sensors

Article

A Lane Detection Method Based on a Ridge Detector and Regional G-RANSAC

Zefeng Lu [†], Ying Xu [†,*], Xin Shan, Licai Liu, Xingzheng Wang and Jianhao Shen

College of Mechatronics and Control Engineering, Shenzhen University, Shenzhen 518060, China
* Correspondence: yxu@szu.edu.cn; Tel. +86-0755-2653-2036
† These authors contributed equally to this work.

Received: 6 August 2019; Accepted: 16 September 2019; Published: 18 September 2019

Abstract: Lane detection plays an important role in improving autopilot's safety. In this paper, a novel lane-division-lines detection method is proposed, which exhibits good performances in abnormal illumination and lane occlusion. It includes three major components: First, the captured image is converted to aerial view to make full use of parallel lanes' characteristics. Second, a ridge detector is proposed to extract each lane's feature points and remove noise points with an adaptable neural network (ANN). Last, the lane-division-lines are accurately fitted by an improved random sample consensus (RANSAC), termed the (regional) gaussian distribution random sample consensus (G-RANSAC). To test the performances of this novel lane detection method, we proposed a new index named the lane departure index (LDI) describing the departure degree between true lane and predicted lane. Experimental results verified the superior performances of the proposed method over others in different testing scenarios, respectively achieving 99.02%, 96.92%, 96.65% and 91.61% true-positive rates (TPR); and 66.16, 54.85, 55.98 and 52.61 LDIs in four different types of testing scenarios.

Keywords: lane division lines' detection; ridge detector; BP neural network; feature extraction; RANSAC

1. Introduction

Lane-division line detection plays a critical role in improving the safety level of intelligent electric vehicles (IEVs). Currently, there are two main methods to detect lane-division-lines: feature-based detection and model-based detection. Feature-based detection extracts the edge location distribution, connected shadow area, color and texture differences from graphs to detect lane-division-lines [1–3]. Zhang proposed a Hough transform based fitting-lane method for tracking [4]. Yoo used gradient-enhancing conversion for illumination-robust lane detection [5]. Geiger designed a Bayes model to discriminate lane-division line pixels from other pixels [6]. Peng proposed a lane-division line detection method with the statistical Hough transform based on a gradient constraint [7]. To extract significant features, Ma converted the color space of RGB to the CIELab color model and detected the lane-division-lines by using k-means clustering [8]. Son proposed a lane detection method based on the color feature and clustering method [9]. Jung proposed a lane-division line detection method based on the Haar feature [10]. Wang used lane detection by combining the self-clustering algorithm, fuzzy C-mean, and fuzzy rule to process the spatial information and Canny algorithm to extract edge features [11]. Other feature-based detection methods [12–17] also achieved good performances in normal-conditions scenarios; however, the main drawback of this approach is that it is easily disturbed by noise, as it ignores the model of the lane-division-lines.

Aiming to solve this shortcoming, many model-based detection methods are proposed. Zhou proposed a novel lane detection method based on the geometrical model and Gabor filter [18]. Wang proposed a lane detection approach based on inverse perspective mapping [19]. Baofeng proposed a detection system with a linear approximation method [20]. Although lane-division-lines detection technology has shown great development [21–26], feature extraction is still heavily influenced by complex road situations, such as light variance, under-road road signs and shadow interference [27]. Hence, more researchers are focusing on deep learning-based detection methods [28–32], which are showing better results, but these models heavily depend on the quality of training samples [33]. The artificial neural network is applied to remove noise from the image or feature map. Rashidha proposed an adaptive-size median filter for pulse noise removal based on a neural network [34]. With the development of computation, a deeper neural network was applied to reduce noise in samples. Kim Y proposed an adaptive filter based convolutional neural network [35]. Kuang removed noise by a deep convolution neural network, which showed good performances in various, unknown-noise scenarios [36].

In this paper, we aim to solve this problem by combining the advantages of feature detection and model detection methods. In particular, we propose a ridge detector to extract ridges from the aerial view map. Then, we remove the noise according to the lane-division-lines model. A sample set is generated by combining a ridge-positive sample and a noise-negative sample. In order to improve the robustness of the ridge detector, we fully utilize the information between frames. Specifically, we design a six-dimensional feature for each sample point to retrain a three-layer backpropagation (BP) neural network of each ten frames [37]. The six-dimensional feature consists of the abscissa; ordinate values; 3 × 3 convolution filter, 7 × 7 convolution filter and 11 × 11 convolution filter feature values; and the frequency within ten frames. We use the detection data set to update the neural network weights in real-time for the next ten frames' detection. The confidence level for each of the ridge points is decided based on the results of ridge detector. The confidence values are decided by the corresponding pixels of the two-dimensional Gaussian distribution's covariance matrix. To robustly fit the ridge points, we sample three ridge points from three different areas then generate k new points by two-dimensional Gaussian inverse-transformation sampling for each selected ridge point. Last, we fit the $3k + 3$ points by the random sample consensus (RANSAC) algorithm, in which the least-squares' objective function is solved by the stochastic sub-gradient descent (SGD). In this research, we used an adaptable neural network (ANN) to discriminate noise and ridge points, and improved the traditional RANSAC algorithm by considering the confidence of remaining ridge-feature points.

The remainder of this paper is organized as follows. In Section 2, related work is introduced. In Section 3, the lane-division line feature extraction method, based on an adaptable ridge detector and the between-frames neural network, is proposed, and the regional gaussian distribution random sample consensus (G-RANSAC) fitting method is proposed as well. The experimental results are shown in section 4, and finally, the conclusion is drawn in Section 5.

Nomenclature: Let $\boldsymbol{I}$ stand for real matrix of an image; $\boldsymbol{I}^{i,j}$ stands for the row i column j element of matrix $\boldsymbol{I}$. h and w represent for height and width of an image, respectively. Symbol $\otimes$ represents for convolution operation; $min(\boldsymbol{I})$ and $max(\boldsymbol{I})$ represent the minimum and maximum element of matrix $\boldsymbol{I}$, respectively.

2. Related Work

Lane-division lines can be detected using a static camera sensor or vehicle-mounted camera sensor. For the static camera sensor, the application scenarios include precise vehicle positioning and intelligent transportation. The main methods include method are based on the trajectories of vehicles [38,39] and pixel-entropy [40]. For the vehicle-mounted camera sensor, the application scenarios include intelligent vehicles and advanced driver assistant systems (ADAS) [41,42]. In this paper, we focus on lane-division line detection based on vehicle-mounted camera sensor.

Lane detection methods based on a moving camera sensor are divided into two steps, lane feature extraction, and feature points fitting. For feature extraction, Sobel and Canny edge features are usually used as the lane division lane feature, but the edge feature is susceptible to noise interference. To solve this problem, a ridge detected method based on ridge-feature is proposed [43], which contains two matrix-convolution operations, two matrix-differential operations, three matrix-point product operations, and a divergence calculation. In this paper, we propose a simpler ridge detection method based on lane width with better results in real time, as shown in Section 4.1.

For feature point fitting, Hough transform is the usual method [15], but it is an inefficient method, thus J Guo proposed a lane-division, lane-feature-fitting method based on RANSAC [44]. The main steps of this method are the following: (1) Fewer fitting points are selected randomly from the whole point set. (2) The model is fit based on the selected points in step 1 by the least-square method. (3) Step 1 is repeated until the maximum iterative number is reached. (4) The best fitting model from multiple iterations is selected, according to whole point set's lowest error. Compared with Hough transform, this method improves fitting efficiency. However, randomly selecting fitting points from whole feature points is a bad strategy for lane detection; the reason is illustrated in Figure 9b. Therefore, in Section 3.4 of this paper, we propose to divide the ridge-feature map into three regions, and then resample, based on the Gaussian distribution according to ridge coefficient. The experimental results show significantly improved fitting efficiency.

3. Materials and Methods

The detection algorithm is proposed here, which is divided into five sub-sections, as shown in Figure 1. In Section 3.1, a graphical preprocessing method based on inverse perspective transformation is presented to obtain the aerial view map. In Section 3.2, a feature extraction method based on an adaptable ridge detector is developed to extract the feature of ridges in the lane-division-lines. In Section 3.3, we extract a six-dimensional feature for each pixel to attain the neural network, which is used to discriminate between noise and ridge points. In Section 3.4, the regional G-RANSAC is proposed to robustly fit the lane-division-lines. In Section 3.5, we propose a new index, named the lane departure index (LDI), to test the performances of this lane detection method.

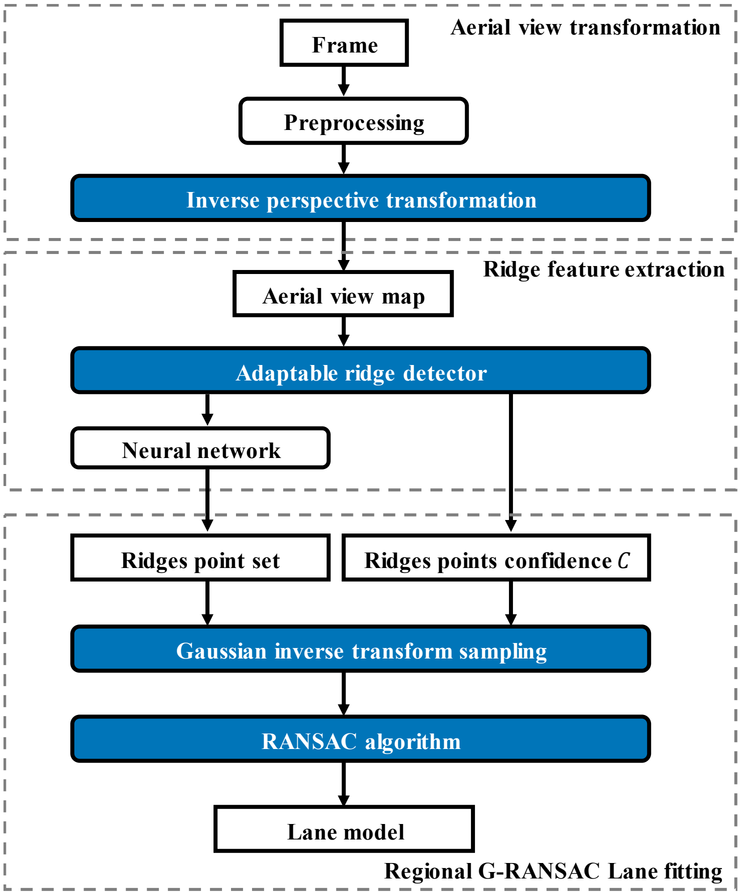

Aerial view transformation
Frame
Preprocessing
Inverse perspective transformation
Ridge feature extraction
Aerial view map
Adaptable ridge detector
Neural network
Ridges point set
Ridges points confidence C
Gaussian inverse transform sampling
RANSAC algorithm
Lane model
Regional G-RANSAC Lane fitting

Figure 1. Detection algorithm block diagram.

3.1. A Graphical-Preprocessing-Method Based Inverse-Perspective Transformation

As the frame captured from the camera is an RGB-image with noise, preprocessing methods consist of a gray image transformation and using a 5 × 5 median filter to reduce image noise. However, there are many additional interferences from image background information. Therefore, we select the region of interest (ROI) by inverse perspective transformation to attain the aerial view map, which eliminates perspective-side effects. According to the lane-division line standard [43], we choose a 35 m by 11.5 m region in the Cartesian coordinate system. The principle of the inverse perspective transformation is shown in Figure 2; (x_v, y_v, z_v) represents the vehicle coordinate system and (c, r) represents the image coordinate system. In the inverse perspective transformation, we convert the image coordinate system (c, r) into a real-world, three-dimensional coordinate system $(x_v, y_v, 0)$. The transforming relationship is described as follows:

$$\begin{cases} x_v = hc\boldsymbol{TG}\left(\frac{2\alpha}{s_x-1}r - \alpha + \theta\right)\sin\left(\frac{2\alpha}{s_x-1}c - \alpha + \gamma\right) + d \\ y_v = hc\boldsymbol{TG}\left(\frac{2\alpha}{s_y-1}r - \alpha + \theta\right)\cos\left(\frac{2\alpha}{s_y-1}c - \alpha + \gamma\right) + l \end{cases} \quad (1)$$

$$c \in N^+, r \in N^+, 0 < c \leq s_x, 0 < r \leq w - s_y$$

where (d, l, h) represent the camera's center coordinates in the vehicle coordinate system; and $\boldsymbol{T}$ and $\boldsymbol{G}$ denote the transfer and rotation matrices, respectively. s_x and s_y stand for the image resolution; θ is the angle between optical axis and the z_v = 0 plane; γ is the angle between optical axis and the vehicle coordinate system's axes, y_v; α denotes half of the camera view angle.

We convert the image to an aerial view map with Equation (1).

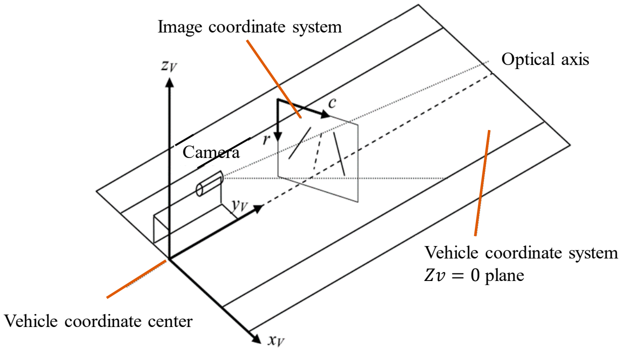


Figure 2. The principles of the inverse perspective transformation.

3.2. Ridge Detector

In general, ridges are low-level features in gray images. In comparison with edge features, ridge-features are more suitable to describe lane-division-lines in situations such as vehicle shadow interference, worn-out ground signs and insufficient illumination [19]. As shown in Figure 3, the ridge is the center line. In the traditional ridge detected method in Section 2, the gradient vector calculation and multiple matrix operation require high computational cost.

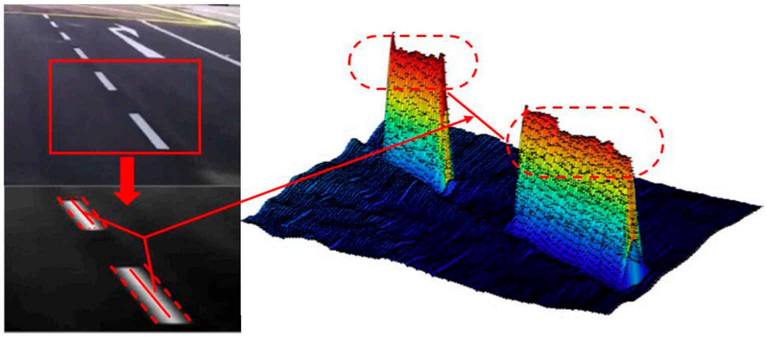


Figure 3. Ridge-feature illustration.

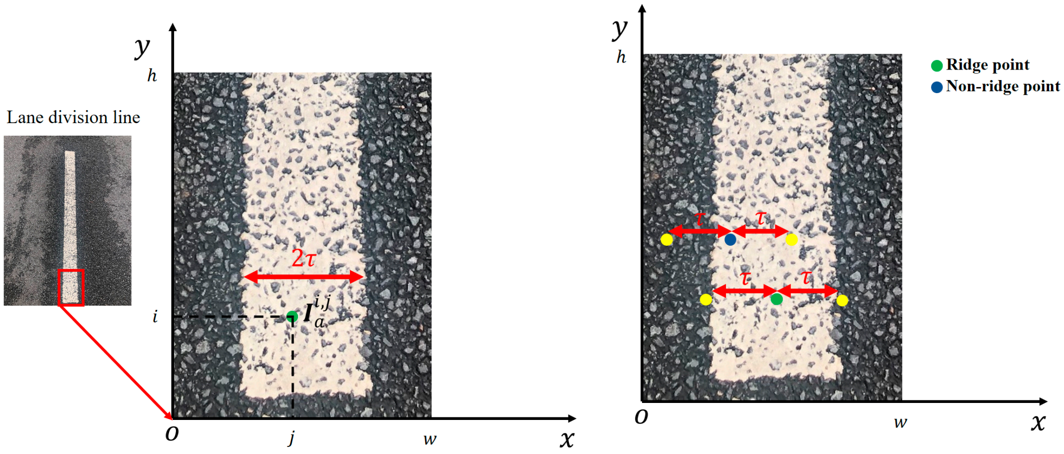
According to the geometry of lane-division-lines in the aerial view map $\boldsymbol{I}_a$, we proposed a simple ridge detector based on lane width. As shown in Figure 4a, we define the green point $\boldsymbol{I}_a^{i,j}$ as row i column j pixel's value, in the image coordinate system. Row i denotes ordinate and column j denotes abscissa; we define 2τ as the lane-division line transverse pixel number in the aerial view map. $\boldsymbol{I}_r$ stands for ridge-feature map, which is defined by Equation (2).

$$\boldsymbol{I}_r^{i,j} = \left(\boldsymbol{I}_a^{i,j} - \boldsymbol{I}_a^{i,j-\tau}\right) + \left(\boldsymbol{I}_a^{i,j} - \boldsymbol{I}_a^{i,j+\tau}\right) \tag{2}$$

$$i \in N^+, j \in N^+, 0 < i \leq h, \tau < j \leq w - \tau$$

where τ denotes half number of lane-division line transverse pixels.

As shown in Figure 4b, v_{in} is defined as the pixel value of the point in a lane-division line and v_{out} is defined as the pixel value of the point outside a lane-division line, where $v_{in} > v_{out}$. The green point $\boldsymbol{I}_a^{i,j}$ represents the ridge point value, v_{in}; its left and bright yellow point are outside the lane-division line with a v_{out} value. The blue point represents non-ridge point with a v_{in} value, its left yellow point is outside the lane-division line with a v_{out} value, and a bright yellow point is in a lane-division line with a v_{in} value. The term $\left(\boldsymbol{I}_a^{i,j} - \boldsymbol{I}_a^{i,j+\tau}\right)$ of a green point is larger than the term $\left(\boldsymbol{I}_a^{i,j} - \boldsymbol{I}_a^{i,j+\tau}\right)$ of a blue point, and the term $\left(\boldsymbol{I}_a^{i,j} - \boldsymbol{I}_a^{i,j-\tau}\right)$ of a green point equivalent to the term $\left(\boldsymbol{I}_a^{i,j} - \boldsymbol{I}_a^{i,j-\tau}\right)$ of a blue point. According to Equation (2), the ridge-feature value $\boldsymbol{I}_r^{i,j}$ of a green point is larger than the ridge-feature value $\boldsymbol{I}_r^{i,j}$ of a blue point.



(a)

(b)

Figure 4. Ridge detector illustration. (**a**) ridge point illustration. (**b**) Equation (2) illustration

In order to increase the difference between the $\boldsymbol{I}_r^{i,j}$ of a ridge point and a non-ridge point, on the basis of Equation (2), we add a punishment term $\left|\boldsymbol{I}_a^{i,j+\tau} - \boldsymbol{I}_a^{i,j-\tau}\right|$, as shown in Equation (3). At the ridge point, the punishment term equates to zero, but the term of the non-ridge point is larger than zero.

$$\boldsymbol{I}_r^{i,j} = \left(\boldsymbol{I}_a^{i,j} - \boldsymbol{I}_a^{i,j-\tau}\right) + \left(\boldsymbol{I}_a^{i,j} - \boldsymbol{I}_a^{i,j+\tau}\right) - \left|\boldsymbol{I}_a^{i,j+\tau} - \boldsymbol{I}_a^{i,j-\tau}\right| \tag{3}$$

We simplify Equation (3) to obtain the following ridge detector mathematical model:

$$\boldsymbol{I}_r^{i,j} = \left(-\boldsymbol{I}_a^{i,j-\tau} + 2\boldsymbol{I}_a^{i,j} - \boldsymbol{I}_a^{i,j+\tau}\right) - \left|\boldsymbol{I}_a^{i,j+\tau} - \boldsymbol{I}_a^{i,j-\tau}\right| \tag{4}$$

According to Equation (4), we calculate the row i column j ridge-feature value $\boldsymbol{I}_r^{i,j}$ for each pixel $\boldsymbol{I}_a^{i,j}, 0 < i \leq h, \tau < j \leq w - \tau$, which is shown in matrix form:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{I}_r^{h,1+\tau} & \cdots & \boldsymbol{I}_r^{h,w-\tau} \\ \vdots & \ddots & \vdots \\ \boldsymbol{I}_r^{1,1+\tau} & \cdots & \boldsymbol{I}_r^{1,w-\tau} \end{bmatrix} &= \begin{bmatrix} \left(-\boldsymbol{I}_a^{h,1}+2\boldsymbol{I}_a^{h,1+\tau}-\boldsymbol{I}_a^{h,1+2\tau}\right) - \left|\boldsymbol{I}_a^{h,1+2\tau}-\boldsymbol{I}_a^{h,1}\right| & \cdots & \left(-\boldsymbol{I}_a^{h,w-2\tau} + 2\boldsymbol{I}_a^{h,w-\tau}-\boldsymbol{I}_a^{h,w}\right) - \left|\boldsymbol{I}_a^{h,w}-\boldsymbol{I}_a^{h,w-2\tau}\right| \\ \vdots & \ddots & \vdots \\ \left(-\boldsymbol{I}_a^{1,1}+2\boldsymbol{I}_a^{1,1+\tau}-\boldsymbol{I}_a^{1,1+2\tau}\right) - \left|\boldsymbol{I}_a^{1,1+2\tau}-\boldsymbol{I}_a^{1,1}\right| & \cdots & \left(-\boldsymbol{I}_a^{1,w-2\tau} + 2\boldsymbol{I}_a^{1,w-\tau}-\boldsymbol{I}_a^{1,w}\right) - \left|\boldsymbol{I}_a^{1,w}-\boldsymbol{I}_a^{1,w-2\tau}\right| \end{bmatrix} \\ &= \begin{bmatrix} \left(-\boldsymbol{I}_a^{h,1}+2\boldsymbol{I}_a^{h,1+\tau}-\boldsymbol{I}_a^{h,1+2\tau}\right) & \cdots & \left(-\boldsymbol{I}_a^{h,w-2\tau} + 2\boldsymbol{I}_a^{h,w-\tau}-\boldsymbol{I}_a^{h,w}\right) \\ \vdots & \ddots & \vdots \\ \left(-\boldsymbol{I}_a^{1,1}+2\boldsymbol{I}_a^{1,1+\tau}-\boldsymbol{I}_a^{1,1+2\tau}\right) & \cdots & \left(-\boldsymbol{I}_a^{1,w-2\tau} + 2\boldsymbol{I}_a^{1,w-\tau}-\boldsymbol{I}_a^{1,w}\right) \end{bmatrix} - \begin{bmatrix} \left|\boldsymbol{I}_a^{h,1+2\tau}-\boldsymbol{I}_a^{h,1}\right| & \cdots & \left|\boldsymbol{I}_a^{h,w}-\boldsymbol{I}_a^{h,w-2\tau}\right| \\ \vdots & \ddots & \vdots \\ \left|\boldsymbol{I}_a^{1,1+2\tau}-\boldsymbol{I}_a^{1,1}\right| & \cdots & \left|\boldsymbol{I}_a^{1,w}-\boldsymbol{I}_a^{1,w-2\tau}\right| \end{bmatrix} \end{aligned} \tag{5}$$

We redefine ridge-feature map $\boldsymbol{I}_r$:

$$\boldsymbol{I}_r = \begin{bmatrix} \boldsymbol{I}_r^{h,1+\tau} & \cdots & \boldsymbol{I}_r^{h,w-\tau} \\ \vdots & \ddots & \vdots \\ \boldsymbol{I}_r^{1,1+\tau} & \cdots & \boldsymbol{I}_r^{1,w-\tau} \end{bmatrix}, \tag{6}$$

aerial view map $\boldsymbol{I}_a$:

$$\boldsymbol{I}_a = \begin{bmatrix} \boldsymbol{I}_a^{h,1} & \cdots & \boldsymbol{I}_a^{h,w} \\ \vdots & \ddots & \vdots \\ \boldsymbol{I}_a^{1,1} & \cdots & \boldsymbol{I}_a^{1,w} \end{bmatrix} \tag{7}$$

and one-dimensional ridge filters $\boldsymbol{R}$ and $\boldsymbol{P}$:

$$\boldsymbol{R} = [-1, \underbrace{0, \ldots, 0}_{\tau - 1\ zeros}, 2, \underbrace{0, \ldots, 0}_{\tau - 1\ zeros}, -1] \tag{8}$$

$$\boldsymbol{P} = [1, \underbrace{0, \ldots, 0}_{2\tau - 1\ zeros}, -1] \tag{9}$$

where τ denotes half the number of the lane-division line's transverse pixels.

According to Equations (6)–(9), we simplify Equation (5):

$$\boldsymbol{I}_r = \boldsymbol{R} \otimes \boldsymbol{I}_a - \left|\boldsymbol{P} \otimes \boldsymbol{I}_a\right| \tag{10}$$

According to Equation (10), we can convert the aerial view map $\boldsymbol{I}_a$ to ridge-feature map $\boldsymbol{I}_r$. We obtain the normalized ridge-feature map $\boldsymbol{I}_n$ by Equation (11).

$$\boldsymbol{I}_n = \begin{bmatrix} \dfrac{\boldsymbol{I}_r^{h,1+\tau} - min(\boldsymbol{I}_r)}{max(\boldsymbol{I}_r) - min(\boldsymbol{I}_r)} & \cdots & \dfrac{\boldsymbol{I}_r^{h,w-\tau} - min(\boldsymbol{I}_r)}{max(\boldsymbol{I}_r) - min(\boldsymbol{I}_r)} \\ \vdots & \ddots & \vdots \\ \dfrac{\boldsymbol{I}_r^{1,1+\tau} - min(\boldsymbol{I}_r)}{max(\boldsymbol{I}_r) - min(\boldsymbol{I}_r)} & \cdots & \dfrac{\boldsymbol{I}_r^{1,w-\tau} - min(\boldsymbol{I}_r)}{max(\boldsymbol{I}_r) - min(\boldsymbol{I}_r)} \end{bmatrix} \tag{11}$$

Pictorial examples of ridge-feature extraction are shown in Figure 5.

(a)

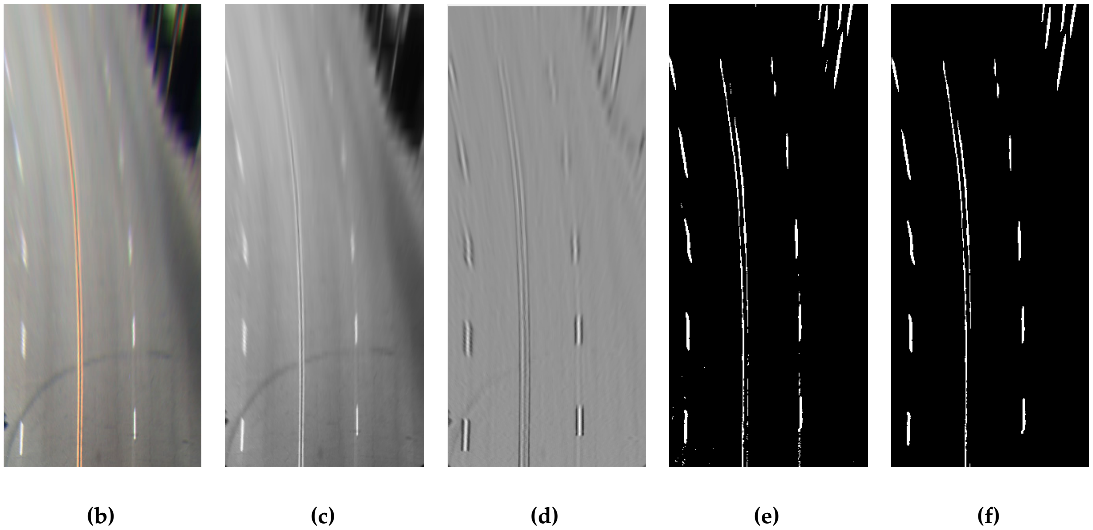


Figure 5. Ridge-feature extraction illustration. (**a**) Original map $\boldsymbol{I}$. (**b**) Aerial view map $\boldsymbol{I}_a$. (**c**) Gray map. (**d**) Ridge map $\boldsymbol{I}_r$. (**e**) Ridge binary map $\boldsymbol{I}_b$. (**f**) Map when areas with pixel numbers less than 30 are removed.

$\boldsymbol{I}_n$ describes the possibility of a pixel belonging to a ridge point set; we set the value of $\boldsymbol{I}_n$ as the confidence level for the regional G-RANSAC algorithm in Section 3.4.

Thus, we attain the statistical histogram of $\boldsymbol{I}_n$ and select the highest bin corresponding value as the threshold t to covert $\boldsymbol{I}_n$ to a binary ridge binary image $\boldsymbol{I}_b$. According to the experimental results, the mathematical morphology process is applied to image $\boldsymbol{I}_b$ to remove areas with pixel numbers less than 30.

3.3. 6-Dimensional Feature Extraction and Retraining a BP Neural Network for Removing Noise

In this section, we propose an adaptable classification method for binary ridge-feature image $\boldsymbol{I}_b$ to discriminate noise and ridge points. The noise point is comprised of another objection noise point and imaging noise point. No-objection noise is caused by the sign on the ground, tree shadow, or other vehicles; therefore, they appear in particular areas. We remove the no-objection noise point by pixel position information according to the lane-division line model. The imaging noise points are randomly distributed and caused by the camera sensor, but it is a separate and accidental process. Therefore, we remove the imaging noise points by calculating the number of surrounding feature points and the frequency of 10 consecutive frames. The frequency affects the real-time performance and adaptability of neural network. The frequency of 10 was selected based on experimental results.

A six-dimensional feature $\boldsymbol{F}^{i,j}$ for the row i column j pixel is proposed to discriminate noise and ridge points, which is as follows:

$$\boldsymbol{F}^{i,j} = \left[i, j, \boldsymbol{N}_3^{i,j}, \boldsymbol{N}_7^{i,j}, \boldsymbol{N}_{11}^{i,j}, k_{ij}\right] \tag{12}$$

$$\boldsymbol{N}_3 = \boldsymbol{ones}(3) \otimes \boldsymbol{I}_b$$

$$\boldsymbol{N}_7 = \boldsymbol{ones}(7) \otimes \boldsymbol{I}_b$$

$$\boldsymbol{N}_{11} = \boldsymbol{ones}(11) \otimes \boldsymbol{I}_b$$

where $\boldsymbol{I}_b$ stands for binary ridge-feature image; i, j denotes row i column j; $\boldsymbol{N}_3$, $\boldsymbol{N}_7$ and $\boldsymbol{N}_{11}$ denote the convolution feature matrices; $\boldsymbol{ones}(n)$ is a n-rank square matrix's element equal to 1; $\boldsymbol{N}_3^{i,j}$ is the row i column j element of the convolution feature matrix $\boldsymbol{N}_3$; $\boldsymbol{N}_7^{i,j}$ is the row i column j element of convolution feature matrix $\boldsymbol{N}_7$; $\boldsymbol{N}_{11}^{i,j}$ is the row i column j element of convolution feature matrix $\boldsymbol{N}_{11}$; and k_{ij} represents for the frequency of pixel (i, j) within 10 frames.

According to the lane-division line model, binary ridge image $\boldsymbol{I}_b$ can be divided into two sets. The ridge point set is expressed as $\boldsymbol{I}_{b+}$, in which abscissa $i \in = (25, 60)\ or\ (150, 190)\ or\ (285,300)$, which came from experiment result in Figure 6. However, the constant bounds are ineffective when in the vehicle change lane. Variable bounds are an effective method to handle this scenario, which will be exploited in the feature. The noise point set is expressed as $\boldsymbol{I}_{b-}$. Point set $\boldsymbol{I}_{b+}$ would be processed by a well-trained BP neural network and further divided into ridge point subset $\boldsymbol{I}_{b++}$ and no-ridge point subset $\boldsymbol{I}_{b+-}$. The ridges' subset $\boldsymbol{I}_{b++}$ is be fitted to the line.

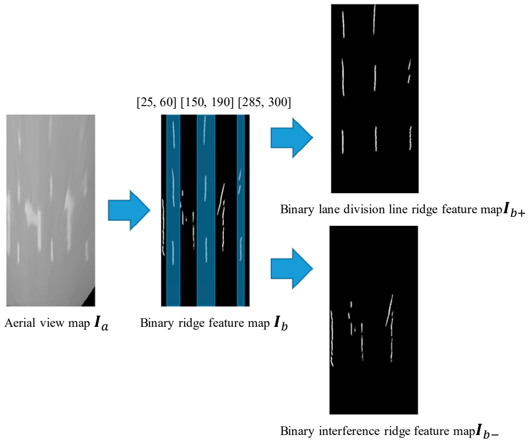

[25, 60] [150, 190] [285, 300]
Binary lane division line ridge feature map$\boldsymbol{I}_{b+}$
Aerial view map $\boldsymbol{I}_a$
Binary ridge feature map $\boldsymbol{I}_b$
Binary interference ridge feature map$\boldsymbol{I}_{b-}$

Figure 6. Pictorial example for dividing binary ridge map $\boldsymbol{I}_b$

To improve the detection robustness, within each 10 frames, we retrain the BP neural network by a positive sample set $\boldsymbol{I}_{b++}$ and negative sample sets $\boldsymbol{I}_{b+-}$ and $\boldsymbol{I}_{b-}$. Cross-entropy is applied as a loss function, and the sigmoid function is adopted as an activation function. The structure of the BP neural network is shown in Figure 7.

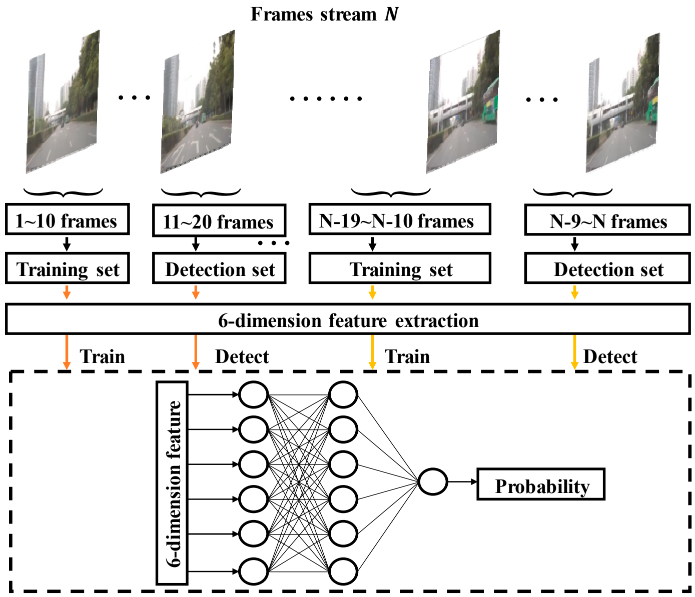


Figure 7. Retrain the backpropagation (BP) neural network for each 10 frames.

3.4. Regional G-RANSAC

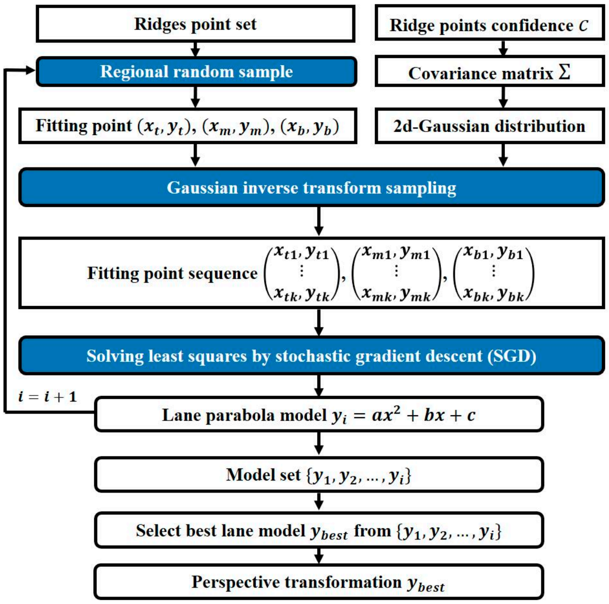

Ridges point set
Ridge points confidence C
Regional random sample
Covariance matrix Σ
Fitting point (x_t, y_t), (x_m, y_m), (x_b, y_b)
2d-Gaussian distribution
Gaussian inverse transform sampling
Fitting point sequence $\begin{pmatrix} x_{t1}, y_{t1} \\ \vdots \\ x_{tk}, y_{tk} \end{pmatrix}, \begin{pmatrix} x_{m1}, y_{m1} \\ \vdots \\ x_{mk}, y_{mk} \end{pmatrix}, \begin{pmatrix} x_{b1}, y_{b1} \\ \vdots \\ x_{bk}, y_{bk} \end{pmatrix}$
Solving least squares by stochastic gradient descent (SGD)
$i = i + 1$
Lane parabola model $y_i = ax^2 + bx + c$
Model set $\{y_1, y_2, \ldots, y_i\}$
Select best lane model y_{best} from $\{y_1, y_2, \ldots, y_i\}$
Perspective transformation y_{best}

Figure 8. Regional gaussian distribution random sample consensus (G-RANSAC) land fitting block diagram.

The RANSAC algorithm is applied to fit the point set, which contains a number of noise points. [43]. However, random sampling from the whole point set is not a good strategy for lane model fitting. As shown in Figure 8, we improve RANSAC algorithm by considering ridge point confidence and sampling areas. The binary ridges feature map $\boldsymbol{I}_b$ is divided into three areas $\boldsymbol{I}_{b\_t}$, $\boldsymbol{I}_{b\_m}$ and $\boldsymbol{I}_{b\_b}$, shown in Figure 9a. We randomly select a ridge point from each area, defined as (x_t, y_t), (x_m, y_m), and (x_b, y_b), respectively. Fitting efficiency would be improved in Figure 9c, compared with Figure 9b.

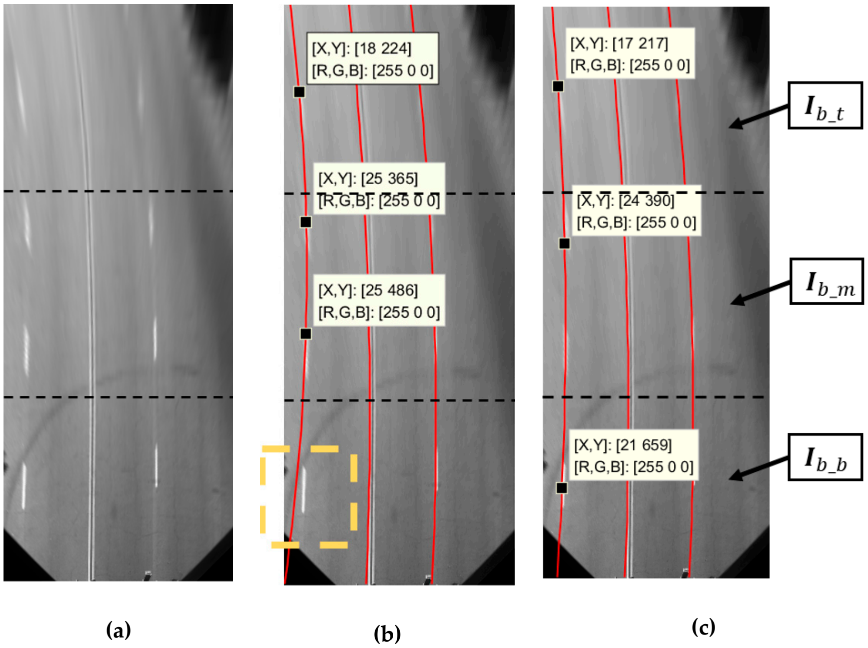


Figure 9. Regional fitting points selection illustration. (**a**) Aerial map. (**b**) Select. fitting points randomly from whole map. (**c**) Select fitting points randomly from three areas.

The feature points selection method is suitable for lane fitting, but it is still possible to select noise points. Therefore, we assume that the selected row, i column, j ridge point and $\boldsymbol{I}_b^{i,j}$ coordinate (j,i), is subject to two-dimensional Gaussian distribution, in which probability density function is defined as follows:

$$f_{j,i}(x,y) = \frac{1}{2\pi\sqrt{(\sigma_1^2\sigma_2^2)}}\exp\left(-\frac{1}{2}\left[\left(\frac{x-u_1}{\sigma_1}\right)^2 + \left(\frac{y-u_2}{\sigma_2}\right)^2\right]\right)$$
$$x \in N^+, y \in N^+, 0 < x \le w, 0 < y \le h \tag{13}$$

where $\begin{pmatrix}u_1\\u_2\end{pmatrix} = \begin{pmatrix}j\\i\end{pmatrix}$ means vector, and σ_1 and σ_2 are the covariance matrix $\boldsymbol{\Sigma}_{i,j}$'s elements. We hypothesis that variables x and y are non-correlated, hence covariance matrix $\boldsymbol{\Sigma}_{i,j}$ is defined as $\boldsymbol{\Sigma}_{i,j} = \begin{bmatrix}{\sigma_1}^2 & 0\\0 & {\sigma_2}^2\end{bmatrix}$.

The covariance matrix $\boldsymbol{\Sigma}_{i,j}$ describes the uncertainty degree of selected point $\boldsymbol{I}_b^{i,j}$ belonging to the ridge point set. In other words, the selected point $\boldsymbol{I}_b^{i,j}$ in the ridge point set is the fitting target, so we set the covariance matrix $\boldsymbol{\Sigma}_{i,j}$ as a small value to limit the inverse transformation resampling range. The other case is that the selected point $\boldsymbol{I}_b^{i,j}$ is not in the ridge point set; then, we set a big covariance matrix $\boldsymbol{\Sigma}_{i,j}$, giving a chance for this point to generate ridge points at the inverse-transformation sampling stage.

We then define the confidence level $\boldsymbol{C}^{i,j}$ as the probability of selected point $\boldsymbol{I}_b^{i,j}$, which belongs the to ridge point set. $\boldsymbol{C}^{i,j}$ denotes the row i, column j element of the ridge points' confidence matrix $\boldsymbol{C}$, which is defined in Equation (14):

$$\boldsymbol{C} = \frac{\boldsymbol{ones}(3)\otimes\boldsymbol{I}_n}{9} \tag{14}$$

where $\boldsymbol{ones}(3)$ is when the three-rank square matrix's element equals 1. $\boldsymbol{I}_n$ stands for normalized ridge-feature map. Covariance matrix $\boldsymbol{\Sigma}_{i,j}$ is calculated with confidence $\boldsymbol{C}^{i,j}$ of the point $\boldsymbol{I}_b^{i,j}$ in binary ridge-feature map $\boldsymbol{I}_b$, as shown in Equation (15):

$$\boldsymbol{\Sigma}_{i,j} = \begin{bmatrix}\frac{\alpha}{\boldsymbol{C}^{i,j}} & 0\\0 & \frac{\alpha}{\boldsymbol{C}^{i,j}}\end{bmatrix} \tag{15}$$

where α denotes scale coefficient.

In summary, the selected point $\boldsymbol{I}_b^{i,j}$ initiates a two-dimensional Gaussian distribution with mean vector (j,i) and covariance matrix $\boldsymbol{\Sigma}_{i,j}$. Because abscissa j and ordinate i are non-correlated, the two-dimensional Gaussian distribution can be converted into two one-dimensional Gaussian distributions, $\mathcal{N}(\mu,\sigma^2)$. Therefore, abscissa distribution $x\sim\mathcal{N}(j,\frac{\alpha}{\boldsymbol{C}^{i,j}})$ and ordinate distribution $y\sim\mathcal{N}(i,\frac{\alpha}{\boldsymbol{C}^{i,j}})$ arise; their probability density functions are shown in Equations (16) and (17), respectively.

$$f(x) = \frac{1}{\sqrt{2\pi\times\frac{\alpha}{\boldsymbol{C}^{i,j}}}}exp\left(-\frac{(x-j)^2}{2\frac{\alpha}{\boldsymbol{C}^{i,j}}}\right) \tag{16}$$

$$f(y) = \frac{1}{\sqrt{2\pi\times\frac{\alpha}{\boldsymbol{C}^{i,j}}}}exp\left(-\frac{(y-i)^2}{2\frac{\alpha}{\boldsymbol{C}^{i,j}}}\right) \tag{17}$$

The corresponding cumulative distribution function is shown in Equations (18) and (19).

$$F(x) = \frac{1}{\sqrt{2\pi\times\frac{\alpha}{\boldsymbol{C}^{i,j}}}}\int_{-\infty}^{x} exp\left(-\frac{(x-j)^2}{2\frac{\alpha}{\boldsymbol{C}^{i,j}}}\right)dx \tag{18}$$

$$F(y) = \frac{1}{\sqrt{2\pi\times\frac{\alpha}{\boldsymbol{C}^{i,j}}}}\int_{-\infty}^{y} exp\left(-\frac{(y-i)^2}{2\frac{\alpha}{\boldsymbol{C}^{i,j}}}\right)dy \tag{19}$$

where $exp(x)$ denotes an exponential function, α denotes a scale coefficient and (j,i) is the coordinate point of ridge-feature map $\boldsymbol{I}_r$. In Equations (18) and (19), (j,i) represents the mean vector of the Gaussian distribution.

We describe Equations (18) and (19) by the error function $erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-\eta^2} d\eta$, as shown in Equations (20) and (21), respectively.

$$\Phi_x(z) = \frac{1}{2}\left[1 + erf\left(\frac{z-j}{\sqrt{2\frac{\alpha}{C^{i,j}}}}\right)\right] \quad (20)$$

$$z \in N^+, 0 < z \le w$$

$$\Phi_y(t) = \frac{1}{2}\left[1 + erf\left(\frac{t-i}{\sqrt{2\frac{\alpha}{C^{i,j}}}}\right)\right] \quad (21)$$

$$t \in N^+, 0 < t \le h$$

The inverse function of $\Phi_x(z)$ and $\Phi_y(t)$ is shown in Equations (22) and (23).

$$\Phi_x{}^{-1}(p) = \sqrt{2\frac{\alpha}{C^{i,j}}} \times erf^{-1}(2p-1) + j \quad (22)$$

$$min\big(\Phi_x(z)\big) \le p \le max\big(\Phi_x(z)\big)$$

$$\Phi_y{}^{-1}(q) = \sqrt{2\frac{\alpha}{C^{i,j}}} \times erf^{-1}(2q-1) + i \quad (23)$$

$$min\left(\Phi_y(t)\right) \le q \le max\left(\Phi_y(t)\right)$$

where α denotes scale coefficient, and (j,i) is a coordinate point of ridge-feature map $\boldsymbol{I}_r$, in Equations (18) and (19). (j,i) denotes the mean vector of the Gaussian distribution.

According to inverse transformation sampling [45], we attain the selected point sequence $\begin{pmatrix} x_{i1}, y_{i1} \\ \vdots \\ x_{ik}, y_{ik} \end{pmatrix}$ by $\begin{pmatrix} \Phi_x{}^{-1}(p_1), \Phi_y{}^{-1}(q_1) \\ \vdots \\ \Phi_x{}^{-1}(p_k), \Phi_y{}^{-1}(q_k) \end{pmatrix}$, where k represents the inverse transformation sampling numbers. $(p_1, p_2, \dots, p_k)$ and $(q_1, q_2, \dots, q_k)$ are subject to uniform distribution $\begin{pmatrix} x_{i1} \\ \vdots \\ x_{ik} \end{pmatrix}$, which is subject to $\mathcal{N}(j, \frac{\alpha}{C^{i,j}})$ and $\begin{pmatrix} y_{i1} \\ \vdots \\ y_{ik} \end{pmatrix}$, which are subject to $\mathcal{N}(i, \frac{\alpha}{C^{i,j}})$.

The traditional RANSAC algorithm, which randomly fits selected points by least-squares, in which the objective function is l, is shown in Equation (24).

$$min\ l = \sum_{i=1}^{n}(y_i - f(x_i))^2 \quad (24)$$

where n represents the number of fitting points, (x_i, y_i) denotes the i^{th} fitting points and $f(x_i)$ denotes the fitting model. We define $f(x) = ax^2 + bx + c$, for each x_i, where x_i belongs to (0, w]. In this paper, we choose the stochastic sub-gradient descent (SGD) method [46] to solve Equation (24), because the number of selected point sequence $\begin{pmatrix} x_{i1}, y_{i1} \\ \vdots \\ x_{ik}, y_{ik} \end{pmatrix}$ is large if the inverse-transform sampling parameter k is large. The objective function's partial derivatives with respect to a, b and c are as follows:

$$\frac{\partial L}{\partial a} = 2\sum_{i=1}^{n}\sum_{j=1}^{k}\left(ax_{ij}^2 + bx_{ij} + c - y_{ij}\right)x_{ij}^2 \quad (25)$$

$$\frac{\partial L}{\partial b} = 2\sum_{i=1}^{n}\sum_{j=1}^{k}\left(ax_{ij}^2 + bx_{ij} + c - y_{ij}\right)x_{ij} \quad (26)$$

$$\frac{\partial L}{\partial c} = 2\sum_{i=1}^{n}\sum_{j=1}^{k}\left(ax_{ij}^{2} + bx_{ij} + c - y_{ij}\right) \tag{27}$$

where n represents the number of randomly selected points and k represents the number of inverse transform sampling.

The stochastic sub-gradient descent for solving the least-squares receives four input parameters: (i) step size λ, (ii) the number of iterations t_{max}, (iii) the number of examples to use for calculating sub-gradient l, (iv) and the fitting point sequence $\boldsymbol{X}$. Algorithm 1 describes the proposed method in pseudocode.

Algorithm 1 The stochastic sub-gradient descent for solving least-squares

```
Input:
    $\lambda$,    $t_{max}$,    $l$,    $\boldsymbol{X}$
Initialize:
$a^{(1)} = 1,\ b^{(1)} = 1, c^{(1)} = 1;$
For t=1, 2, …, $t_{max}$ do
    Choose $\boldsymbol{X}_t \subseteq \boldsymbol{X}$, where $|\boldsymbol{X}_t| = l$
    Set $\eta_t = \frac{1}{\lambda t}$
    $a^{(t+1)} \leftarrow a^{(t)} - \eta_t \frac{\partial L}{\partial a}$ , $b^{(t+1)} \leftarrow b^{(t)} - \eta_t \frac{\partial L}{\partial b}$ , $c^{(t+1)} \leftarrow c^{(t)} - \eta_t \frac{\partial L}{\partial c}$
End for.
```

3.5. Lane Departure Index (LDI)

The true-positive rate (TPR) and false-positive rate (FPR) are common indices in lane detection [16], which is used to measure the ratio of correctly fitting frames to the total frames, described as follows:

$$\text{TPR} = \frac{NTP}{NLT} \tag{28}$$

$$\text{FPR} = \frac{NFP}{NLT} \tag{29}$$

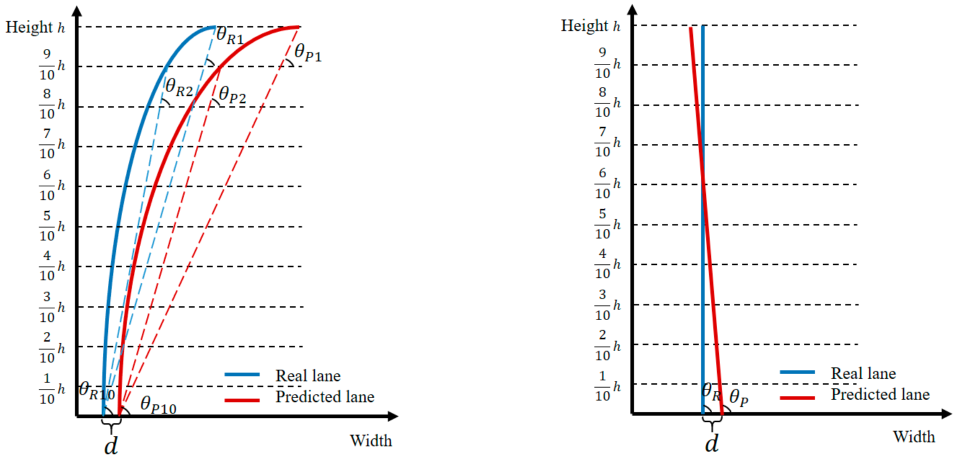
where NTP (number of true-positive) is the number of correctly predicted lane detections; NLT (number of lanes positive) is the real lane number in the test video; and NFP (number of false-positive) is the number of wrongly predicted lane detections.

However, for a lane-division line, the judgment of TPR and FPR indices are binary: correct predictions or wrong predictions. It is very dangerous for an autopilot system to predict a lane division lane inaccurately, so it is necessary to find an index to describe the departure degree between the predicted lane and real predicted in a single frame.

Herein, we propose a new measure index termed the lane departure index (LDI) to describe the departure degree. The curve-line lane model is simplified to ten straight lines based on ten points of curved lane, as shown in Figure 10a and defined in Equation (30).

$$\begin{gathered} \text{LDI} = 100 - \sum_{j=1}^{m}\sum_{i=1}^{n}\left(1 + \frac{d_{ji}}{w}\right) \times \left[\alpha_1\left(\theta_{R1\_ji} - \theta_{P1\_ji}\right)^2 + \alpha_2\left(\theta_{R2\_ji} - \theta_{P2\_ji}\right)^2 + \cdots + \alpha_{10}\left(\theta_{R10\_ji} - \theta_{P10\_ji}\right)^2\right] \\ i \in N^{+}, j \in N^{+}, 0 < i \leq n, 0 < j \leq m, \alpha_l = \frac{1}{10}, l \in N^{+}, 0 < l \leq 10 \end{gathered} \tag{30}$$

where d_{ji} denotes the abscissa error between ith real lane, and ith predicted lane of jth frame when the ordinate is zero; α_1 to α_{10} are the weight of ten simplified straight line; θ_{R1_ji} denotes the first simplified straight line of ith real lane in jth frame; θ_{P1_ji} denotes the first simplified straight line of ith predicted lane in jth frame; θ_{R2_ji} denotes the second simplified straight line of ith real lane in jth frame; θ_{P2_ji} denotes the second simplified straight line of ith predicted lane in jth frame; θ_{R10_ji} denotes the tenth simplified straight line of ith real lane in jth frame; θ_{P10_ji} denotes the tenth simplified straight line of ith predicted lane in jth frame; m denotes video frame number; and n denotes the number of the jth frame real lane-division line.



(a) **(b)**

Figure 10. Illustration of lane departure index (LDI). (**a**) Curved lane in an aerial view map. (**b**) Straight lane in aerial view map.

Shown in Figure 10b, as a particular case of a two-degree polynomial curve lane, Equation (30) also holds true for a straight lane, and the terms $\left(\theta_{ji}^{R1}-\theta_{ji}^{P1}\right), \left(\theta_{ji}^{R2}-\theta_{ji}^{P2}\right), \cdots$, and $\left(\theta_{ji}^{R10}-\theta_{ji}^{P10}\right)$ are equal to each other.

4. Results

In this section, three experiments are carried out. In Section 4.1, there is a comparison experiment about the operating speed of the ridge-feature model [43]. The proposed method is given and the computational complexity is analyzed. In Section 4.2, the median filter, regional noise removing method and BP neural network are applied to remove noise adaptability, and their effectiveness is verified. In Section 4.3, the effectiveness of fitting method regional G-RANSAC is verified by comparing with traditional RANSAC and the Hough transform. Lastly, in Section 4.4, the comparative experimental results of the whole proposed method and other lane detection methods are given to verify the improvement in challenging scenarios.

4.1. An Analysis of the Ridge-Feature Extraction Method's Operating Speed

The traditional ridge-feature extraction method [43] contains two matrix convolution operations, two matrix differential operations, three matrix point product operations and a divergence calculation. The proposed method in Equation (10) contains two matrix convolution operations and a matrix subtraction operation.

We compared the running speed of the traditional ridge-feature extraction method and the proposed method. Software platform: MATLAB R2018b. Hardware platform: CPU: Intel Core i5-4570 CPU (3.20 GHz), Memory: 32 GB and GPU: NVIDIA GeForce GTX 1080 Ti. Testing video: shown in Table 3.

Table 1. Comparison experiment about the operating speed of the traditional method and the proposed method.

Testing Scenario Video	Method [43] (frame/s)	Proposed Method (frame/s)
Scenario 1	32	82
Scenario 2	30	81
Scenario 3	31	79
Scenario 4	34	85

As shown in Table 1, compared with method [43], the proposed method operating speed is improved 2.56, 2.70, 2.55 and 2.50 times in the four testing scenario videos respectively.

4.2. A BP Neural Network Applied to Remove Noise

We chose 100 frames from a driving video and retrained the BP neural network for each 10 frames and tested the neural network with the next 10 frames. The neural network's receiver operating characteristic (ROC) curve is shown in Figure 11. There are nine ROC curves in Figure 11, representing each 10-frame training of the neural network. With training, the neural network performance got better. More specifically, the classification performances of the BP neural network for the next 10 frames are listed in Table 2. The results illustrate that the neural network's accuracy is equal to 0.830 at the beginning and after nine retraining frames, the accuracy increase to 0.889, which verifies that the neural network is adaptable for the current detection scenario.

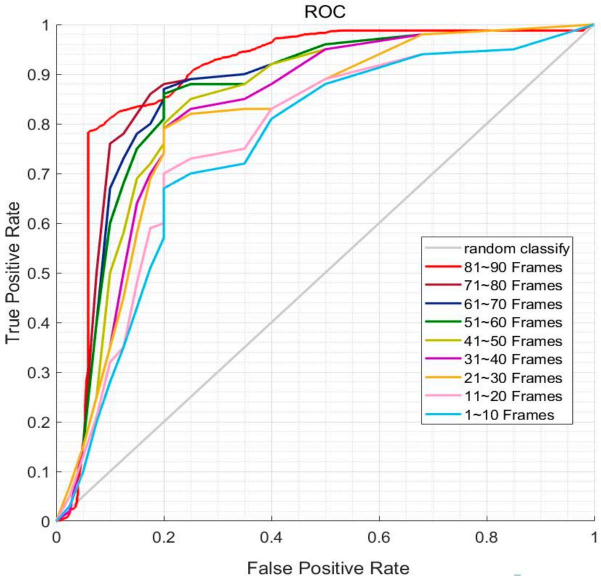


Figure 11. receiver operating characteristic (ROC) curve of the BP neural network.

Table 2. Result of the BP neural network for different frames.

Train Dataset	Test Dataset	Accuracy	mAP
1–10 frames	11–20 frames	0.830	0.753
11–20 frames	21–30 frames	0.855	0.776
21–30 frames	31–40 frames	0.861	0.802
31–40 frames	41–50 frames	0.860	0.816
41–50 frames	51–60 frames	0.877	0.832
51–60 frames	61–70 frames	0.882	0.847
61–70 frames	71–80 frames	0.881	0.854
71–80 frames	81–90 frames	0.886	0.865
81–90 frames	91–100 frames	0.889	0.883

In order to test the degree of improvement of the BP neural network applied to removing noise, we compared the proposed method with a hybrid median filter [47] and the regional noise removing method [48] in four different types of testing scenarios, which are listed in Table 3, including a scenario with normal illumination and good pavement; one with intense illumination and shadow interruption; another with normal illumination and a sign-on-the-ground interruption; and finally, one with poor illumination and vehicle interference. The fitting method is by the traditional RANSAC algorithm.

Table 3. Different testing video scenarios.

Testing Scenario Video	Illumination condition	Pavement condition	Frame number
Scenario 1	normal	good	2327
Scenario 2	intense	shadow interrupt	2139
Scenario 3	normal	road sign interrupt	2231
Scenario 4	poor	vehicle interrupt	2391

As shown in Table 4, in the normal illumination and good pavement scenario video, the performance of the proposed method is similar to hybrid median filter and regional noise removing, but the proposed method is better than hybrid median filter and regional noise removing in abnormal illumination and bad pavement conditions. Furthermore, the proposed method achieves a more stable and better performance in different scenarios. This experiment aims to verify the effectiveness of the proposed de-noising method, as the general fitting method, the traditional RANSAC algorithm, is applied to fit ridge-feature points to generate results in Table 4, the results in bold means the best performance in the corresponding scenarios.

Table 4. Comparative results of noise removing methods.

Testing Scenario Video	Noise Removing Methods	TPR (%)	LDI
Scenario 1	Hybrid median filter	93.81	42.88
	Regional noise removing	94.92	50.19
	Proposed method	**95.13**	**52.31**
Scenario 2	Hybrid median filter	83.91	32.71
	Regional noise removing	86.32	35.45
	Proposed method	**92.21**	**40.16**
Scenario 3	Hybrid median filter	85.03	33.16
	Regional noise removing	86.12	37.52
	Proposed method	**91.32**	**49.19**
Scenario 4	Hybrid median filter	85.91	32.02
	Regional noise removing	84.23	35.54
	Proposed method	**90.43**	**39.51**

4.3. Regional G-RANSAC Fitting Method Verification

The experimental parameters are shown as follows: scale coefficient $\alpha = 1.2$, fitting point number $n = 3$, inverse transformation sampling number $k = 100$, RANSAC iteration number 60, SGD step size $\lambda = 0.01$, SGD iterations $t = 200$ and the number of examples to use for calculating sub-gradient $l = 10$.

To compare with Hough transform and the traditional RANSAC algorithm, we tested the proposed method in four different types of testing scenarios: normal illumination and good pavement; intense illumination and shadow interruption; normal illumination and a sign-on-the-ground interruption; and finally, one with poor illumination and vehicle interference (as listed in Table 5). The proposed de-noising method described in section 3.3 is applied to experiment in Table 5, the results in bold means the best performance in the corresponding scenarios.

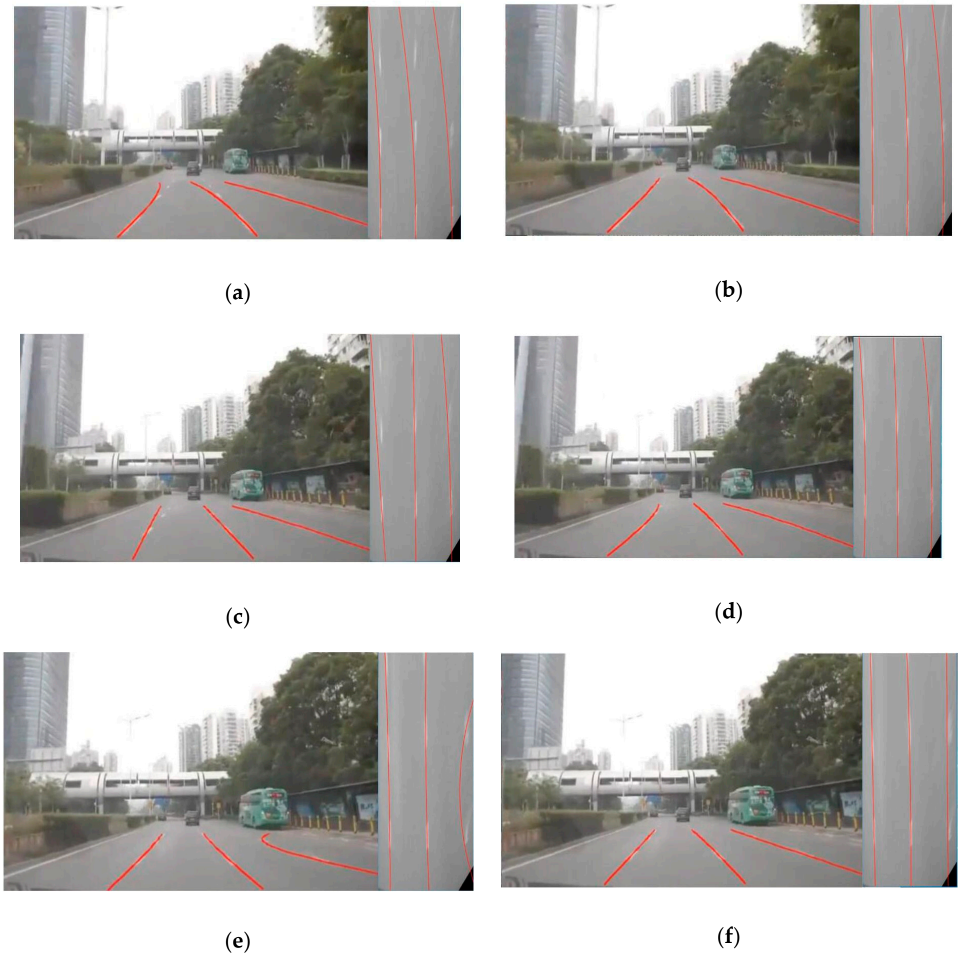
Table 5. Comparative results of fitting methods in different test scenarios.

Testing Scenario Video	Fitting Method	TPR (%)	LDI
Scenario 1	Hough transform	98.87	54.88
	Traditional RANSAC	95.13	52.31
	Proposed method	**99.02**	**66.16**
Scenario 2	Hough transform	88.71	37.21
	Traditional RANSAC	92.21	40.16
	Proposed method	**96.92**	**54.48**

	Hough transform	85.67	33.22
Scenario 3	Traditional RANSAC	91.32	49.19
	Proposed method	**96.65**	**55.98**
	Hough transform	83.21	30.10
Scenario 4	Traditional RANSAC	90.43	39.51
	Proposed method	**91.61**	**52.61**

The proposed method achieved 99.02%, 96.92%, 96.65%, and 91.61% TPR in the four different testing scenarios, respectively. In addition, for the LDI, the proposed method achieved 20.55% and 26.48% more than the Hough transform and traditional RANSAC in normal illumination and good pavement conditions; 46.41% and 35.66% more in intense illumination and shadow interruption scenarios; 68.51% and 13.80% more in normal illumination and sign on the ground interruption scenarios; and 74.78% and 33.16% more in poor illumination and vehicle interferance scenario.

In Figure 12, the comparative results of traditional RANSAC and regional G-RANSAC are shown. Figure 12a,c and e show the fitting by traditional RANSAC algorithm. Figure 12b,d and f is fitting by regional G-RANSAC. Shown in Figure 12a–d, the traditional RANSAC algorithm selects fitting points from the whole ridges' point feature map, which increases the probability of selecting noise points. We proposed selecting fitting points from three divided areas to improve the fitting effectiveness. Shown in Figure 12e,f the traditional RANSAC algorithm missed the feature point, hence the fitting effect in Figure 12e is poor, but the proposed method considers the Gaussian distribution of feature points based on a ridges' confidence, thus improved the robustness of lane fitting.



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 12. The comparative result of traditional RANSAC and regional G-RANSAC. (**a**, **c**, **e**) fitting results of the traditional RANSAC, (**b**, **d**, **f**) fitting results of the regional G-RANSAC.

4.4. Lane Detection Frame Verifying Experience

Here, the proposed lane detection frame tests are shown. We chose recent lane detection methods to verify the effectiveness of the proposed lane detection method. Comparative methods are listed in Table 6. Testing scenario videos are the same video as above, including scenarios with normal illumination and good pavement; intense illumination and shadow interruption; normal illumination and a sign-on-the-ground interruption; and finally, one with poor illumination and vehicle interference.

Table 6. Comparison of lane detection methods.

Lane Detection	Preprocess	Feature Extraction	Lane Model	Fitting Method
Method 1[4]	Median filter	Sobel filter	Straight line	Hough transform
Method 2 [9]	ROI selection based on vanishing point	Canny filter	Straight line	Least square
Method 3 [38]	Gray image	Vehicle trajectories	Straight line	Least square
Proposed method	Inverse perspective transformation	Ridge detector	Parabola	Regional G-RANSAC

As shown in Table 7 (the results in bold means the best performance in the corresponding scenarios), the proposed method has better performances than method 1 in four different types of testing scenario videos; the improvement is significant in the testing scenario of occlusion interruption, including sign interruption and vehicle interruption. The method 1 lane fitting by Hough transform, achieved 98.13% TPR and 64.10 LDI in normal illumination and good pavement condition testing scenarios, but 82.75% TPR 30.32 LDI and 85.14% TPR 33.17 LDI in sign interruption and vehicle interference scenarios, respectively. The results indicate that Hough transform is not good at the occlusion interruption scenario.

Method 3, based on vehicle trajectories, performed badly: 50.64%, 48.81%, 50.37% and 47.32% TPR; and 30.94, 21.96, 29.40 and 27.16 LDI in the four different types of testing scenarios, respectively. Additionally, method 3 fails to detect lane-division-lines in some situations; for example, when there is no preceding vehicle or the preceding vehicle changes lanes. However, compared with scenario 1, method 3 only reduced TPR 0.27% in scenario 3, while the proposed method reduced it 2.37%. The reason is that method 3 focuses on vehicle trajectories rather than lane-division line, so the fitting result is not affected by the sign marking on the ground.

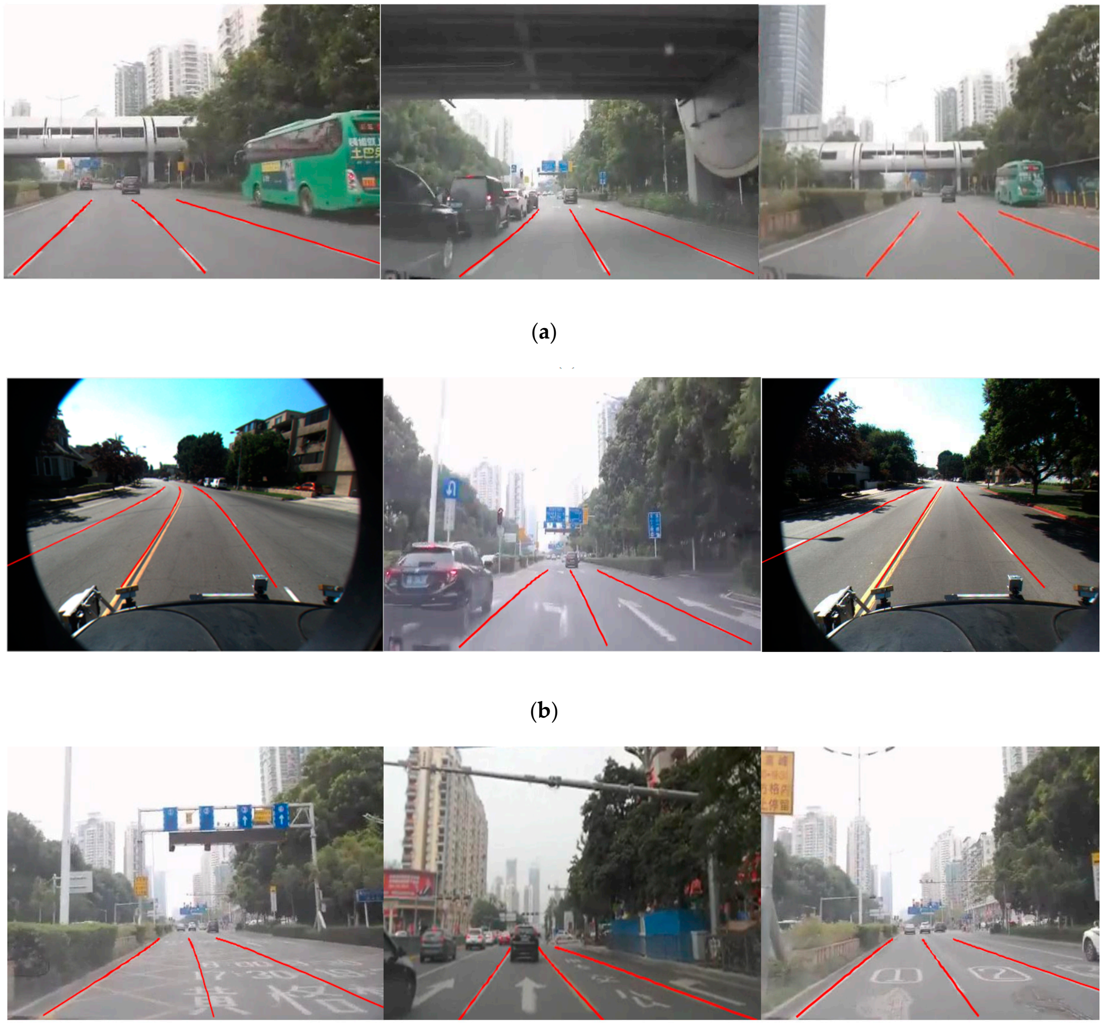
The proposed method has satisfactory performances in four different types of testing scenarios, achieving 99.02%, 96.92%, 96.65% and 91.61% TPR; and 66.16, 54.85, 55.98 and 52.61 LDI in the four different types of testing scenarios, respectively. The results show that the proposed method is effective in challenging scenarios.

Table 7. Comparative results of lane detection methods.

Testing Scenario Video	Lane Detection Method	TPR (%)	LDI
Scenario 1	Method 1	98.13	64.10
	Method 2	94.54	49.28
	Method 3	50.64	30.94
	Proposed method	**99.02**	**66.16**
Scenario 2	Method 1	88.23	38.23
	Method 2	87.75	31.18
	Method 3	48.81	21.96
	Proposed method	**96.92**	**54.85**
Scenario 3	Method 1	82.75	30.32
	Method 2	87.41	34.19

	Method 3	50.37	29.40
	Proposed method	**96.65**	**55.98**
Scenario 4	Method 1	85.14	33.17
	Method 2	82.66	29.90
	Method 3	47.32	27.16
	Proposed method	**91.61**	**52.61**

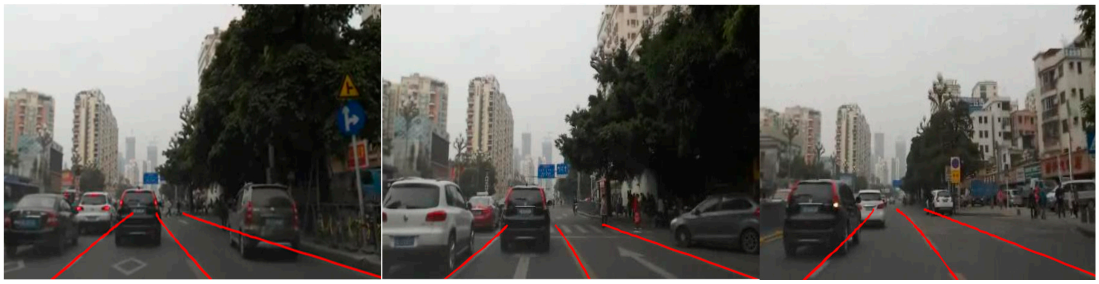
In Figure 13a, because of good lighting conditions and road conditions, there are few noise points in the ridge-feature map, resulting in good detection. In Figure 13b, the testing scenario has shadow interruption due to strong lighting, whereas the proposed method can remove shadow noise points and show good lane-division line fitting. In Figure 13c, the sign on the ground is the main interference, but the lane-division line model and BP neural network has the capability to discriminate between ridge pixel points and sign pixel points, which results in a good git. In Figure 13d, the scenario has poor illumination, vehicle interfere generates a lot of noise points and it lacks key ridge points; however, with the advantage of the G-RANSAC algorithm, the proposed method can fit the lane-division line with a small number of ridge points.



(**a**)

(**b**)

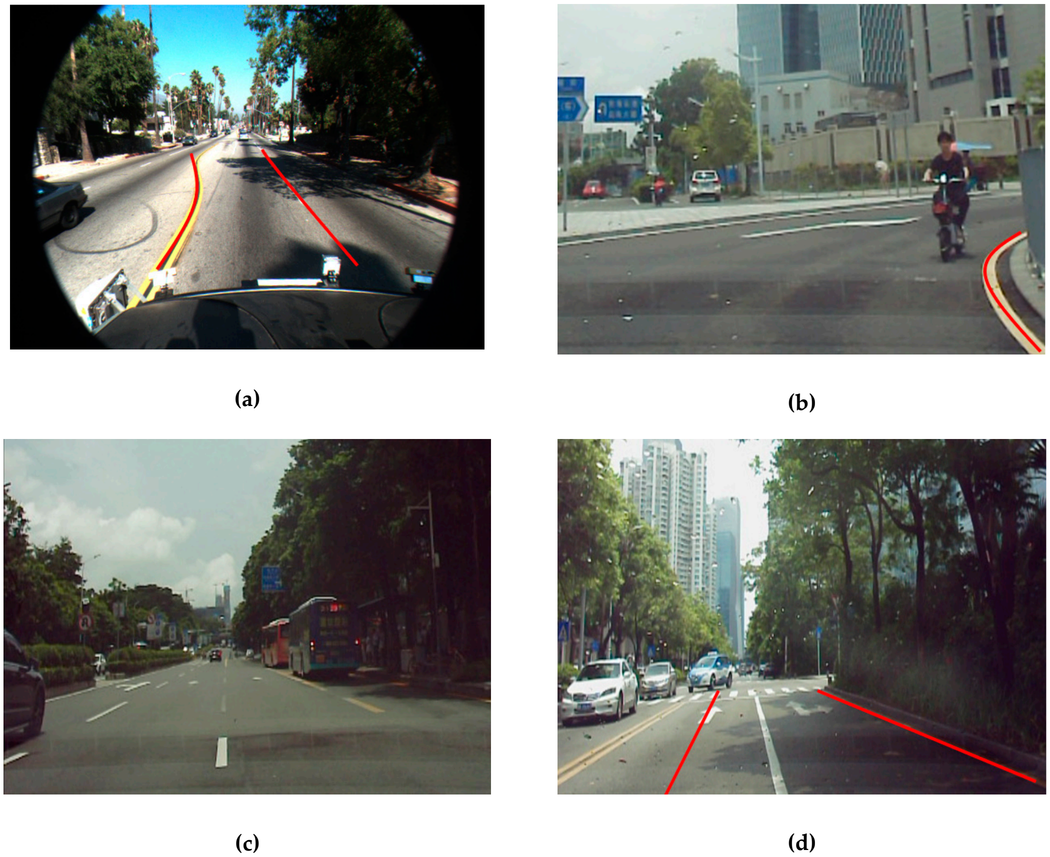
(c)



(d)

Figure 13. The proposed method detects the lane in four different types of scenarios. (**a**) Scenario 1 with normal illumination and good pavement. (**b**) Scenario 2 with intense illumination and shadow interruption. (**c**) Scenario 3 with normal illumination and sign-on-the-ground interruption, and (**d**) scenario 4 with poor illumination and vehicle interference.

Herein, the drawbacks of the proposed method are described. First, as the proposed method fitting model, Parabola, cannot provide perfect fitting when the lane-division line bends continuously, shown in Figure 14a. Second, the proposed method fails to detect lane-division-lines when its abscissa crosses three ranges (25, 60) and (150, 190) and (285, 300) at same time, shown in Figure 14b. Third, the proposed method fails to detect lane-division-lines when the vehicle itself changes lane, shown in Figure 14c,d.



(a)

(b)

(c)

(d)

Figure 14. Incorrect detection with the proposed method. (**a**) The left lane-division line is not a good fit. (**b**) Detection failure of the left lane division. (**c**) Detection failure of lane division. (**d**) The left predicted lane division does not match the left, real lane-division line.

5. Conclusions

In this paper, we proposed a lane-division-lines detection method based on ridge detector and regional G-RANSAC. The main innovation is summarized as: First, we removed noise points by an adaptable neural network. The experimental results verified that the adaptable neural network achieves better detection performance than a hybrid median filter and regional noise removing, in challenging scenarios. Secondly, we improved the traditional RANSAC by considering the confidence levels of pending fitting points. The experimental results indicate that the regional G-RANSAC achieves better detection performance in TPR and LDI compared to traditional RANSAC and Hough transform in different scenarios. Last, we compared the whole proposed method with other lane detection methods on four types of testing scenario videos, including a scenario with normal illumination and good pavement; one with intense illumination and shadow interruption; another with normal illumination and a sign-on-the-ground interruption; and finally, one with poor illumination and vehicle interference. The experimental results show, regardless of normal or challenging scenarios, the proposed method achieves 0.91%, 9.85%, 10.57% and 7.60% improvements in TPR; and 3.21%, 43.47%, 84.63% and 58.61% improvements in LDI in the four different types of testing scenarios compared to the other lane detection methods, especially in the sign and vehicle interference scenario. Since the proposed method cannot adaptively separate the lane-division line regions on the abscissa, the lane-division line cannot be well fitted in the case where the lane dividing line bends continuously and the vehicle changes lanes. Variable region bounds are an effective method to solve the problem [49], which will be studied in the future.

Author Contributions: Conceptualization, Z.L. and Y.X.; methodology, S.X. and Y.X.; software, S.X.; validation, Z.L.; formal analysis, L.L.; investigation, Y.X.; resources, S.X.; data curation, S.X.; writing—original draft preparation, Z.L.; writing—review and editing, L.L.; visualization, Z.W.; supervision, Z.W and J.S.; project administration, Y.X.; funding acquisition, Y.X.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61403259; the National Natural Science Foundation of China, grant number 51577120; and the Science and Technology Research and Development Foundation of Shenzhen, grant number JCYJ20170302142107025

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hillel, A.B.; Lerner, R.; Levi, D.; Raz, G. Recent progress in road and lane detection: A survey. *Mach. Vis. Appl.* **2014**, *25*, 727–745.
2. Ramer, C.; Lichtenegger, T.; Sessner, J.; Landgraf, M.; Franke, J. An adaptive, color based lane detection of a wearable jogging navigation system for visually impaired on less structured paths. In Proceedings of the 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), Singapore, 26–29 June 2016; pp. 741–746.
3. Dang, H.S.; Guo, C.J. Structure Lane Detection Based on Saliency Feature of Color and Direction. In *Applied Mechanics and Materials*; Trans Tech Publications: Zurich, Switzerland, 2014; pp. 2876–2879.
4. Zhang, X.; Zhu, X. Autonomous path tracking control of intelligent electric vehicles based on lane detection and optimal preview method. *Expert Syst. Appl.* **2019**, *121*, 38–48.
5. Yoo, H.; Yang, U.; Sohn, K. Gradient-enhancing conversion for illumination-robust lane detection. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1083–1094.
6. Geiger, A.; Lauer, M.; Wojek, C.; Stiller, C.; Urtasun, R. 3d traffic scene understanding from movable platforms. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 1012–1025.
7. Peng, Y.; Gao, H. Lane detection method of statistical Hough transform based on gradient constraint. *J. Intell. Inf. Syst.* **2015**, *4*, 40–45.
8. Ma, C.; Xie, M. A method for lane detection based on color clustering. In Proceedings of the 2010 Third International Conference on Knowledge Discovery and Data Mining, Phuket, Thailand, 9–10 January 2010; pp. 200–203.
9. Son, J.; Yoo, H.; Kim, S.; Sohn, K. Real-time illumination invariant lane detection for lane departure warning system. *Expert Syst. Appl.* **2015**, *42*, 1816–1824.

10. Jung, H.; Min, J.; Kim, J. An efficient lane detection algorithm for lane departure detection. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23–26 June 2013; pp. 976–981.
11. Wang, J.G.; Lin, C.J.; Chen, S.M. Applying fuzzy method to vision-based lane detection and departure warning system. *Expert Syst. Appl.* **2010**, *37*, 113–126.
12. Song, W.; Yang, Y.; Fu, M.; Li, Y.; Wang, M. Lane detection and classification for forward collision warning system based on stereo vision. *IEEE Sens. J.* **2018**, *18*, 5151–5163.
13. Li, Q.; Zhou, J.; Li, B.; Guo, Y.; Xiao, J. Robust Lane-Detection Method for Low-Speed Environments. *Sensors* **2018**, *18*, 4274.
14. Wu, C.B.; Wang, L.H.; Wang, K.C.; Technology, S.F.V., Ultra-low Complexity Block-based Lane Detection and Departure Warning System. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 582–593.
15. Sun, Y.; Li, J.; Sun, Z. Multi-Stage Hough Space Calculation for Lane Markings Detection via IMU and Vision Fusion. *Sensors* **2019**, *19*, 2305.
16. Lee, C.; Moon, J.H. Robust lane detection and tracking for real-time applications. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 4043–4048.
17. Xing, Y.; Lv, C.; Wang, H.; Cao, D.; Velenis, E. Dynamic integration and online evaluation of vision-based lane detection algorithms. *IET Intell. Transp. Syst.* **2018**, *13*, 55–62.
18. Zhou, S.; Jiang, Y.; Xi, J.; Gong, J.; Xiong, G.; Chen, H. A novel lane detection based on geometrical model and gabor filter. In Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, San Diego, CA, USA, 21–24 June 2010; pp. 59–64.
19. Wang, J.; Mei, T.; Kong, B.; Wei, H. An approach of lane detection based on inverse perspective mapping. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 Octorber 2014; pp. 35–38.
20. Baofeng, W. Curve of Lane Line Detected method based on Linear Approximation. *J. Beijing Univ. Technol.* **2016**, *36*, 470–474.
21. Küçükmanisa, A.; Akbulut, O.; Urhan, O. Robust and real-time lane detection filter based on adaptive neuro-fuzzy inference system. *IET Image Process.* **2019**, *13*, 1181–1190.
22. Gupta, A.; Choudhary, A. A Framework for Camera-Based Real-Time Lane and Road Surface Marking Detection and Recognition. *IEEE Trans. Intell. Veh.* **2018**, *3*, 476–485.
23. Wang, H.; Wang, Y.; Zhao, X.; Wang, G.; Huang, H.; Zhang, J. Lane Detection of Curving Road for Structural High-way with Straight-curve Model on Vision. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5321–5330.
24. Ozgunalp, U. Robust lane-detection algorithm based on improved symmetrical local threshold for feature extraction and inverse perspective mapping. *IET Image Process.* **2019**, *13*, 975–982.
25. Cáceres Hernández, D.; Kurnianggoro, L.; Filonenko, A.; Jo, K. Real-time lane region detection using a combination of geometrical and image features. *Sensors* **2016**, *16*, 1935.
26. Cao, J.; Song, C.; Song, S.; Xiao, F.; Peng, S. Lane detection algorithm for intelligent vehicles in complex road conditions and dynamic environments. *Sensors* **2019**, *19*, 3166.
27. Yurtsever, E.; Yamazaki, S.; Miyajima, C.; Takeda, K.; Mori, M. Integrating driving behavior and traffic context through signal symbolization for data reduction and risky lane change detection. *IEEE Trans. Intell. Veh.* **2018**, 3.3, 242-253.
28. Li, J.; Mei, X.; Prokhorov, D.; Tao, D. Deep neural network for structural prediction and lane detection in traffic scene. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 690–703.
29. Nan, Z.; Wei, P.; Xu, L.; Zheng, N. Efficient lane boundary detection with spatial-temporal knowledge filtering. *Sensors* **2016**, *16*, 1276.
30. Ma, Y.; Havyarimana, V.; Bai, J.; Xiao, Z. Vision-Based Lane Detection and Lane-Marking Model Inference: A Three-Step Deep Learning Approach. In Proceedings of the 2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), Taipei, Taiwan, 26–28 December 2018; pp. 183–190.
31. Neven, D.; De Brabandere, B.; Georgoulis, S.; Proesmans, M.; Van Gool, L. Towards end-to-end lane detection: An instance segmentation approach. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Suzhou, China, 26–30 June 2018; pp. 286–291.
32. Hoang, T.M.; Baek, N.R.; Cho, S.W.; Kim, K.W.; Park, K.R. Road lane detection robust to shadows based on a fuzzy system using a visible light camera sensor. *Sensors* **2017**, *17*, 2475.
33. Chalapathy, R.; Chawla, S. Deep learning for anomaly detection: A survey. *arXiv* **2019**, arXiv:1901.03407.
34. Rashidha, R.; Simon, P., An adaptive-size median filter for impulse noise removal using neural network-based detector. *Int. J. Signal Imaging Syst. Eng.* **2016**, *9*, 305–310.

35. Kim, Y.; Soh, J.W.; Cho, N.I., Adaptively Tuning a Convolutional Neural Network by Gate Process for Image Denoising. *IEEE Access* **2019**, *7*, 63447–63456.
36. Kuang, X.; Sui, X.; Liu, Y.; Chen, Q.; Guohua, G., Single infrared image optical noise removal using a deep convolutional neural network. *IEEE Photonics J.* **2017**, *10*, 1–15.
37. Hansen, L.K.; Salamon, P. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 993–1001.
38. Melo, J.; Naftel, A.; Bernardino, A.; Santos-Victor, J. Detection and classification of highway lanes using vehicle motion trajectories. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 188–200.
39. Ren, J.; Chen, Y.; Xin, L.; Shi, J. Lane detection in video-based intelligent transportation monitoring via fast extracting and clustering of vehicle motion trajectories. *Math. Probl. Eng.* **2014**, *2014*, 156296.
40. Hermosillo-Reynoso, F.; Torres-Roman, D.; Santiago-Paz, J.; Ramirez-Pacheco, J. A Novel Algorithm Based on the Pixel-Entropy for Automatic Detection of Number of Lanes, Lane Centers, and Lane Division Lines Formation. *Entropy* **2018**, *20*, 725.
41. Yi, S.C.; Chen, Y.C.; Chang, C.H. A lane detection approach based on intelligent vision. *Comput. Electr. Eng.* **2015**, *42*, 23–29.
42. Bertozzi, M.; Broggi, A., GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE Trans. Image Process.* **1998**, *7*, 62–81.
43. López, A.; Serrat, J.; Canero, C.; Lumbreras, F.; Graf, T. Robust lane markings detection and road geometry computation. *Int. J. Automot. Technol.* **2010**, *11*, 395–407.
44. Guo, J.; Wei, Z.; Miao, D. Lane detection method based on improved RANSAC algorithm. In Proceedings of the 2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems, Taiwan, China, 25-27 March 2015; pp. 285–288.
45. Miller Frederic, P.; Vandome, A.F.; Mcbrewster, J. *Inverse Transform Sampling*; Alphascript Publishing: Saarbrücken, Germany, 2010; Volume 2, pp. 22–24.
46. Dorigo, M.; Blum, C. Ant colony optimization theory: A survey. *Theor. Comput. Sci.* **2005**, *344*, 243–278.
47. Srivastava, S.; Lumb, M.; Singal, R. Improved lane detection using hybrid median filter and modified Hough transform. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2014**, *4*, 30–37.
48. Aly, M. Real time detection of lane markers in urban streets. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 7–12.
49. Izquierdo, A.; Lopez-Guede, J.M.; Graña, M. Road Lane Landmark Extraction: A State-of-the-art Review. In Proceedings of the International Conference on Hybrid Artificial Intelligence Systems, León, Spain, 4–6 September 2019; pp. 625–635.