghbari, and A. Makinouchi, "Hill-climbing algorithm for efficient color-based image segmentation," in *Proc. IASTED Int. Conf. Signal Process., Pattern Recognit., Appl. (SPPRA)*, 2003, pp. 17–22.

[10] K. Kluge, "Extracting road curvature and orientation from image edge points without perceptual grouping into features," in *Proc. Intell. Vehicles Symp.*, Oct. 1994, pp. 109–114.

[11] A. H. S. Lai and N. H. C. Yung, "Lane detection by orientation and length discrimination," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 4, pp. 539–548, Aug. 2000.

[12] Z. Wennan, C. Qiang, and W. Hong, "Lane detection in some complex conditions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 117–122.

[13] H. Shen, S. Li, F. Bo, X. Miao, F. Li, and W. Lus, "Intelligent vehicles oriented lane detection approach under bad road scene," in *Proc. 9th IEEE Int. Conf. Comput. Inf. Technol.*, Oct. 2009, pp. 177–182.

[14] P. Lindner, S. Blokzyl, G. Wanielik, and U. Scheunert, "Applying multi level processing for robust geometric lane feature extraction," in *Proc. IEEE Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, 2010, pp. 248–254.

[15] C.-K. Cheong, "Design of lane detection system based on color classification and edge clustering," in *Proc. IEEE Quality Electron. Design*, Jul. 2011, pp. 266–271.

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

[17] W. Liu, H. Zhang, B. Duan, H. Yuan, and H. Zhao, "Vision-based realtime lane marking detection and tracking," in *Proc. Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2008, pp. 49–54.

[18] J.-H. Bae and J.-B. Song, "Monocular vision-based lane detection using segmented regions from edge information," in *Proc. Int. Conf. Ubiquitous Robots Ambient Intell.*, Nov. 2011, pp. 499–502.

[19] H. Li and F. Nashashibi, "Robust real-time lane detection based on lane mark segment features and general *a priori* knowledge," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2011, pp. 812–817.

[20] G. Pushen, Z. Bolun, and L. Haoran, "Lane detection of complex environment and navigation based on IPM," *China Modern Edu. Equip.*, no. 21, pp. 68–71, 2013.

[21] D. Ding, J. Yoo, J. Jung, and S. Kwon, "An urban lane detection method based on inverse perspective mapping," *Adv. Sci. Technol. Lett.*, vol. 63, pp. 53–58, 2015.

[22] L.-X. Peng and Y.-Q. Tao, "Analysis on lane detection and departure under IPM," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 8, no. 4, pp. 413–447, 2015.

[23] X. Q. Yan *et al.*, "Lane mark identification method based on edge distribution function: Lane mark identification method based on edge distribution function," *J. Comput. Appl.*, vol. 30, no. 4, pp. 974–976, 2010

[24] Z. Liu and Y. Chen, "A study on lane departure detection based on edge distribution function," *Automotive Eng.*, no. 7, pp. 626–629, 2007.

[25] C. Xiaoyan, *Research on the Real Time Lane Detection Based on Monocular Vision*. Nanjing, China: Univ. of Posts and Telecommunications, 2015.

[26] P. M. Daigavane and P. R. Bajaj, "Road lane detection with improved canny edges using ant colony optimization," in *Proc. Int. Conf. Emerg. Trends Eng. Technol.*, Nov. 2011, pp. 76–80.

[27] G. H. Chen *et al.*, "Lane detection based on improved canny detector and least square fitting," *Adv. Mater. Res.*, vols. 765–767, pp. 2383–2387, Sep. 2013.

[28] Y. Chai, S. J. Wei, and X. C. Li, "The multi-scale Hough transform lane detection method based on the algorithm of Otsu and canny," in *Proc. Adv. Mater. Res.*, vol. 1042. 2014, pp. 126–130.

[29] Z. Ronghui, W. Haiwei, Z. Xi, W. Lei, and J. Tonghai, "Lane detection algorithm at night based-on distribution feature of boundary dots for vehicle active safety," *Inf. Technol. J.*, vol. 11, no. 5, pp. 642–646, 2012.

[30] C.-H. Yu and C.-Y. Su, "An improved lane detection algorithm and the definition of the error rate standard," in *Proc. ICDIP*, 2012, p. 83341H.

[31] L. Hou, C. Fan, and D. I. Shuai, "Motorway lane detection method based on straight-line model," *J. Qingyuan Polytech.*, no. 6, pp. 73–76, 2011.

[32] M. Haloi and D. B. Jayagopi, "A robust lane detection and departure warning system," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 126–131.

[33] X. Gong, "Lane detection method based on recursive binary fitting," *J. Fiber Bioeng. Informat.*, vol. 8, no. 2, pp. 373–380, 2015.

[34] M. Foedisch, R. Madhavan, and C. Schlenoff, "Symbolic road perception-based autonomous driving in urban environments," in *Proc. 35th IEEE Appl. Imagery Pattern Recognit. Workshop*, Oct. 2006, p. 12.

[35] H. Lin, S. Ko, W. Shi, Y. Kim, and H. Kim, "Lane departure identification on highway with searching the region of interest on Hough space," in *Proc. Int. Conf0. Control, Autom. Syst.*, Oct. 2007, pp. 1088–1091.

[36] B.-F. Wu and C.-T. Lin, "Robust lane detection and tracking for driving assistance systems," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2007, pp. 3848–3853.

[37] J. Liu, Z. Li, H. Zhang, and C. Lv, "A vision-based road recognition algorithm," in *Proc. IEEE Conf. Ind. Electron. Appl.*, Jun. 2008, pp. 284–287.

[38] 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.

[39] H. Lin, S. Ko, H. Kim, and Y. Kim, "Road oundary detection with double filtering for intelligent vehicle," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2007, pp. 686–690.

[40] S. Srivastava, M. Lumb, and R. Singal, "Lane detection using median filter, Wiener filter and integrated Hough transform," *J. Autom. Control Eng.*, vol. 3, no. 3, pp. 258–264, 2015.

[41] Q. Lin, Y. Han, and H. Hahn, "Real-time lane departure detection based on extended edge-linking algorithm," in *Proc. Int. Conf. Comput. Res. Develop.*, May 2010, pp. 725–730.

[42] J. Yu, Y. Han, and H. Hahn, "An efficient extraction of on-road object and lane information using representation method," in *Proc. IEEE Int. Conf. Signal Image Technol. Internet Based Syst.*, Nov. 2008, pp. 327–332.

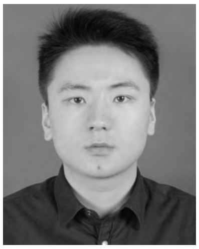
[43] P. C. Wu, C. Y. Chang, and C. H. Lin, "Lane-mark extraction for automobiles under complex conditions," *Pattern Recognit.*, vol. 47, no. 8, pp. 2756–2767, Aug. 2014.

[44] X. Wang, Y. Wang, and C. Wen, "Robust lane detection based on gradient-pairs constraint," in *Proc. IEEE Control Conf.*, Jul. 2011, pp. 3181–3185.

[45] W. Dandan, *Research and Implementation of Lane Recognition and Recognition Technology Based on Morphology Feature*. Shenyang, China: Northeastern Univ., 2012.

[46] *Road Traffic Signs and Markings—Part 3: Road Traffic Markings*. Beijing, China: Standardization Admin. China, 2009.

[47] S.-Y. Guo, W.-J. Zhai, Q. Tang, and Y.-J. Zhu, "Combining the Hough transform and an improved least squares method for line detection," *Comput. Sci.*, vol. 39, no. 4, pp. 196–200, 2012.



HANYU XUAN born in 1993. He received the bachelor's degree in computer science and technology from Chizhou College, Chizhou, China, in 2015, and the master's degree in software engineering from the Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, China. His research interests include artificial intelligence and image processing.

HONGZHE LIU born in 1971. She received the Ph.D. degree from the School of Computer and Information Technology, Beijing Jiao tong University, Beijing, China, and the M.S. degree in computer science from California State University, USA, in 1999. She is currently an Assistant Professor with the Computer Science Department, Beijing Union University, Beijing, where she is the Vice Director of the Beijing Key Laboratory of Information Service Engineering. Her research interests include semantic computing, artificial intelligence, and distributed systems.

JIAZHENG YUAN born in 1971. He received the Ph.D. degree from the Computer Science Department, Beijing Jiaotong University, in 2007. He is currently a Professor of software engineering with Beijing Union University, Beijing, China. His current research interests include graph and image processing, machine learning, and artificial intelligence.

QING LI born in 1983. She received the Ph.D. degree from the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China, in 2014. She is currently with Beijing Union University. Her research interests are computer vision and visual computing.