SPECIAL SECTION ON EMERGING TECHNOLOGIES
ON VEHICLE TO EVERYTHING (V2X)

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 12, 2018, accepted November 29, 2018, date of publication December 27, 2018,
date of current version March 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2018.2885568

An All-Weather Lane Detection System Based on Simulation Interaction Platform

NAN MA[1], GUILIN PANG[1], XIAOJUN SHI[1], AND YUN ZHAI[2]
[1]College of Robotics, Beijing Union University, Beijing 100101, China
[2]E-Government Research Center, Chinese Academy of Governance, Beijing 100089, China

Corresponding author: Nan Ma (xxtmanan@buu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61871038 and Grant 61672178 and in part by the Beijing Natural Science Foundation under Grant 4182022.

ABSTRACT Computer vision is an important part of the autonomous vehicles to gain the perception of the surrounding environment. In order to satisfy the adaptability of lane detection system under different illumination and different road conditions, an effective lane detection method based on image classification and hybrid isomeric operators is proposed. Camera correction matrix and distortion coefficient are obtained by using checkerboard grid images, and they will be cached to be shared with subsequent image streams to improve the speed of real-time lane detection. In the process of preprocessing, the images under different illumination conditions are classified and processed, and the inverse perspective transformation is used to deal with the 2-D image flow so that the image flow has a certain depth of data. According to the intensity of the whole gray mean, the edge detection is carried out by using heterogeneous operators and combining with a variety of threshold filtering methods to extract the lane pixels. The lane detection module transmits the vehicle steering angle and the deviate distance from the road center line to the decision layer, effectively implementing the interactive simulation. Experiments show that the algorithm has good real-time performance, stability, and robustness under different illumination and road conditions.

INDEX TERMS Autonomous vehicles, computer vision, human computer interaction, simulation, image edge detection.

I. INTRODUCTION

Autonomous Driving Vehicle is an important part for implementing the strategy of ≪Made in China 2025≫ and "the Notice of the State Council on the Issue of the New Generation of Artificial Intelligence Development Plan", ≪Made in China 2025≫ proposed to establish an intelligent network vehicle Independent research development system and production supporting system by 2020. Autonomous driving technology of autonomous driving vehicle has become one of the research hotspots in the field of artificial intelligence [1], [2]. Lane detection technology is highly important for autonomous driving vehicle to perceive the surrounding environment [3]. In order to satisfy the actual requirements, lane detection algorithm must ensure good reliability, real-time and robustness. With the development of autonomous driving technology, we need a lot of practical road tests. At the same time, a large amount of resources will be consumed and there is a certain danger. Therefore, we carried out related tests on the virtual simulation interaction platform provided by AUTOMOTIVE ARTIFICAL INTELLIGENCE(AAI), Germany. In the virtual test group competition of the second World Intelligent Driving Challenge in 2018, we found that different illumination conditions and road conditions are the biggest obstacles to lane detection. Therefore, designing a system which can detect lane stably all-weather will effectively enhance the robustness of lane detection. In this system, the accuracy and real-time identification of lane detection in different illumination conditions and different road conditions are the most important.

At present, extensive research efforts have been concentrated on a variety of lane detection and object recognition algorithms based on machine vision [4]. In order to improve the real-time performance of the algorithm, Liu et al. [5] used LSD algorithm [6] to detect lane lines, but did not consider the correlation of line segments, so it is easy to be interfered by uncorrelated line segments in sensitive noise environment. Chen *et al.* [7] use the scan line to extract the midpoint of the line element and the Hough transform to extract the lane

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

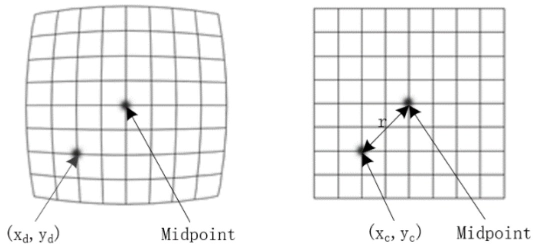

FIGURE 1. Camera distortion correlation parameters.

line. This method can meet the real-time requirement, but the robustness needs to be improved under the conditions of illumination change and vehicle interference.

It is difficult to satisfy the real-time and robustness of lane detection when the road image becomes complex due to the change of illumination and lane damage [8]. A new and all-weather lane detection algorithm based on simulation interaction platform is proposed in this paper. The actual test shows that the system can detect lane in real time and reliably under different illumination conditions and different road conditions. Although there has been much work on data classification in recent years [9], it is still a challenging task.

II. CAMERA CALIBRATION

The image flow collected by monocular camera lens has a certain degree of distortion, resulting in image distortion. We use $D = (k_1, k_2, p_1, p_2, k_3)$ to represent five distortion parameters [10], and the correction formulas for radial distortion is as below:

$$x_c = x_d(1 + k_1r^2 + k_2r^4 + k_3r^6) \tag{1}$$

$$y_c = y_d(1 + k_1r^2 + k_2r^4 + k_3r^6) \tag{2}$$

The correction formulas for tangential distortion is as below:

$$x_c = x_d + [2p_1x_dy_d + p_2(r^2 + 2x_d^2)] \tag{3}$$

$$y_c = y_d + [p_1(r^2 + 2x_d^2) + 2p_2x_dy_d] \tag{4}$$

In the above formulas:

- x_c and y_c represent coordinates after correction.
- x_d and y_d represent distorted coordinates
- k_1, k_2, k_3 represent radial distortion parameters
- r represents the distance from the corrected coordinates to the center of the picture
- p_1, p_2 represents tangential distortion parameters

Using the cv2.findChessboardCorners() function which in OpenCV to find out the pixel position of the diagonal (the intersection of black and white checkerboards) in the image of the chessboard; The relative positions of these diagonal points in the real world are constructed, and these relative positions are simplified into integral values [11]. For example, the first point of the third row is [0, 2, 0].

After obtaining the pixel location and actual relative position of the chessboard diagonal, then record them. According to the pixel position and actual relative position, the camera distortion parameters are calculated and cached by using the cv2.calibrateCamera() function in OpenCV. Correcting the image stream which are captured by the camera will take advantage of this distortion factor. In Figure 3, (a) is before the original image being corrected, and (b) is after the original image being corrected. After comparison, the image taken by the camera after the correction is closer to the real situation. The distortion caused by the ‘‘twisted line’’ is also corrected.

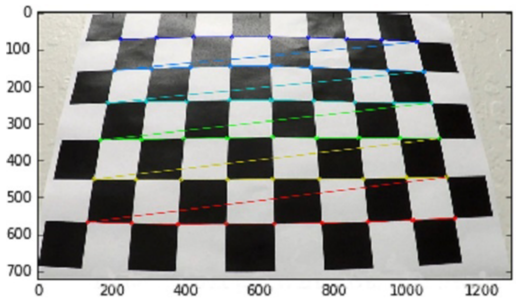

FIGURE 2. Chessboard diagonal in chessboard image.

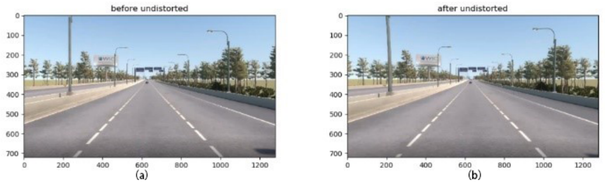

FIGURE 3. Comparison of original and corrected images.

III. TWO METHODS OF IMAGE VALUE COVERSION

The binarization of the image is to set the gray level of the points on the image to 0 or 255, that is to say, the whole image presents a clear black-and-white effect [12]. The binary image which can still reflect the whole and local features of the image can be obtained from the gray image with 256 brightness levels through appropriate threshold selection. In digital image processing, binary image occupies a very important position [13]. Especially in practical image processing, many of the current systems are made up of binary image processing. To process and analyze the binary image, the first thing is to obtain the binary image by binarizing the gray image. In this way, when the image is further processed, the set property of the image is only related to the position of the point whose pixel value is 0 or 255, and the multi-level value of the pixel is no longer involved, so that the processing becomes simple, and the data processing and compression is small.

A. SOBEL OPERATOR

The Sobel operator is a first-order differential operator. It calculates the gradient of one pixel by using the gradient value of the adjacent area of the pixel, and then chooses according to certain absolute value [14]. The essence is to reflect the gray difference characteristics of adjacent pixels. Not only the computation is small, but the algorithm is simple. The edge enhancement effect and noise suppression ability are better, but the edge of extraction is relatively coarse;

The operator consists of a set of transverse and a set of longitudinal 3∗3 matrices, which are convoluted with the image in plane, and then the approximate values of transverse and longitudinal luminance difference can be obtained respectively. Its form is filter operator, used to extract the edge, can use fast convolution function, simple and effective, so it is widely used. Unfortunately, the Sobel operator does not strictly distinguish the main body from the background of the image. In other words, the Sobel operator is not processed based on image grayscale. Because Sobel operator does not strictly simulate human visual physiological characteristics, the extracted image contours are sometimes unsatisfactory.

B. CANNY OPERATOR

The edge extracted by Canny operator is relatively complete, and the lane detected by Canny operator is single pixel wide, which has good signal-to-noise ratio and detection accuracy [15]–[17]. Canny algorithm uses Gaussian filter to smooth the image, suppress the noise in the image and prevent some unexpected individual prominent positions from being mistakenly detected as edges, affecting the subsequent Lane extraction; uses the first derivative finite difference to calculate the magnitude and direction of the gradient [18].

IV. ALL WEATHER LANE DETECTION METHOD BASED ON HYBRID ISOMERS

A. IMAGE STREAM DATA PREPROCESSING

1) IMAGE CLASSIFICATION

The global gray mean value of image based on global statistical features has strong ability to filter the local image noise error, so the local image noise caused by the local image illumination changes has little influence on it [19]. In this system, the mathematic characteristic standard of the image under different illumination conditions is the global gray mean value of image. If the gray value of the image is distributed in 1-K level, the average gray value of the whole image is as below:

$$u_r = \sum\nolimits_{i=1}^{K} ip(i), \quad p(i) = \frac{m_i}{M}, \quad M = \sum\nolimits_{i=1}^{K} m_i \tag{5}$$

Among them, m_i is the total pixel number of a certain gray value in the image; M is the total pixel number of the image; $p(i)$ is the probability of a certain gray value in the image gray value.

In order to verify the rationality of classifying images by using the global gray mean, a large number of image samples were collected under strong light, night, weak light and normal light conditions. The experimental statistics show that the overall gray mean of the image can well reflect the intensity of the image. As shown in Fig. 4 (a)-(f), the mean gray mean value increases with illumination enhancement.

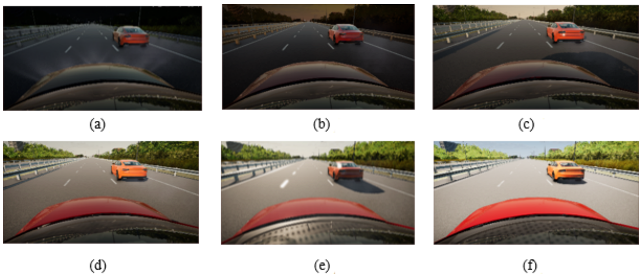

FIGURE 4. Different illumination images and global gray mean value. (a) $\mu = 65$. (b) $\mu = 97$. (c) $\mu = 135$. (d) $\mu = 173$. (d) $\mu = 204$. (f) $\mu = 219$.

2) WEAK LIGHT LEVEL IMAGE STRETCHING

The purpose of gray scale stretching is to enhance the contrast of each part of the original image, highlight the gray areas of interest in the image, and suppress the gray areas of interest [20]. Its essence is to enhance image contrast and maximize the gray difference between lane and road surface. The grayscale stretching transformation often uses the three-segment linear transformation function [21]. The formula is as follows:

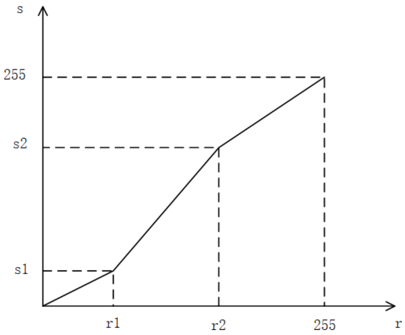

FIGURE 5. Sketch map of piecewise linear gray transformation.

$$f(r) = \begin{cases} \frac{s_1}{r_1} & r0 \le r < r_1 \\ \frac{s_2 - s_1}{r_2 - r_1}(r - r_1) + s_1 & r_1 \le r \le r_2 \\ \frac{255 - s_2}{255 - r_2}(r - r_2) + s_2 & r_2 < r < 255 \end{cases} \tag{6}$$

r_1, r_2 give the gray range that needs to be transformed; s_1, s_2 parameters determine the slope of the current transformation, S1 is the gray level of the original image, S2 is the gray level of the converted image. This gray transformation function is controlled by adjusting the control point $(r_1, s_1)(r_2, s_2)$, it can extend or compress any gray interval of the original image. As shown in Figure 5, if the gray level of an image is concentrated in a darker area and the image is darker, the gray scale stretching function can be used to stretch (slope $>$ 1) the gray range of the object to improve the image. As shown in Fig. 6 (b), the gray level of the darker area of the image is stretched, and the difference between the lane and the road surface is increased, which is beneficial to the subsequent binary processing.

3) GRAY SCALE EQUALIZATION OF STRONG ILLUMINATION IMAGES

Fig. 7 (a)-(b) show grayscale histogram equalization, which changes the randomly distributed image grayscale histogram into a uniformly distributed histogram. By mapping and

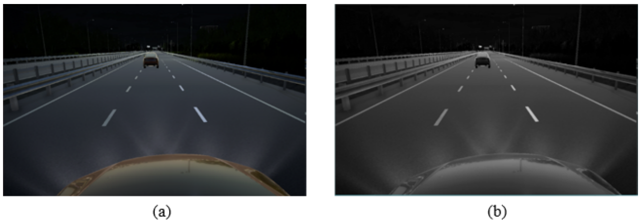

FIGURE 6. **Comparison of weak light images before and after gray stretching. (a) Before gray stretching. (b) after gray stretching.**

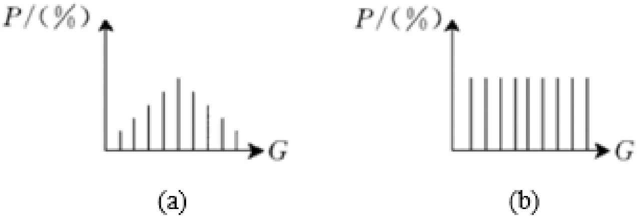

FIGURE 7. **Histogram equalization.**

transforming the pixel gray level of the original image, the probability density of the gray level of the transformed image is uniformly distributed [22]. This means that the dynamic range of the grayscale is increased and the contrast of the image is improved.

The gray level of a pixel in the original image is i, after grayscale equalization, the grayscale is $y = f(i)$, the mapping function must satisfy two conditions (the gray level is L =255):

1) $f(i)$ is a monotone increasing function in the range of $0 \leq i \leq L-1$. This is to ensure that grayscale equalization does not disrupt the order of the gray level of the original image. The grayscale level of the original image is still kept from black to white (or from white to black) after transformation.
2) For $0 \leq i \leq L-1$, there is $0 \leq y \leq L-1$, This condition guarantees the consistency of the dynamic range of gray values before and after transformation. The mapping function of gray equalization is defined as the following formula:

$$y = f(i) = S(i) \circ L/(H \circ W) \tag{7}$$

Among them, $S(i) = \sum_{i=0}^{i} N_i$, H is image height, W is image weight. According to the above formulas, each pixel in the original image is transformed to achieve gray level equalization. Fig. 8 (a) is the original gray image; Fig. 8 (b) is the gray histogram of Fig. 8 (a); Fig. 9 (a) is the result of image gray equalization; Fig. 9 (b) is the gray histogram of 9 (a), as shown in the figure:

B. IMAGE INVERSE PERSPECTIVE TRANSFORMATION

In image processing, it is necessary to set up a region of interest (ROI) [23]. Of course, different visual tasks have different ROI. For Lane detection, ROI is the lane area in front of the vehicle. The selection of ROI in the system is shown in Figure 10:

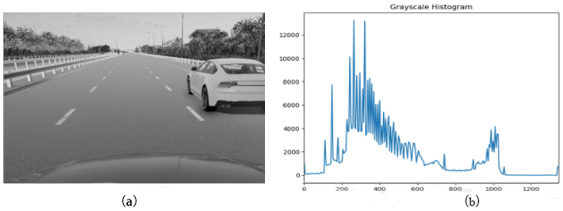

FIGURE 8. **Contrast map before gray equalization. (a) Original image. (b) Pixel detection image.**

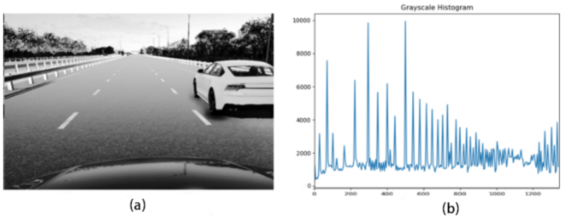

FIGURE 9. **Contrast map after gray equalization. (a) Original image. (b) Pixel detection image.**

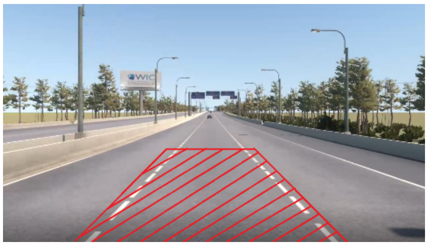

FIGURE 10. **Inverse perspective transformation ROI (region of interest).**

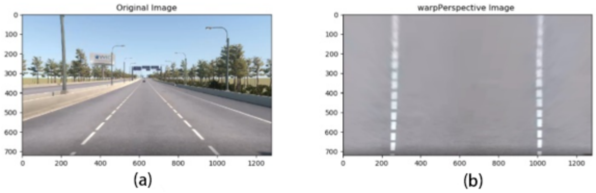

FIGURE 11. **Original image and reverse perspective transformation image.**

Inverse perspective transformation is the projection of images to a new visual plane, also known as projection mapping [24]. Because the view in the original image is not conducive to subsequent image processing, we can get the bird's eye view by inverse perspective transformation, and display ROI in the new view. Using two functions (cv2.getPerspective() Transform function and cv2.warpPerspective() function) in OpenCV to finish inverse perspective transformation. Figure 11 (a) is the original image; Figure 11 (b) is the result of inverse perspective transformation:

C. SOBEL, CANNY CONSTRUCTION OF HYBRID HETEROGENEOUS OPERATOR ALGORITHM

1) THE LANE DETECTION SYSTEM DESIGN ARCHITECTURE BASED ON S-C MIX ISOMERISM OPERATOR

The lane detection design architecture based on S-C hybrid heterogeneous operator can be divided into six parts: camera

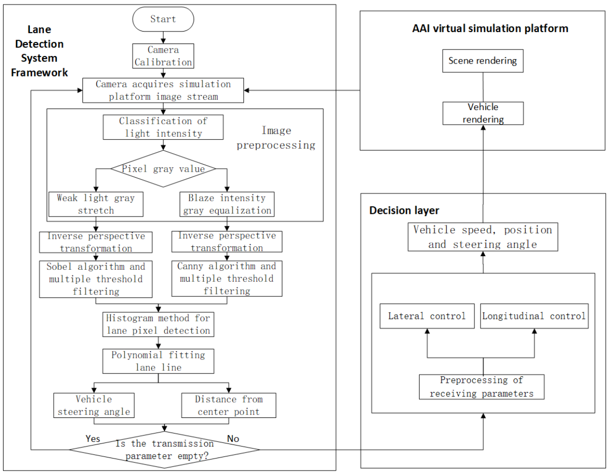
Lane Detection System Framework
Start
Camera Calibration
Camera acquires simulation platform image stream
Classification of light intensity
Image preprocessing
Pixel gray value
Weak light gray stretch
Blaze intensity gray equalization
Inverse perspective transformation
Inverse perspective transformation
Sobel algorithm and multiple threshold filtering
Canny algorithm and multiple threshold filtering
Histogram method for lane pixel detection
Polynomial fitting lane line
Vehicle steering angle
Distance from center point
Yes
Is the transmission parameter empty?
No
AAI virtual simulation platform
Scene rendering
Vehicle rendering
Decision layer
Vehicle speed, position and steering angle
Lateral control
Longitudinal control
Preprocessing of receiving parameters

FIGURE 12. System Flow Chart.

calibration, image stream data preprocessing, image inverse perspective transformation, binarization of images by S-C hybrid heterogeneous operator combined with RGB and HLS hybrid threshold filtering method, peak detection and fitting of lane lines by Histogram method, and related parameters calculation. The overall flow of the algorithm is shown in Figure 12:

First, the camera is calibrated and the road image is preprocessed, Then the pixel gray value is judged and classified, the high-light image is equalized, and the low-light image is stretched. After the preprocessing, the inverse perspective transformation is used to get the bird's eye view of the image, Then the binary image is processed by S-C hybrid heterogeneous operator and RGB threshold filtering algorithm. The peak value of the binary image is detected and the sliding window is fitted. On the basis of fitting lane line, the angle and deviation distance from the center line are calculated and transmitted to the decision layer. If the two values with large amplitude, they will not be used, the next frame of image is detected again.

2) S-C CONSTRUCTION OF MIXED HETEROGENEOUS OPERATORS

The efficiency of Sobel operator in practical application is high. Therefore, when the precision meets the application requirements, the Sobel operator is used to process the image whose gray mean is less than r_2 (Formula 6). The X-direction Sobel operator tends to detect vertical edges, while the y-direction Sobel operator tends to detect horizontal edges [25]. In lane detection problem, the lane lines we are concerned about are usually vertical. At the same time, we want to filter some horizontal lines, so in this algorithm, we use the x-direction Sobel operator.

If A represents the original image, G_x represents the horizontal edge detection of the image, and G_y represents the vertical edge detection of the image, the formula is as follows:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A,\ G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A \quad (8)$$

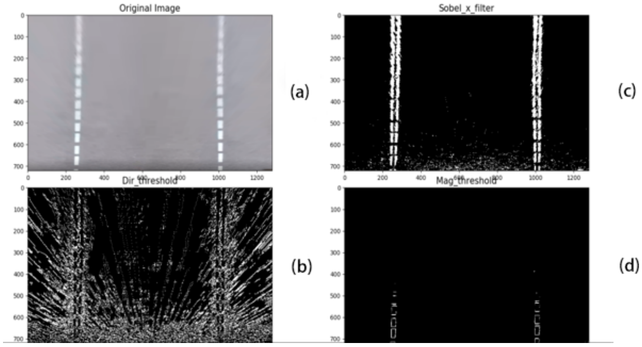

FIGURE 13. Sobel operator processing results.

By using the Sobel operator to convolute in both directions, we can get the approximate values of the horizontal and vertical gradient of each pixel of the image. Further, we can calculate the gradient by combining the two approximations:

$$G = \sqrt{G_x^2 + G_y^2} \quad (9)$$

The following formula can be used to calculate the gradient direction:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (10)$$

The original image is shown in Figure 13 (a); the Sobel operator processing results along the transverse gradient direction are shown in Figure 13 (b); the Sobel operator convolution results along the X direction are shown in Figure 13 (c); the Sobel operator processing results along the longitudinal gradient direction are shown in Figure 13 (d); and the Sobel operator processing results along the transverse convolution can be seen in Figure 13 (c). It got the best result. At the same time, the gradient direction also provides a reference for the selection of lane line pixels.

Sobel operator is difficult to satisfy the precision requirement of practical applications for the strong illumination image with the global gray mean $\geq r_2$ (formula 6). Therefore, the *Canny* operator is used to deal with strong illumination images.S_x represents edge detection of images along the horizontal direction. S_y represents the edge detection of images along the longitudinal direction. $S[i,j]$, $S[i+1,j]$, $S[i,j+1]$, $S[i+1,j+1]$ represent respectively four directions of gradient: horizontal, main diagonal, counter-diagonal and vertical, the formula is as follow:

$$S_x = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix},\quad S_y = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \quad (11)$$

$$P[i,j] \approx (S[i,j+1] - S[i,j] + S[i+1,j+1] - S[i+1,j])/2 \quad (12)$$

$$Q[i,j] \approx (S[i,j] - S[i+1,j] + S[i,j+1] - S[i+1,j+1])/2 \quad (13)$$

Among them, S is global gray value, P represents X directional gradient threshold. Q represents Y directional gradient threshold.

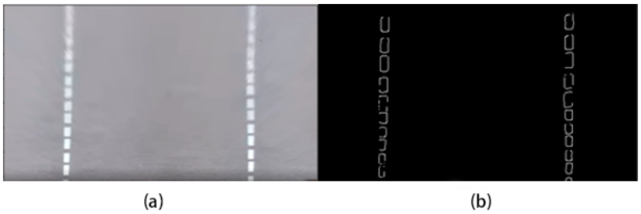

FIGURE 14. Comparison of original image and Canny operator processing effect.

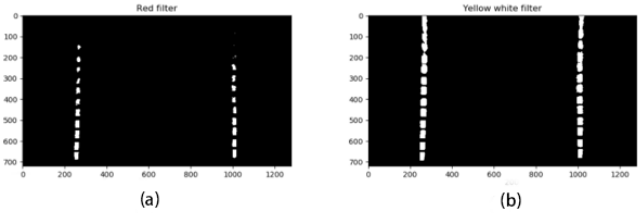

FIGURE 15. Comparison of thresholding effect of RGB.

The formula for calculating amplitude $M[i,j]$ and direction angle $\theta[i,j]$ is as follows:

$$M[i,j] = \sqrt{P[i,j]^2 + Q[i,j]^2} \tag{14}$$

$$\theta[i,j] = \arctan(Q[i,j]/P[i,j]) \tag{15}$$

The gradient amplitude was inhibited by non-maximal method. The local maximum gradient was retained while all other gradient values were inhibited. Only the sharpest position in the gradient variation is retained. The gradient intensity of the current point and the positive and negative gradient direction points is compared. Comparing the gradient intensity of the current point and the positive and negative gradient directions compared. Only if the gradient value of the current point is greater than the gradient strength of other points in the same direction will it be retained, otherwise discarded, it will be set to 0.

After non-maxima suppression, the output edges will contain a small number of non-edge pixels. Double threshold edge connection, set high and low two thresholds, if and only if there is a point of contact gradient is greater than the high threshold, and the gradient of all edge pixels are greater than the low threshold is retained, otherwise discarded.

The result of Canny operator convolution processing is as below:

The color feature of lane line is one of the basis of lane line detection. According to the National Road standard, lane line only has white and yellow, so we can filter these two colors in RGB color space to extract the pixels of lane line. Fig. 15 (a) Sets the appropriate threshold on Red channel to filter the yellow and white lanes; Fig. 15 (b) Sets the threshold directly on the three channels to obtain the optimal filtering effect by combining them. From the results, we can see that RGB color threshold can filter out the lane image well under normal light conditions. But it is not stable in complex environment light.

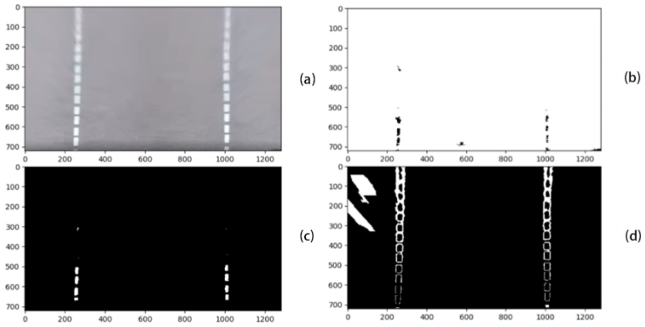

FIGURE 16. HLS threshold.

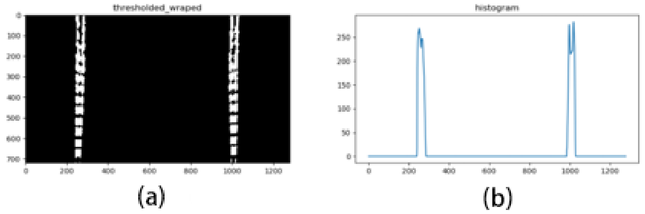

FIGURE 17. Peak detection results.

Therefore, we need to use the color threshold of HLS (hue, brightness, saturation) to detect the lane line under complex environmental light. We can directly use the cv2.cvtColor() method in OpenCV to convert images from RGB color space to HLS color space. Then the lane lines carry out thresholding in three channels of HLS color space. Figure 16 below is the renderings:

Fig. 16 (a) is the original image before RGB thresholding; Fig. 16 (b), (c), (d) is the result of RGB thresholding of H, S, L channels respectively; The result of HLS thresholding filtering shows that the combination of H and S channels has the best thresholding effect.

Obviously, a single threshold filtering method can't cope with complex and changeable road conditions. Therefore, we combine the above methods to extract the lane line features, considering the real-time and lane edge detection effect. In this system, Canny operator and color thresholding are used to enhance the edge of the strong illumination image, while Sobel operator and color thresholding are used to enhance the normal illumination image and the weak illumination image.

3) LANE PEAK DETECTION

The binary image is obtained by S-C mixed heterogeneous operator and a variety of threshold filtering methods. We can use the binary image to detect the lane pixels and get the coordinate values of the base points of the lane. Histogram is an algorithm based on pixel histogram [26]. Since the pixels of two lanes are concentrated in a certain range of the x-axis, the image is divided into two left and right, and the peak of pixel distribution on both sides of the x-axis is the base point of the lane. As shown in Figure 17:

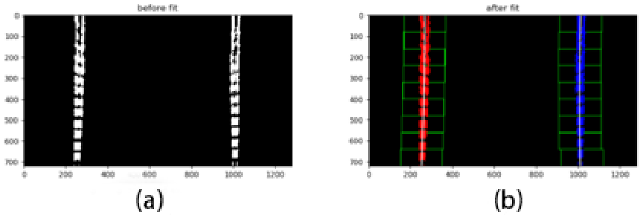

FIGURE 18. Sliding window fitting lane line.

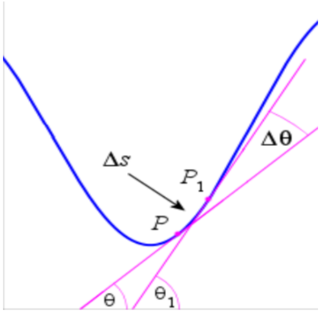

FIGURE 19. Schematic diagram of vehicle steering angle calculation.

As shown in Figure 17 (a), the lane lines on the left and right correspond to the peak values on the left and right of the pixel histogram of the lane line in Figure 17 (b). Then run the sliding window [27] and record the coordinates of pixels which are not 0. The window is scanned from the bottom to record the coordinates of pixels. If the number of pixels is greater than a certain threshold, the mean value of these pixels is used as the center of the next sliding window. Fig. 18 (a) shows the mixed heterogeneous operator and the graph after threshold filtering, (b) shows the sliding window fitting Lane visual interface. Fitting a quadratic polynomial with the pixels of the recording coordinates, the curve is fitted with the pixels of the past five consecutive frames.

The polynomial formula is as follows:

$$x = ay^2 + by + c \tag{16}$$

In this system, curvature is calculated by the following methods:

$$R = \frac{(1 + (2ay + b)^2)^{\frac{3}{2}}}{|2a|} \tag{17}$$

Among them, a is the quadratic term coefficient of a polynomial, b is the primary term coefficient of polynomials;

Calculate the distance from the lane center line by the following methods:

$$D = P_{cen} - P_{veh} \tag{18}$$

Among them, D represents the distance from center, P_{cen} is the middle point of image, P_{veh} is the lane center.

Vehicle steering angle is calculated by fitting the slope of curve generated by polynomial of lane pixels. In figure 19, P, P_1 represents the two points on the curve, respectively, representing the angle calculated from the slope of the current point. Θ X represents the angle calculated from the current point slope. The formula is as follows:

$$\theta = \arctan(\frac{d_x}{d_y}) \tag{19}$$

$$\Delta\theta = \theta_1 - \theta_2 \tag{20}$$

V. REAL-TIME DETECTION AND VERIFICATION OF ALL-WEATHER LANE BASED ON VIRTUAL SIMULATION PLATFORM

A. INTRODUCTION OF SYSTEM TEST PLATFORM

The algorithm proposed in this paper adopts lane detection algorithm based on monocular vision. The monocular camera (model: GS3-U3-41C6C-C) and 35mm zoom lens are placed in front of the screen of the simulation platform. The height is 1.10m, and the resolution of the captured image stream is 1920x1080. which is the initial filtering of the interference information in the image stream. In order to filter the disturbance information in the image stream, the resolution of the image stream captured by the monocular camera is processed to 1280x720. The configuration of the in-depth learning workstation is as follows: Ubuntu 16.04 system, Intel Core i7 CPU, NVIDIA GTX 1080Ti graphics card, 32GB memory, 2T 7200 RPM enterprise mechanical hard disk. The software development environment of this system is PyCharm and CLion. LCM transmits the vehicle steering angle and the distance from the center of the lane to the decision-making layer in the lane detection module. The German company AAI provides servers. The real time speed, steering angle and vehicle position of the decision layer are transmitted to the server by Gigabit cable. After receiving the parameters, the AAI simulator can render the vehicle and the virtual scene in real time. The operation cycle of lane detection system based on simulation interaction platform is 50ms.

B. ANALYSIS OF TEST RESULTS

In the virtual test group competition of the World Intelligent Driving Challenge in 2018, we tested the stability, real-time performance and robustness of the all-weather lane recognition system based on the simulation interactive platform to deal with different illumination and different traffic image streams. As shown in Figure 20, we collected the AAI simulator image stream through the monocular camera.

In order to ensure the rationality of lane line evaluation, the area which is useless for vehicle control is removed because of the close distance from the vehicle. It also eliminates areas that are likely to cause greater errors because they are too far away from vehicles. Only the lane detection results in the 10-50m Road area in front of the current vehicle are scored and counted. The visible area in the image stream is 0.1H-0.5H. The pixel-based grid scoring statistical area is used to discretize the position information of each lane into a set of sequential points.

$$P_{TP} = \frac{TP}{TP + FP + MP} \times 100\%(21) \tag{21}$$

TABLE 1. A statistical table of the results of the detection of heterogeneous operators under different general gray mean values and comparison of test results of other algorithms under the same conditions.

	S-C hybrid heterogeneous algorithm (Ours)				Spatial CNN (Pan et al.2017) [28]				Point set optimization lane recognition method based on segment switching model (Rong et al.2017) [29]			
Gray mean μ	Frames	P_{TP}/%	P_{FP}/%	P_{MP}/%	Frames	P_{TP}/%	P_{FP}/%	P_{MP}/%	Frames	P_{TP}/%	P_{FP}/%	P_{MP}/%
0< μ <=51	2523	92.53	5.57	1.90	2523	90.09	4.05	5.86	2523	88.07	6.64	5.29
51< μ <=102	9835	96.16	1.12	2.72	9835	96.05	1.37	2.58	9835	91.37	4.15	4.48
102< μ <=153	9954	99.51	0.94	0.55	9979	98.36	0.65	0.99	9954	95.04	3.08	1.88
153< μ <=204	10157	97.44	1.98	0.58	10163	95.07	3.27	1.66	10157	90.62	5.16	4.22
204< μ <=255	2876	95.10	3.37	1.63	2876	91.59	4.51	3.90	2876	89.32	6.52	4.16

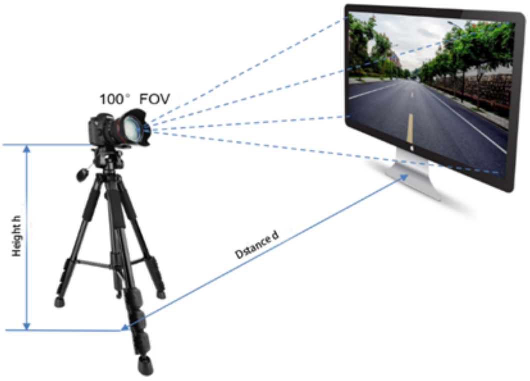

FIGURE 20. Camera capturing image stream intent.

Among them, *TP* is the total positive rate of all lane line images detected in the frame image. *FP* is total false positive rate. *MP* is total missing rate. Because the lane line in reality has a certain width error, it is allowed to detect the position of the lane line and the real value of the lane line marked with a deviation of 5 pixels.

From the data of S-C mixed heterogeneous operators in Table 1, it can be seen that the system has good robustness and stability in dealing with extreme cases ($0 < \mu <= 51$, $204 < \mu <= 255$) of the global gray mean distribution of some images caused by various complex climatic factors (cloudy, foggy, snow, water, etc.). For the image whose gray-level mean distribution is $102 < \mu <= 204$, the positive detection rate is as high as 99.51%, and the error detection rate is the lowest. Comparing the TP data of different algorithms in Table 1, we can see that the hybrid heterogeneous algorithm has better accuracy and robustness than the Spatial CNN and the point set optimization lane recognition method based on the segment switching model in lane detection under different image gray mean conditions. Figure 21-23 shows the visual result graphs of processing at noon, dusk and night.

Shadows and disturbances caused by overpasses, roadway trees, road markings and tall buildings across the driveway during actual driving. And heavy weather conditions such as heavy rain, snowfall and sand dust will cause interference to lane detection. The above bad road conditions and weather conditions are easy to increase the false detection rate. As shown in Figure 24-28, we tested road fouling, vehicle occlusion, road shadow, rainy day road and road marking jamming. According to the data in Table 2, the system has strong robustness and stability after a large number of tests.

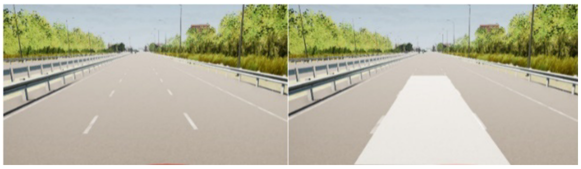

FIGURE 21. Noon image and processing result graph.

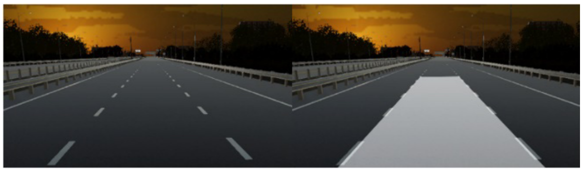

FIGURE 22. Dusk image and processing result graph.

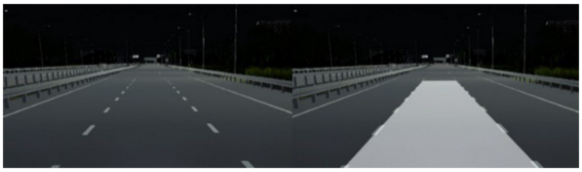

FIGURE 23. Night image and processing result graph.

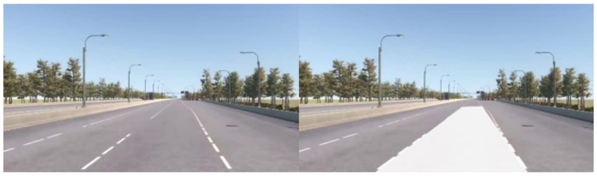

FIGURE 24. Road fouling and processing result graph.

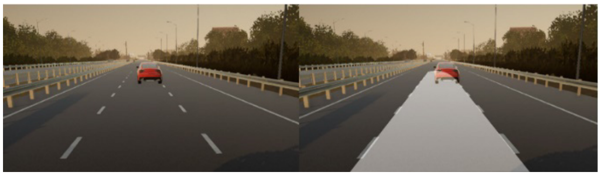

FIGURE 25. Vehicle occlusion and processing result graph.

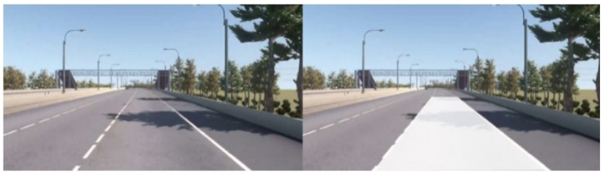

FIGURE 26. Road shadow and processing result graph.

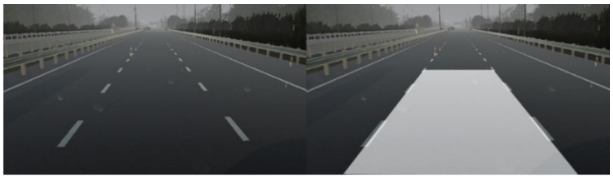

FIGURE 27. Rainy day road condition and processing result graph.

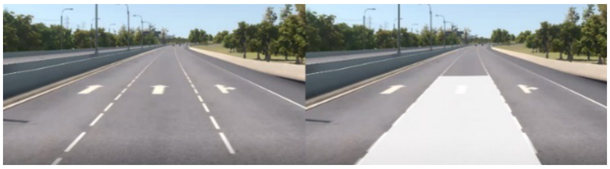

FIGURE 28. Road marking jamming and processing result graph.

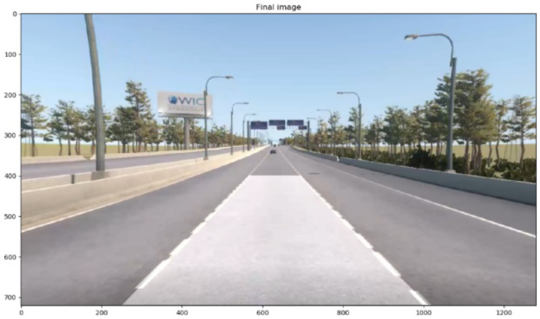

FIGURE 29. Visual inspection result diagram.

TABLE 2. Test results under different road conditions.

Main influencing factors	Frame number	Average $P_{TP}/\%$	Average $P_{FP}/\%$	Average $P_{MP}/\%$
Road fouling	2201	92.90	1.57	4.53
Vehicle occlusion	5835	93.72	3.52	2.76
Road shadow	9954	92.51	3.94	3.55
Rainy day road	4157	92.44	3.98	3.58
Road marking jamming	3876	93.10	2.27	4.63

In the 2nd World Intelligent Driving Challenge, the system can accurately and real-time detect the lanes in different illumination and different road image streams. In the existing hardware environment, the system processing speed is 20 frames per second on average, the excellent cooperation decision layer is 50ms, and the accuracy of correct detection of complex road conditions and complex weather environment image streams is 97.12%. Therefore, we won the first prize in the virtual test group of the second World Intelligent Driving Tournament. Figure 29 is a visual result graph of lane detection.

VI. CONCLUDING REMARKS

This system presents an all-weather lane detection system based on simulation interactive platform. It has good real-time, robustness and stability for lane detection under different illumination and different road conditions. The camera is calibrated to get the distortion parameters; the images with different illumination intensity are pre-processed; the lane features are extracted and filtered from the images with different illumination conditions. The lane line is fitted by polynomial. Finally, the polynomial fitting curve calculates the lane deviation angle and the vehicle deviation from the center of the lane and other parameters, and passes them to the decision layer for processing. The decision layer is real-time decision-making, and the parameters such as steering angle, speed and vehicle real-time location are transmitted to the AAI simulator. Won the first prize in the second world Intelligent Driving Challenge (first prize) shown that: The algorithm proposed in this paper not only improves the accuracy of lane detection, but also meets the requirements of all-weather, real-time and robustness, and has strong adaptability to different road conditions and weather conditions. However, the system does not achieve multi-lane detection and prediction of lane extension curvature, which will be our future research direction to improve.

REFERENCES

[1] Y. Ren, F. Liu, Z. Liu, C. Wang, and Y. Ji, "Power control in D2D-based vehicular communication networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5547–5562, Dec. 2015.

[2] C. Wu, Z. Liu, T. Yoshinaga, Y. Ji, and D. Zhang, "Spatial intelligence toward trustworthy vehicular IoT," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 22–27, Oct. 2018.

[3] N. Ma, Y. Gao, and J. Li, "Interactive cognition in self-driving," *China Sci., Inf. Sci.*, vol. 48, no. 8, pp. 1083–1096, Aug. 2018.

[4] X. Zhao, N. Wang, S. Du, Y. Gao, J. Sun, and Y. Zhang, "Beyond pairwise matching: Person reidentification via high-order relevance learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3701–3714, Aug. 2018.

[5] W. Liu, S. Li, and X. Huang, "Extraction of lane markings using orientation and vanishing point constraints in structured road scenes," *Int. J. Comput. Math.*, vol. 91, no. 11, pp. 2359–2373, 2014.

[6] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A line segment detector," *IPOL J.*, vol. 2, pp. 35–55, Mar. 2012.

[7] L. Chen, Q. Li, and Q. Mao, "Lane line detection and tracking method based on imaging model," *China J. Highway Transp.*, vol. 24, no. 6, pp. 96–102, 2011.

[8] C. Lee and J.-H. Moon, "Robust lane detection and tracking for real-time applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 4043–4048, Dec. 2018.

[9] Y. Gao, H. Zhang, X. Zhao, and S. Yan, "Event classification in microblogs via social tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 3, 2017, Art. no. 35.

[10] T. Hanning, "A least squares solution for camera distortion parameters," *J. Math. Imag. Vis.*, vol. 45, no. 2, pp. 138–147, Feb. 2013.

[11] S. Guo, Z. Li, and C. Zhao, "A lane detection method combining Hough transform and Kalman filter," *J. China Univ. Metrol.*, 2017.

[12] Y. Li, H. Huang, and X. Li, "Night lane detection based on Canny operator and Hough transform," *Sci. Technol. Eng.*, no. 31, pp. 234–237, 2016.

[13] J. Son, H. Yoo, and S. Kim, "Real-time illumination invariant lane detection for lane departure warning system," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1816–1824, Mar. 2015.

[14] B. Jiang, J. Liu, and K. Sun, "Refinement of sobel operator-edge detection," *J. Shenyang Normal Univ.*, 2010.

[15] L. Ding and A. Goshtasby, "On the Canny edge detector," *Pattern Recognit.*, vol. 34, no. 3, pp. 721–725, Mar. 2001.

[16] H. Hu, Y. Wen, and D. Niyato, "Public cloud storage-assisted mobile social video sharing: A supermodular game approach," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 545–556, Mar. 2017.

[17] H. Hu, Y. Wen, and D. Niyato, "Spectrum allocation and bitrate adjustment for mobile social video sharing: Potential game with online QoS learning approach," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 4, pp. 935–948, Apr. 2017.

[18] Q. Xu, S. Varadarajan, C. Chakrabarti, and L. J. Karam, "A Distributed canny edge detector: Algorithm and FPGA implementation," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 2944–2960, Jul. 2014.

[19] X. Zhou, X. Yang, and X. Yao, "The study of cloud classification and detection in remote sensing image," *J. Graph.*, vol. 35, no. 5, pp. 768–773, 2014.

[20] L. Heng-Yi, Y. Liu, and X. C. Xue, "Methods and experiments of background subtraction and grayscale stretch for remote sensing images," *Chin. J. Liquid Cryst. Displays.*, vol. 27, no. 2, pp. 235–239, 2012.

[21] M. Chen, C. Chen, and J. Gu, "An adaptive piecewise linear transformation method based on histogram," *Foreign Electron. Meas. Technol.*, no. 2, pp. 36–38, 2015.

[22] K. S. Sim, C. P. Tso, and Y. Y. Tan, "Recursive sub-image histogram equalization applied to gray scale images," *Pattern Recognit. Lett.*, vol. 28, no. 10, pp. 1209–1221, Jul. 2007.

[23] K. Tiwari and P. Gupta, "An efficient technique for automatic segmentation of fingerprint ROI from digital slap image," *Neurocomputing*, vol. 151, pp. 1163–1170, Mar. 2015.

[24] Q. Dai, Y. Wang, and G. Han, "Perspective image rectification based on improved hough transformation and perspective transformation," *Chin. J. Liquid Cryst. Displays.*, vol. 27, no. 4, pp. 552–556, 2012.

[25] X. Feng, W. Fang, and D. Yan, "Image edge detection based on the median filtering and Sobel,canny operator," *J. Heilongjiang Hydraulic Eng.*, 2009.

[26] Q. Li, N. Zheng, and H. Cheng, "Springrobot: A prototype autonomous vehicle and its algorithms for lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 300–308, Dec. 2004.

[27] P. Cataldi, M. Grangetto, T. Tillo, E. Magli, and G. Olmo, "Sliding-window raptor codes for efficient scalable wireless video broadcasting with unequal loss protection," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1491–1503, Jun. 2010.

[28] X. Pan, J. Shi, X. Wang, X. Tang, and P. Luo. (Dec. 2017). "Spatial as deep: Spatial CNN for traffic scene understanding." [Online]. Available: https://arxiv.org/abs/1712.06080

[29] H. Rong, M. L. Zhang, and X. J. Zhang, "Point set optimization lane line identification method based on the subsection switch model," *Sci. Technol. Eng.*, 2016.

NAN MA received the Ph.D. degree in computer application from the University of Science and Technology Beijing, Beijing, China, in 2013. She is currently an Associate Professor with the College of Robotics, Beijing Union University, Beijing. She has published more than 30 papers in international journals and conferences. Her current research interests cover interactive cognition, multimedia content analysis, and knowledge discovery.

GUILIN PANG is currently pursuing the degree in software engineering with Beijing Union University. He received the First Prize of the 2nd World Intelligent Driving Challenge Virtual Scene Group.

XIAOJUN SHI is currently pursuing the degree in software engineering with Beijing Union University. He received the Second Prize of the 2nd World Intelligent Driving Challenge Virtual Scene Group.

YUN ZHAI was born in 1979. He received the Ph.D. degree from the School of Computer and Communication Engineering, University of Science and Technology Beijing, in 2012. His current research interests include knowledge discovery and e-government intelligence.