


ITC 4/50 Information Technology and Control Vol. 50 / No. 4 / 2021 pp. 722-735 DOI 10.5755/j01.itc.50.4.29094	Lane Line Extraction in Raining Weather Images by Ridge Edge Detection with Improved MSR and Hessian Matrix	
	Received 2021/05/14	Accepted after revision 2021/10/28
	 http://dx.doi.org/10.5755/j01.itc.50.4.29094	

HOW TO CITE: Wang, W., Berholm, F., Hu, K., Zhao, L., Feng, S., Tu, A., Fan, E. (2021). Lane Line Extraction in Raining Weather Images by Ridge Edge Detection with Improved MSR and Hessian Matrix. *Information Technology and Control*, 50(4), 722-735. <https://doi.org/10.5755/j01.itc.50.4.29094>

Lane Line Extraction in Raining Weather Images by Ridge Edge Detection with Improved MSR and Hessian Matrix

W. Wang

Department of Computer Science and Eng., Shaoxing University, Shaoxing, 312000, China
 School of Information Engineering, Chang'an University, Xi'an, 710064, China

F. Berholm

Royal Institute of Technology, Stockholm 10044, Sweden

K. Hu, L. Zhao, S. Feng, A. Tu and E. Fan

Department of Computer Science and Eng., Shaoxing University, Shaoxing, 312000, China

Corresponding author: W. Wang, E. Fang and L. Zhao: wangwx@usx.edu.cn

To accurately detect lane lines in road traffic images at raining weather, a edge detection based method is studied, which mainly includes four algorithms. (1) Firstly an image is enhanced by an improved Retinex algorithm; (2) Then, an algorithm based on the Hessian matrix is applied to strengthen lane lines; (3) To extract the feature points of a lane line, a ridge edge detection algorithm based on five line detection in four directions is proposed, in which, in light on the possible positions of lane lines in the image, it detects the maximum gray level points in the local area of the detecting point within the pre-set valid detection region; and (4) After the noise removal based on the minimum circumscribed rectangles, the candidate points of lane lines are connected as segments, and for the gap filling between segments, in order to make connection correctly, the algorithm makes the filling in two steps, short gap and long gap fillings, and the long gap filling is made on the combination of segment angle difference and gap distance and gap angle. By testing hundreds of images of the lane lines at raining weather and by comparing several traditional image enhancement and segmentation algorithms, the new method of the lane line detection can produce the satisfactory results.

KEYWORDS: Lane line detection, raining weather, Retinex, Hessian matrix, ridge edge, gap filling.

1. Introduction

With the continuous development and application of intelligent transportation system, Safety Driving Assist System has gradually come into public sight and become popular in people's life. In the process of normal driving, lane line tracking, lane line identification, or lane line departure warning have been the earliest components of image-based driver assistance systems. Since 1990s, they have been studied and implemented for the situations in the good viewing conditions and the clear lane markings on road. After then, the accuracy for particular situations, the robustness for a wide range of scenarios, time efficiency, and integration into higher-order tasks define visual lane detection and tracking as the continuing research subject [9].

At present, this kind of lane line marking detection algorithms based on Machine vision and Image processing can be divided into two categories: the first kind of algorithms are feature-based, which is the similar to the road detection in an aerial or a remote-sensing image [28] or the similar to the crack detection in a pavement image [27], such as crack detection by Steger and Hydrodynamics with improved Fractional differential [26], Crack detection in shadowed images on gray level deviations in a moving window [25]; and the second kind of algorithms are model-based [34]. The early warning model of lane departure decision based on taking the current position of driving vehicle in lane as the decision basis to analyze whether lane departure will occur. The position coordinates of the driving vehicle on road can be obtained by the detection algorithms [7].

In previous studies, there are many image processing based methods/algorithms reported: e.g., Image contrast enhancement [14], Object tracking [10] and Patches matching [19], so on. Zhou et al. [41] suggested a method: the longitudinal lane departure warning system is mainly utilized to detect lane departure collision caused by too fast speed or out of control direction, while the transverse lane departure warning system is mainly for monitoring lane departure collision caused by driver's inattention or driver's abandonment of steering operation. Shin et al. [20] reviewed past and current studies in this topic, and their review can help the researchers to plan future research and help readers in selecting suitable methods for their own targeted scenario. In 2016, Du et al. proposed a method for lane line detection, an edge detector is

used to extract the pixels on lane line firstly, then, after image smoothing, the sequential Random Sample Consensus is applied into a model fitting procedure to make sure each detected lane line is correct [4].

In the existing methods/algorithms, most of them applied Hough transform. Shu et al. [33] studied a driver assistant system, in which, a camera is mounted on the vehicle front window to detect the lane lines roughly, then the Hough transform is applied to identify the lane lines. Zhang et al. [35] studied a lane line tracking algorithm based on an improved Hough straight line transform. Fang et al. [6] made an improved algorithm based on the Hough straight line transform, firstly, the edges are detected and grouped in an image, then the Hough transform is adopted to identify the lane lines. Wei et al. [30] suggested a lane line detection method based on the constraint Hough transform double edge extraction, the lane line area extraction is based on the lane width feature and color feature, then, the Canny edge detector is utilized to obtain the lane line edges, the lane line features are extracted through the lane line edge and area information, and finally the straight lane lines are identified by a modified Hough transform. Deng and Wu [3] utilized a double lane line edge detection method in light on the constraint condition and Hough transform to detect lane lines. Xu et al. [32] studied a method for real-time performance of lane departure warning system in a complex environment, the fusion of the two algorithms is performed, firstly, the lane line background is extracted by using Tophat and Otsu algorithms respectively and then the results are fused by "AND" operation, and finally the probabilistic Hough transform is applied for straight line detecting. They stated that their method has an average detection accuracy of 96.07% and an average detection speed of 32 ms/frame. Winserng et al. [31], Suvarna et al. [21] also did the similar researches.

In last several years, the research based on deep learning and neural network has been a hot topic in different applications [37], such as Deep convolutional neural network for natural image classification [38], Discrete dynamics in nature and society, Multi-receptive field network for fast and accurate single image super-resolution [8] and Deep learning for automatic modulation classification [40], and so on. Some semantic segmentation methods are also applied for lane line detection, such as ENet network [18], CNN [17] and spatial CNN (SCNN) network [39] and Dee-

plab V3+ [1] and Deeplab V3+ network [13]. The ENet network is a network for online performance in the mobile scenarios. The SCNN network is designed for linear objects such as lane line scene. The Deeplab V3+ and Deeplab V3+ network are mainly compared with the attention structure.

Anyhow, the traffic accidents at raining weather are with heavy traffic flow and vehicle scratch [2], different theories are applied such as Dempster-Shafer theory [5], so it is important that the vehicles keep in the two-lane lines by a warning system which fast and accurately detecting lane lines can reduce traffic accidents.

In this paper, we concentrate on the lane line detection in the video images at raining weather. The method is studied mainly based on a new edge detection algorithm, including preprocessing and post processing algorithms. Firstly, the image is smoothed and enhanced, then, the center feature points in the lane lines are detected based on discontinuity characteristics, and finally the lane line segments are regulated and connected.

2. Image Enhancement on Retinex

In the raining weather, the traffic images are vague. To make image enhancement, several traditional image enhancement algorithms have been tested for our images, among them, the best one is the Retinex based algorithm [2], and the reason may be described as the follows.

When there are the lighting spaces or spectral changes, the visual quality of the vague image will have a great decrease. The visibility of image color and shadow details is very poor. While in the case of lighting, regardless of the changes in the spectrum of blue or red, and which will reduce the whole scene visibility of details and colors, the defect of the lighting is quite common. Also, for the scene with the white surface (like cloud or snow), the color and details of non-white areas of the image has the poor visibility. Therefore, we need a general algorithm to improve these vague images. The Retinex based algorithm is a good choice to enhance this kind of images.

In the two effects of dynamic range compression and tone recovery, the single-scale Retinex (SSR) algorithm can only improve the other at the cost of sacrificing one function. Therefore, a group of researchers such as Jobson et al. [11] proposed to combine the

enhancement results of different scales linearly to overcome the shortcomings of SSR model, a multi-scale Retinex (MSR) algorithm with local and global information taken into account. The main idea of this algorithm is to estimate the illumination component by combining the center function of several different scales and weighted average. The MSR algorithm can produce a single output image with good dynamic range compression, color stability and good tone recovery. This is why the SSR algorithm is extended into MSR algorithm. In last 20 years, the MSR is widely used for different videos and images [15].

The simple description of SSR algorithm is as follows:

$$R_m(x, y) = \log \left(\frac{I_m(x, y)}{F(x, y) * I_m(x, y)} \right) \quad (1)$$

$$G(x, y) = P e^{-(x^2+y^2)/\sigma^2}, \quad (2)$$

where, x, y are the image coordinates, $I_m(x, y)$ is the m th spectral band image at point (x, y) ; $R_m(x, y)$ is the Retinex result of the m th spectral band at point (x, y) ; $G(x, y)$ is a Gaussian function, σ is the constant of Gaussian around space, P is a scale, $*$ is the convolution symbol.

Due to the balance of the dynamic range compression and color reproduction, it is very essential to choose a good scale in SSR equation. If you do not want to sacrifice dynamic range compression or color reproduction, the MSR algorithm is a good solution [11], and it is a weighted combination of different scales SSR:

$$R_n = \sum_{i=1}^N \omega_i R_{in}, \quad (3)$$

where, N is the number of dimensions, R_n is the SSR result of the n th spectral band, R_{in} is the Retinex output component of the i th scale of the n th spectral band, and ω_i is the weight value for the n th scale. The most obvious problems about the MSR are to choose the appropriate number of dimensions, scale values and weights.

In our improved Retinex MSR algorithm, the Retinex scales are not constant. It takes different values in different regions, and the scale values are determined by the image depth. First of all, we get a dark channel image using a 15×15 mask, and calculate the transmission map $t(x, y)$. The value of $t(x, y)$ is between 0 to 1, and which represents it how far away from the camera.

In the transmission map, the dark region is far away from camera and the bright region is opposite. In our method, it is doing small scale transformation where far from the camera and doing large scale transformation near the camera. We assume that it has a linear relationship between the scale parameter $c(x, y)$ and the transmission map $t(x, y)$.

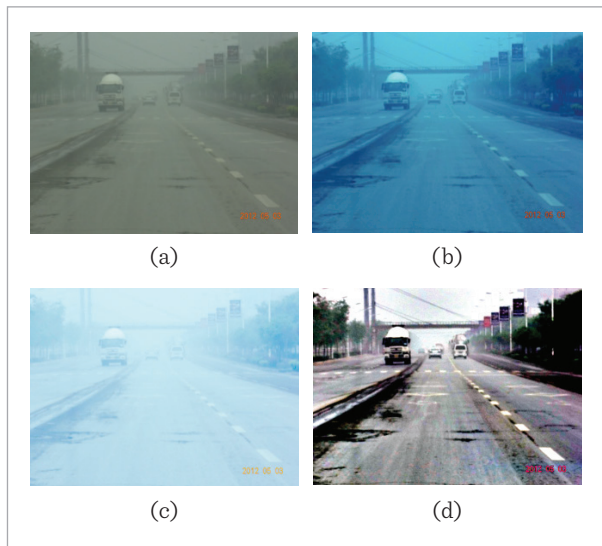
$$c(x, y) = \frac{t(x, y) - \min(t(x, y))}{\max(t(x, y)) - \min(t(x, y))} \times 100 + 10, \quad (4)$$

where, $\max(t(x, y))$ and $\min(t(x, y))$ are the maximum and minimum values of $t(x, y)$. Thus, we obtain the scale $c(x, y)$, and its range is 10 to 110, from a small scale to a large scale. According to the scale parameters obtained by Eq.(4), we do Gaussian filter to the haze image, and then calculate MSR result.

Figure 1 presents an example, the original image is a vague and dark one in a raining day as shown in (a); after the Linear stretching, the image contrast is not increased, and the right lane line in the bottom of the image is clearer than before, but the lane lines in the middle part are still vague, see (b); if the image in (a) is enhanced by the Log-transformation, the image light intensity seems to be increased, but the contrast is not good, and the lane line is more vague as shown in (c); and when the image in (a) is enhanced by MSR, the re-

Figure 1

Vague image enhancement with three different algorithms: (a) Vague image; (b) Linear stretching; (c) Log-transformation; and (d) Improved MSR



sult is much better, the visibility of the lane line and the front scene are much better than that in (a)-(c).

3. Lane Line Strengthen

After image enhancement, the image contrast and object visibility are increased, but we cannot make sure if the lane lines are strengthened. Hence, an algorithm based on Hessian Matrix is applied for lane line strengthening, which should include three basic parts: image smooth for denoising, contrast stretch and lane line strengthening [24], to do these, the following Hessian matrix-based algorithm [16] is adopted. The basic theory of the algorithm may be described as follows [36].

For an array $I : \Omega \rightarrow R, \Omega \in R^2$, a Gaussian kernel $g(p; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$ is applied firstly, then, the Hessian matrix corresponding to I at point $p(x, y)$ is set as:

$$H_{\sigma}(p) = \begin{pmatrix} g_{xx}(p) & g_{xy}(p) \\ g_{xy}(p) & g_{yy}(p) \end{pmatrix} * I(p) \quad (5)$$

The definition of lane lines is as follows:

$$T_d(p, \theta; \sigma) = t_d(p, \theta; \sigma) * I(p), \quad (6)$$

where, $t_d(p, \theta; \sigma) = g_{xx} \cos^2 \theta + g_{yy} \sin^2 \theta + g_{xy} \sin 2\theta$.

The forward filter $t_f(p; \sigma, \psi_1)$ and the backward filter $t_b(p; \sigma, \psi_2)$ are:

$$t_f(p; \sigma, \psi_1) = t_d(x + d \cos(\theta + \psi_1), y + d \sin(\theta + \psi_1)) \quad (7)$$

$$t_b(p; \sigma, \psi_2) = t_d(x - d \cos(\theta + \psi_2), y - d \sin(\theta + \psi_2)), \quad (8)$$

where, ψ_1, ψ_2 are the angels adopted for detecting the evidence of lane lines in the neighbor pixels. d is an offset parameter which is set at an appropriate value. It is difficult to contribute enough information when d value is low. It can cause the incorrect segmentation for judging the spurious lane line pixels into the real lane lines.

As the response, the two oriented filters are given by $T_f(p; \sigma, \psi_1) = t_f(p; \sigma, \psi_1) * I(p)$ and $T_b(p; \sigma, \psi_2) = t_b(p; \sigma, \psi_2) * I(p)$ respectively, and the enhanced lane lines are in Eq. (9)

$$T^*(p; \sigma) = T_d^*(p) + T_b^*(p) + T_f^*(p), \quad (9)$$

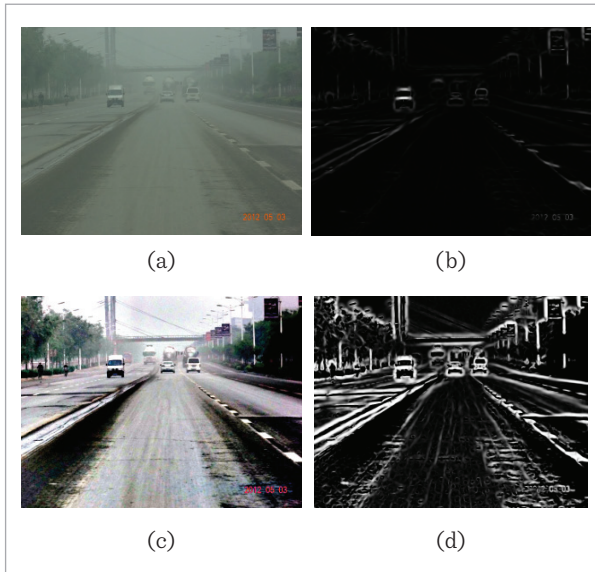
where,

$$T_d^*(p) = \max_{\theta} T_d(p), T_b^*(p) = \max_{\psi_1} T_b(p), T_f^*(p) = \max_{\psi_2} T_f(p).$$

Then we search for the max response at the multi-orientations as the output of image. Figure 2 presents the lane line enhancement by the Hessian matrix based algorithm, if the algorithm directly operates on the original vague image (a), the result image is dark, and the lane line visibility is low (b), and after operating the above MSR algorithm (c), the Hessian matrix based algorithm can give out the high visibility of lane lines (d).

Figure 2

Lane line enhancement by Hessian matrix-based algorithm: (a) Original image; (b) Hessian on (a); (c) improved MSR; and (d) Hessian on (c)



4. Lane Line Extraction

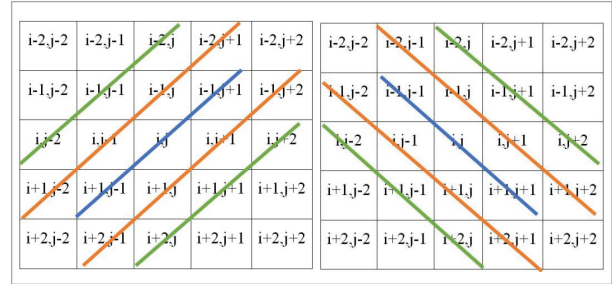
In the traffic image at a raining day, since the gray levels of lane lines are not uniform, the more the lane line points we detect, the lines can be more easily identified. In order to collect as many lane line ridge points as possible, the algorithm is designed as the follows.

For each detecting point in the gray level image $f(i, j)$, in its 5x5 neighborhood, calculate and judge whether the point is a ridge point according to the four directions of 0° , 90° , 45° and 135° , and the weight of each point on the segment line in Figure 3 is inversely pro-

portional to the distance from the detecting point. If the condition is satisfied, it is determined to be a candidate ridge point. As examples, the 5x5 templates in 45° and 135° directions are shown in Figure 3.

Figure 3

Ridge point detection templates on 45° and 135°



In each of the above template, there are five lines for judging if the diagonal middle line (blue color) is a ridge line, then, the detecting point may be a candidate of the ridge points. To make the judgment, the two lines above it and two lines under it are also used for the detection. In a line, each grid represents a pixel, and the label in the grid represents the position of the pixel in the image.

In the same way, the ridge lines in the directions of 0° and 90° can also be determined.

The new algorithm is different to a normal ridge edge detection algorithm, in order to reduce the influence of isolated point noise, the gray level of a pixel is replaced by the weighted average value of a short straight line, that is, the templates in 0° , 45° , 90° and 135° directions are applied to judge if the detecting point is a ridge point candidate. Because a lane line segment is composed of straight line or a curve, it is reasonable to use a short segment instead of a point to detect the lane line segment.

0° direction:

$$\left\{ \begin{array}{l} \sum_{p=-1}^1 f(i, j+p) - \sum_{p=-1}^1 f(i-1, j+p) \geq 0 \Rightarrow F_1 \\ \sum_{p=-1}^1 f(i, j+p) - \sum_{p=-1}^1 f(i+1, j+p) \geq 0 \Rightarrow F_2 \\ \sum_{p=-1}^1 f(i-1, j+p) - \sum_{p=-1}^1 f(i-2, j+p) \geq 0 \Rightarrow F_3 \\ \sum_{p=-1}^1 f(i+1, j+p) - \sum_{p=-1}^1 f(i+2, j+p) \geq 0 \Rightarrow F_4 \end{array} \right. \quad (10)$$

45° direction:

$$\begin{cases} L = [f(i-1, j+1) + f(i, j) + f(i+1, j-1)]/3 \\ L_1 = [f(i+1, j-2) + f(i, j-1) + f(i-1, j) + f(i-2, j+1)]/4 \\ L_2 = [f(i, j-2) + f(i-1, j-1) + f(i-2, j)]/3 \\ L_3 = [f(i+2, j-1) + f(i+1, j) + f(i, j+1) + f(i-1, j+2)]/4 \\ L_4 = [f(i+2, j) + f(i+1, j+1) + f(i, j+2)]/3 \\ L-L_1 \geq 0 \Rightarrow F_1; \quad L-L_2 \geq 0 \Rightarrow F_2; \quad L_1-L_2 \geq 0 \Rightarrow F_3; \quad L_3-L_4 \geq 0 \Rightarrow F_4 \end{cases} \quad (11)$$

90° direction

$$\begin{cases} \sum_{p=-1}^1 f(i+p, j) - \sum_{p=-1}^1 f(i+p, j-1) \geq 0 \Rightarrow F_1 \\ \sum_{p=-1}^1 f(i+p, j) - \sum_{p=-1}^1 f(i+p, j+1) \geq 0 \Rightarrow F_2 \\ \sum_{p=-1}^1 f(i+p, j-1) - \sum_{p=-1}^1 f(i+p, j-2) \geq 0 \Rightarrow F_3 \\ \sum_{p=-1}^1 f(i+p, j+1) - \sum_{p=-1}^1 f(i+p, j+2) \geq 0 \Rightarrow F_4 \end{cases} \quad (12)$$

135° direction:

$$\begin{cases} L = [f(i-1, j-1) + f(i, j) + f(i+1, j+1)]/3 \\ L_1 = [f(i-2, j-1) + f(i-1, j) + f(i, j+1) + f(i+1, j+2)]/4 \\ L_2 = [f(i-2, j) + f(i-1, j+1) + f(i, j+2)]/3 \\ L_3 = [f(i-1, j-2) + f(i, j-1) + f(i+1, j) + f(i+2, j+1)]/4 \\ L_4 = [f(i+2, j) + f(i+1, j-1) + f(i, j-2)]/3 \\ L-L_1 \geq 0 \Rightarrow F_1; \quad L-L_2 \geq 0 \Rightarrow F_2; \quad L_1-L_2 \geq 0 \Rightarrow F_3; \quad L_3-L_4 \geq 0 \Rightarrow F_4 \end{cases} \quad (13)$$

For each pixel in the image, in a 5x5 detecting template, we utilize five lines to detect the ridge point, rather than based on original gray levels of the detection point and its neighboring pixels, in this way, we can reduce many false detected ridge points and the post-processing burden. In an image $f(i, j)$, for each pixel, the equations for the detection in the four directions are as follows.

In each of the above directions, if in the four directions we have one of F_1, F_2, F_3, F_4 , the detecting point is a ridge point, otherwise it is not a ridge point in the corresponding direction. If the detecting point is a ridge point not only in one of the four directions, we choose maximum edge value of the detecting point as its ridge value.

For example, in the vertical 90° direction, we have two values (at the top region and the bottom region), and call them as y_{+90} and y_{-90} .

if $y_{+90}(F_1, F_3) > 0$ and $y_{-90}(F_2, F_4) > 0$, we have $y_{90} = y_{+90} + y_{-90} = F_1 + F_2 + F_3 + F_4$. In the same way, we calculate the other directional y values. For example, if the detecting point has the ridge values in all the four directions, we compute it as:

$$z = \max(y_0, y_{45}, y_{90}, y_{135}). \quad (14)$$

The ridge point value is:

$$g(i, j) = z \quad (15)$$

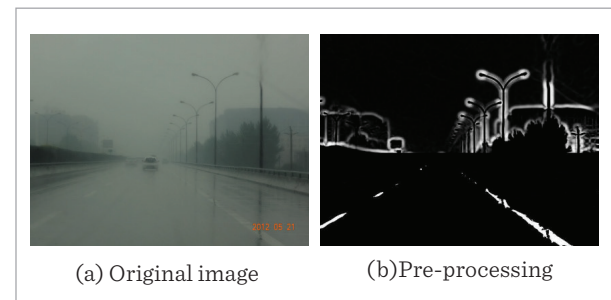
If we output a binary image directly, when we set a threshold T , we can do:

$$\text{If } z > T, g(i, j) = 255, \text{ otherwise } g(i, j) = 0. \quad (16)$$

It is normal that an original image includes a lot of noises which will affect ridge edge detection result. One simple way for reducing the noises is to use a smoothing filter such as the Gaussian smoothing function, which has a width parameter sigma σ , often referred to as the scale space parameter. The choice of σ depends on white spot size distribution. Figure 4 gives the result of the new algorithm for the image of two-lane lines at a raining day. After image pre-processing as described before, the preprocessed images include the both lane line regions and non-lane line regions. In the lane line regions, the lane lines may be not continuous, and the widths are different in different short segments, which need the further processing and image segmentation. The non-lane line regions are not in the pre-estimated areas, so they are easily removed from the image. Hence, in the next Section, we will describe the further image processing and segmentation.

Figure 4

Result of image preprocessing



5. Lane Skeleton and Gap Filling

After lane line segments are identified, the remaining work is to fill gaps between the segments, before the filling, the segments should thin as one pixel width, which can be the basis for endpoint detection and segment angle calculation.

5.1. Lane Skeleton Extraction

Commonly used skeleton extraction algorithms are based on Axis transformation and thinning. Thinning is based on the grass burning model, that is, through the evolution of the object from the outside to the inside to get the axis trajectory. The main idea of the algorithm is to peel the boundary evenly again and again, and the result is that the boundary is the inner-most part (otherwise it will affect the connectivity) to become the skeleton. Axis transformation algorithm [29] simulates the process of the burning grass, that is to say, the medial axis locus can be obtained by applying the object's gradual evolution from edge to center. The main idea is to uniformly peel the edges layer by layer, and consequently, the rest of the inner-most layer can be in skeleton.

In order to compare the advantages and disadvantages of the two skeleton extraction algorithms more precisely, we added the preprocessing function to reduce the burr generation, and added the post-processing function to fill the short gaps on the skeleton line, so as to obtain the insect skeleton detection results in Figure 5. The results show that the post-processing algorithm of the center axis transformation makes the skeleton line more continuous. Although the Thinning algorithm after pretreatment reduces many burrs, it still has obvious burrs.

For lane lines, what we want to connect or sew is the gap between the ends of skeleton line segments. If there are more intersections, it will be difficult to connect, so we hope to get skeleton line segments with no or very few intersections. According to the two skeleton algorithms in Figure 5, we detect and calibrate the endpoint (red square) and intersection (green square), as shown in Figure 6.

As shown in Figure 6, there are 11 segments with different lengths in the skeleton of the middle axis transformation, and there are few intersections (29 endpoints, 5 of the 7 intersections overlap with the endpoints) in general, while there are many intersec-

Figure 5

Extracted skeletons by using two types of algorithms

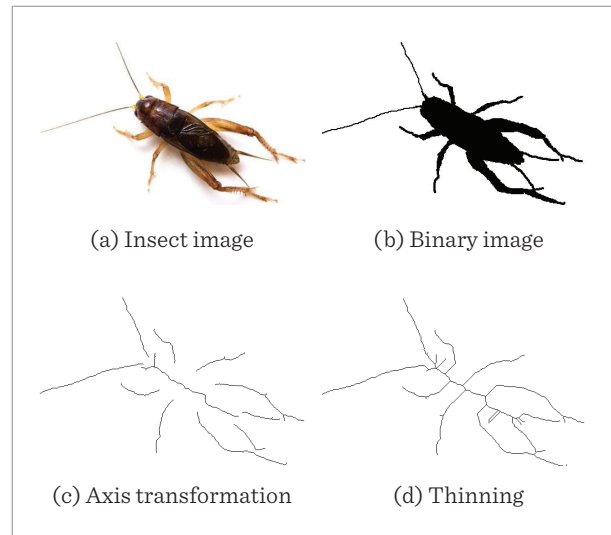
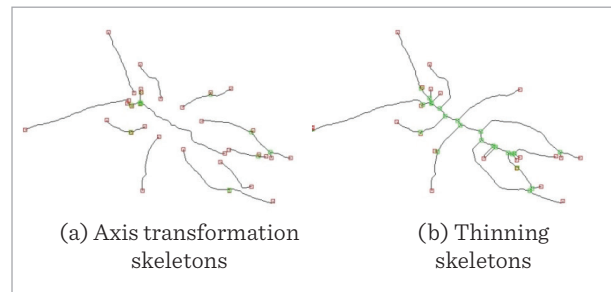


Figure 6

End point and junction point markings of two kinds of skeletons in Figure 5

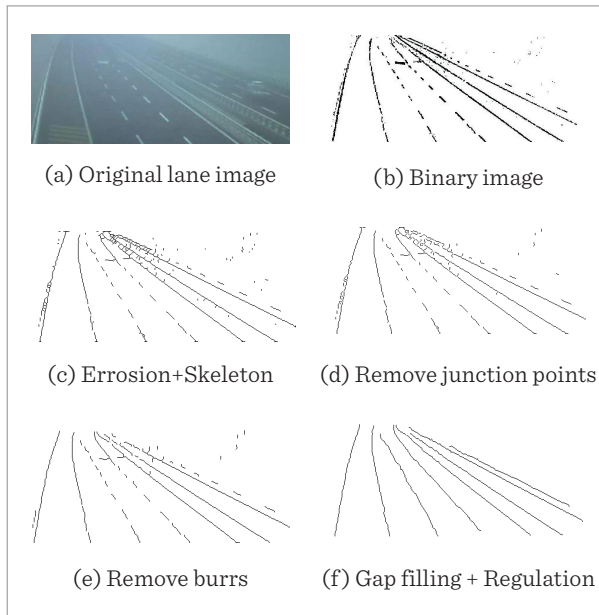


tions on the skeleton line of the thin line transformation, especially on the insect body skeleton and its connection with the limb skeleton, It's the skeleton of an insect's broad body (excluding limbs and antennae). Therefore, in most cases, because the lane lines obtained by gray level or color information are often line targets of different thickness, if the skeleton algorithm is not selected well, it will produce a lot of unnecessary branch lines and burrs, resulting in a lot of intersection points, which will cause difficulties to the subsequent skeleton line connection. However, no matter what kind of skeleton extraction algorithm is selected, the disadvantages of the above two algorithms will exist, so it is necessary to carry out the necessary post-processing.

As above comparison and analysis, Axis transformation algorithm is applied in this study, but it is added post processing functions. For a skeleton point, the basic principle is: in the eight neighborhoods of a detecting point, if there is only a skeleton point as an endpoint, it is a skeleton point; if there are two skeleton points as nodes, it is also the skeleton point; and if there are more than three skeleton points as branch ones, it is the skeleton point too.

To make a segment as one line or one curve, the short branches or burrs should be removed. The task of burr removal is to search for these points and to remove short lines. Firstly, the burr length threshold is set as B and the gap length threshold is A , the burr removal algorithm scans the whole skeletons and detects N skeleton endpoints and M skeleton junction points, remove all the M junction points. Secondly, it considers every endpoint as the detection point and searches for its most closing neighboring endpoint, if the gap length between the two endpoints is less than A , the two points are connected by a straight line. Thirdly, check all the skeleton lengths, if the length is shorter than B , the skeleton line can be removed. Fourthly, set $A = A + A/2, B = B + B/3$, repeat the above procedure. Finally, smooth all of the skeletons by a least square regression algorithm. Figure 7 shows one of examples for lane line detection and regulation.

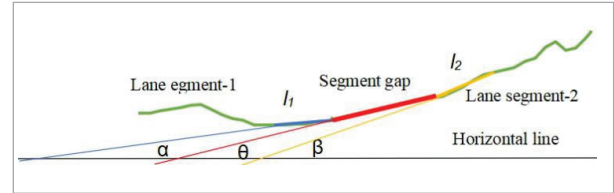
Figure 7
Lane line skeleton detection and lane line regulation



5.2. Gap Filling

The gap length between two neighboring segments d (Figure 8) is measured first, we set two thresholds $d_1 = d_2 / 2$.

Figure 8
Two lane line segments and their parameters



If $d \leq d_1$, the two segments can be connected directly by a straight line;

If $d > d_2$, the two segments cannot be connected;

If $d_1 < d \leq d_2$, whether the two segments should be connected mainly depends on gap distance and segment angles, we set the distance factor:

$$1/2 < D_1 = d_1 / d < 1, \tag{17}$$

where, D_1 is proportional to $1/d$, the greater d is, the smaller D_1 is, i.e., the probability of segment connection is low.

For the angle factor, we set:

$$1/2 < D_2 = 1 / (1 + \tan(|\alpha - \beta|)) < 1, \tag{18}$$

where, $|\alpha - \beta| < \pi/2$; the segment angle is computed as a straight line of 7 points by using least squares regression algorithm. α and β are two neighboring segments' angles as shown in Figure 6, i.e., the factor D_2 is proportional to $1/\tan(|\alpha - \beta|)$, the greater $\tan(|\alpha - \beta|)$ is, the smaller D_2 is, i.e., the probability of segment connection is low.

We calculate D by referring to Figure 6 as the follows:

$$D = (\lambda_1 D_1 + \lambda_2 D_2). \tag{19}$$

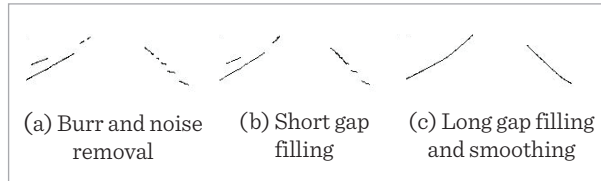
If gap line angle θ (Figure 6) meets $\max(|\theta - \alpha|, |\theta - \beta|) \leq \tau < \pi/2$ and $D \leq D_0$, then connect the two segments by a straight line.

For the above parameters, in our case, $d_1 = 7$ and $d_2 = 14$ pixels; the weight parameters $\lambda_1 + \lambda_2 = 1$, their value selection depends the applications, if the dis-

tance is more important than angle difference, we can choose $\lambda_1 > \lambda_2$, otherwise $\lambda_1 < \lambda_2$, in this application, $\lambda_1 = 0.6$; $\lambda_2 = 0.4$ $\theta = 35^\circ$; $D_0 = 0.6$; and $\tau = 45^\circ$. One example is shown in Figure 9.

Figure 9

Lane line detection after burr and noise removal



6. Experiments and Analyses

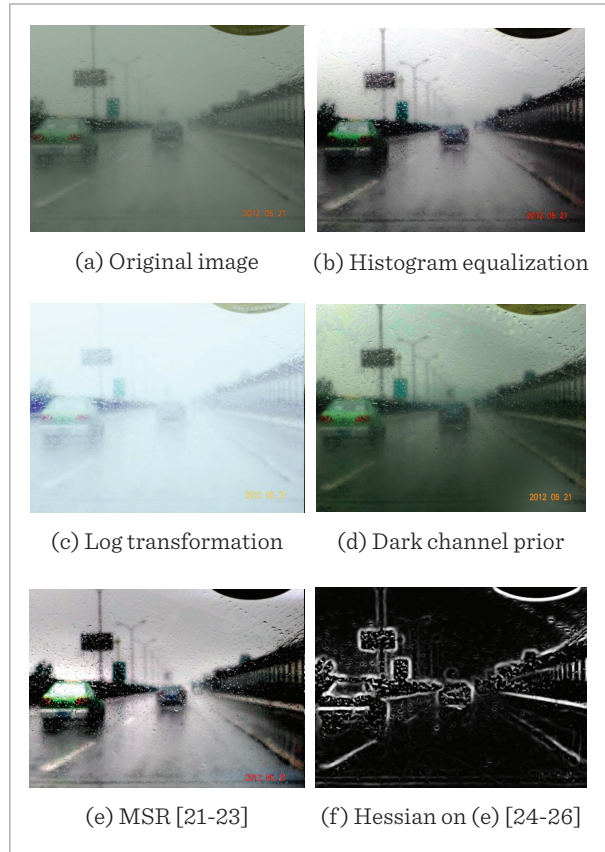
As above description, the lane line detection at raining weather is harder, hence we study a new method for the lane line extraction. The method mainly includes two parts: image pre-processing and lane line detection. The former is for image enchanting and lane line strengthening, and the latter is for lane line feature point detection in a gray level image and segment processing in a binary image. In the following, we present several typical lane line images for validating the studied method, and compare the new method to the existing basic image processing algorithms and recent semantic methods.

6.1. The Method for Typical Lane Line Images

In Figure 10, the original image (a) is a traffic image taken in a raining day, the image is blurring, objects are vague, a small part of lane line on left can be seen clearly, the rest part of left lane line and the whole lane line on right part cannot be seen. After Histogram equalization, the image seems to be clearer than before, but it is still vague as shown in (b). To increase the light intensity, the Log transformation is applied on the original image, the lightning intensity is increased, but the contrast is still low (c). When the Dark channel prior algorithm is applied for the image enhancement, the color information may be increased, but the image is still vague and lane lines cannot be shown up clearly (d). If MSR is utilized, the image contrast is increased, and lane line visibility is improved (e). After Hessian matrix-based algorithm is operated on (e), the satisfactory result can be obtained, as shown in (f).

Figure 10

Enhancement algorithm comparison for the lane line image at raining day



For lane detection part, the main detection procedure and the comparison to other algorithms are presented in Figure 11. When the noise objects are removed in the valid detection region, the lane line feature points are obtained as shown in (a), then, it makes object thinning (b), detects endpoints, and makes short gap filling (c), finally the long gaps are filled as shown in (d), where the two-lane lines are well detected.

In Figure 12, the original image is a fog image caused by raining (a), the right lane line is very fuzzy, and the front part of the left lane is vague. To increase the visibility of lane lines, the improved MSR performs on the image, the result image has a higher contrast, and the lane lines are clearer (b). Before lane line detection, the pre-set valid detection region is cut off between the two red lines (b), as shown in (c). If we directly binarize the cut-image, it will include a lot of noises and

Figure 11

Lane line detection in a binary image

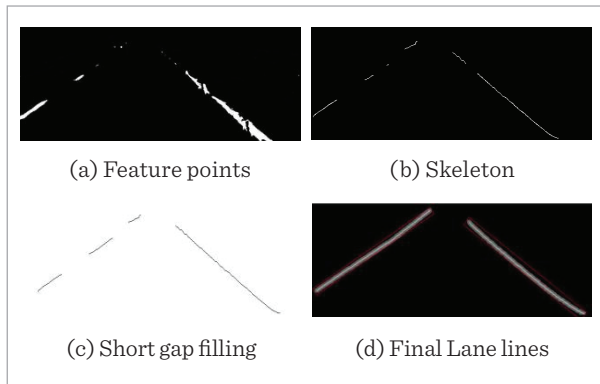
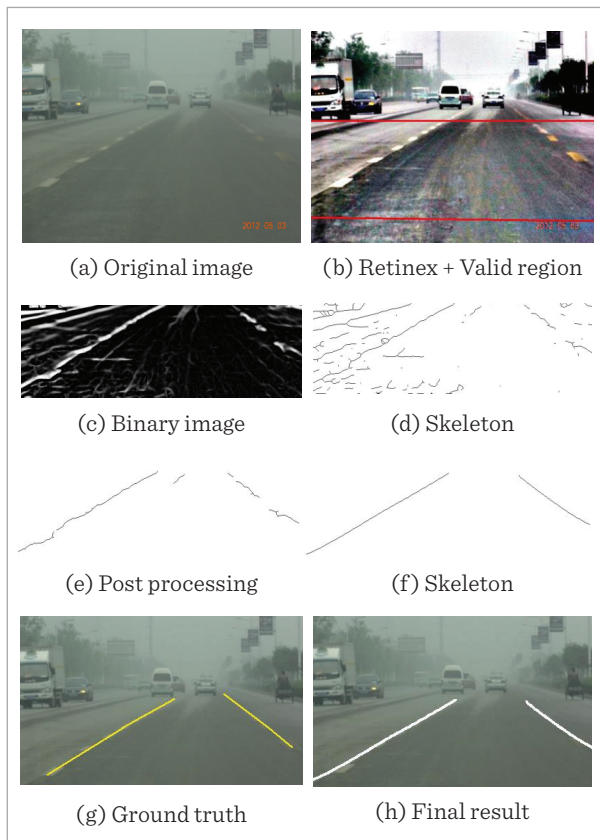


Figure 12

Lane line detection procedure

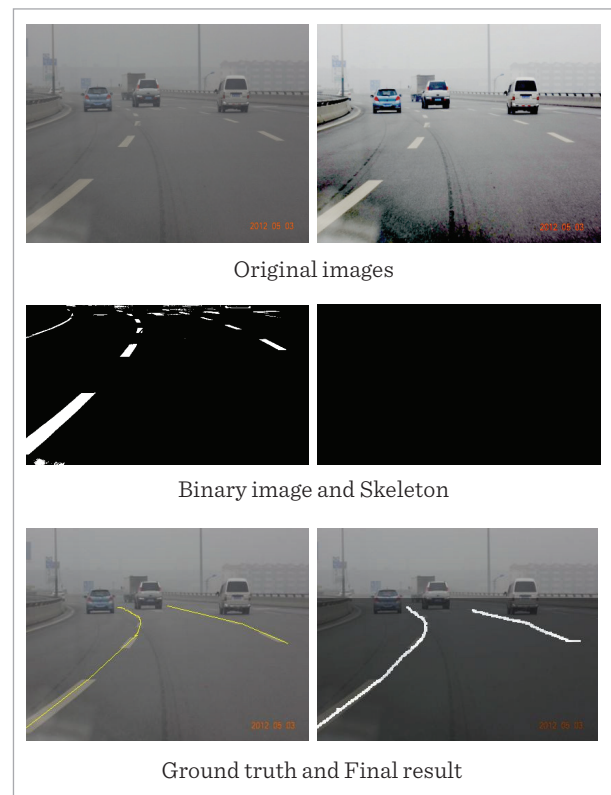


non-lane objects, even we do the thinning operation on the binary image, there are a lot of noise lines (d). Hence, we make the minimum circumscribed rectangle (MCR) [22] for each of the objects, according

the rectangles' elongations, and orientations, for instance, the orientation should be meet $25^{\circ} \leq |\theta| \leq 65^{\circ}$, elongation should be $(\text{length}/\text{width}) > 3$, to remove noise objects. According to these conditions, the most of non-lane line objects can be deleted, then, the thinning result is presented in (e). After noise line removal and gap filling, we can obtain the result image, since lane lines should be smoothed lines or curves, and we use the least square regression method to smooth the lane lines as shown in (f). When we overlay (e) on the original image, the result is in (h) which is almost the same as the ground truth in (g).

Figure 13

Curved lane line detection examples



The studied method is not only available for the lane line detection at a raining day (Figure 12), and it is also available for curved lane line detection (Figure 13), which is difficult for the Hough transform based algorithms. In Figure 13, the left lane line (a) is an obvious curved line, and the lane line visibility can also be increased by our pre-processing algorithms as shown in (b). After the feature point detection (d), then, the thin-

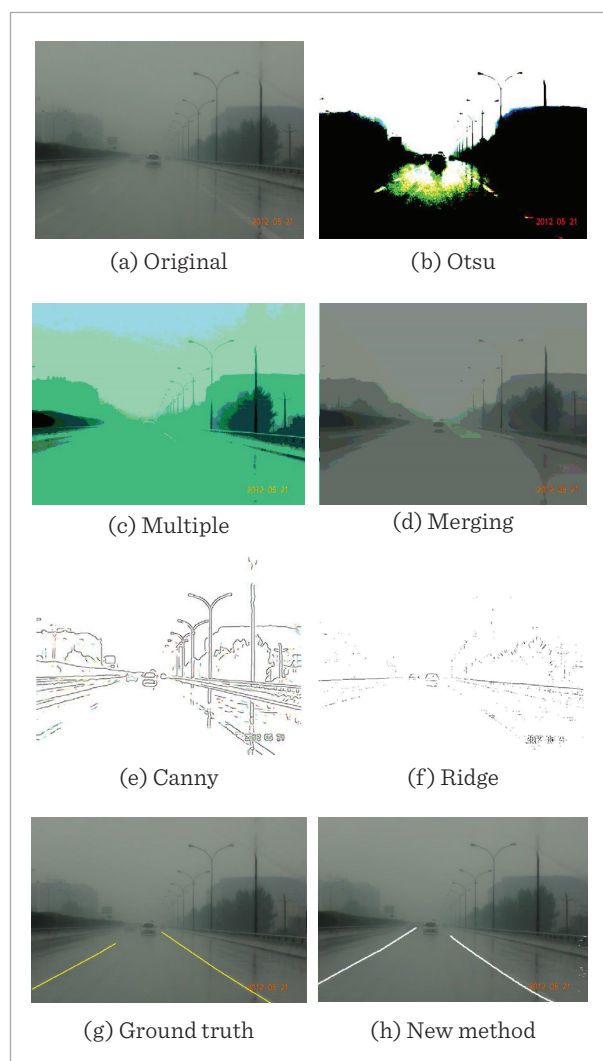
ning (g), endpoint detection and gap filling (h), we can have final lane line detection result in (i).

6.2. Comparison Between Different Methods

For the image processing algorithms for lane line detection, the algorithm comparison is shown in Figure 14, a lane line image was taken in the raining weather, the image includes buildings, cars, trees and other objects, and the road is slippery and has light reflection, hence, the image is vague. We tested both the similarity-based algorithms and the discontinuity-based algorithms.

Figure 14

Comparison of basic image processing algorithms



The similarity-based algorithms such as Otsu thresholding (b), Multiple thresholding (c) and Region Merging (d) can only separate the image into different large areas, but the lane lines are missing. The discontinuity-based algorithms such as Canny (e) and traditional Ridge edge detection (f) are also hard to detect out lane lines. The new algorithm can identify the lane lines well (h), which is the same as the ground truth (g).

In addition to the image segmentation algorithms, as the semantic method development, it has been used into lane line detection: it is good for the images from the normal weather, but not satisfactory for the images from the worse weather or worse environment such as raining days, night time, fog and haze days, etc.

Table 1 shows the results by using current different semantic segmentation methods and the method studied in this paper. The new method has the advantage for lane line detection, and the processing results (see Figures 12-14) are the same as ground truth. In the normal weather, except for Enet [18] which gives low accuracy, SCNN [17], Deeplab V3+ [1] and Deeplab V3+ network [13] can give the accuracy over 93%, but in the worse weather, their highest accuracy is only 68.1%. The new method studied in this paper has 95.1% accuracy for the normal weather, 70.1% accuracy for the worse weather for the 200 images' testing, and it means that in the limited image testing, it is better than the recent semantic segmentation methods in the detection accuracy.

Table 1

Comparison of semantic methods

Data set	Method	Recall rate (%)	Accuracy (%)
Normal weather	Enet [26]	75.5	73.4
	SCNN [27]	95.2	93.2
	Deeplab V3+ [29]	95.4	93.4
	Method [30]	95.8	94.8
Worse weather	Our method	94.7	95.1
	Enet [26]	40.5	38.0
	SCNN [27]	70.5	68.1
	Deeplab V3+ [29]	71.2	65.0
	Method [30]	65.5	62.4
	Our method	72.5	70.1

7. Conclusion

The research content mainly includes four aspects:

- 1 To enhance the vague lane line image caused by raining, an improved Retinex MSR based algorithm is applied, in which the scale in different region is determined by the local image depth.
- 2 Due to the low contrast and high noise of lane line images acquired at the raining day, a lane line strengthening algorithm is designed based on Hessian matrix. The algorithm can strengthen the lane lines after the image enhancement.
- 3 An algorithm is studied to detect the lane line ridge edge feature points, based on the characteristics of the possible position of the lane lines, the algorithm detects the ridge edge points in four directions, and the detection based on five lines in each of the directions.
- 4 After noise and burr removal, thinning and end-point detection, the gaps between segments are filled in two steps, which can be carried out based on segment angles and gap distance and gap angle. It is different to the traditional algorithm which either considers gap distance or considers the difference between two neighboring segments' angles, the new algorithm considers the two-parameter combination by weights and gap angle.

In the process of this study, hundreds of video images of lane lines on freeway at the raining weather are tested with the above method, and the different traditional image enhancement algorithms and line detection algorithms are compared to the new method. The experimental results and algorithm comparison show that the new method (includes several algorithms) proposed in this study can be applied for the automatic detection of the lane lines in the highway at the raining weather and can also be applied for the curved lane line detection. Further research focuses on the extension 2D to 3D [23] in other complex traffic situations.

Acknowledgments

This research is financially supported by Natural Science Foundation of Zhejiang Province (Grant No. TY22F025548, Zhejiang Provincial Natural Science Foundation of China (Grant No.LY20F020011), and Key scientific research project of Shaoxing University (Grant No.2020LG1004).

References

1. Bhme, L., Madesta, F., Sentker, T., Werner, R., Combining Good Old Random Forest and DeepLabv3+ for ISLES, 2018 CT-Based Stroke Segmentation [M], 2019. https://doi.org/10.1007/978-3-030-11723-8_34
2. Brown, M., Windridge, D., Guillemaut, J. Y. A Generalisable Framework for Saliency-Based Line Segment Detection. *Pattern Recognition*, 2015, 48(12), 3993-4011. <https://doi.org/10.1016/j.patcog.2015.06.015>
3. Deng, G., Wu, Y. Double Lane Line Edge Detection Method Based on Constraint Conditions Hough Transform. *IEEE17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES) (2018)*. <https://doi.org/10.1109/DCABES.2018.00037>
4. Du, X., Tan, K. Vision-Based Approach Towards Lane Line Detection and Vehicle Localization. *Machine Vision and Applications*, 2016, 27, 175-191. <https://doi.org/10.1007/s00138-015-0735-5>
5. Fan, L., Fan, E., Yuan, C., Hu, K. Weighted Fuzzy Track Association Method Based on Dempster-Shafer Theory in Distributed Sensor Networks. *International Journal of Distributed Sensor Networks*, 2016, 12(7), 10 Pages. <https://doi.org/10.1177/1550147716658599>
6. Fang, Z., Sheng, L., Kang, S., Yan, S., Wang, M. Improved Lane Line Detection Algorithm Based on Hough Transform. *Pattern Recognition and Image Analysis*, 2018, 28(2), 254-260. <https://doi.org/10.1134/S1054661818020049>
7. Gopalan, R., Hong, T., Shneier, M., Chellappa, R. A Learning Approach Towards Detection and Tracking of Lane Markings. *IEEE Transactions on Intelligent Transportation Systems*, 2012, 13(3), 1088-1098. <https://doi.org/10.1109/TITS.2012.2184756>
8. He, Z., Cao, Y., Du, L. Xu, B., Yang, J., Cao, Y., Tang, S., Zhuang, Y. MRFN: Multi-Receptive-Field Network for Fast and Accurate Single Image Super-Resolution. *IEEE*

- Transactions on Multimedia, 2020, 22(4), 1042-1054, 2020. <https://doi.org/10.1109/TMM.2019.2937688>
9. Hillel, A. B., Lerner, R., Levi, D., Raz, G. Recent Progress in Road and Lane Detection: A Survey. *Machine Vision and Applications*, 2014, 25(3), 727-745. <https://doi.org/10.1007/s00138-011-0404-2>
 10. Hu, K., Ye, J., Fan, E., Shen, S., Huang, L., Pi, J. A Novel Object Tracking Algorithm by Fusing Color and Depth Information Based on Single Valued Neutrosophic Cross-Entropy. *Journal of Intelligent & Fuzzy Systems*, 2017, 32(3), 1775-1786. <https://doi.org/10.3233/JIFS-152381>
 11. Jobson, D. J., Rahman, Z. U., Woodell, G. Properties and Performance of a Center/Surround Retinex. *IEEE Transactions on Image Processing*, 1997, 6, 451-462. <https://doi.org/10.1109/83.557356>
 12. Land, E. The Retinex Theory of Color Vision, *Scientific American*, 1977(6). <https://doi.org/10.1038/scientificamerican1277-108>
 13. Li, H. Lane Line Detection and Tracking Method based on DeepLab-LatNet Network and Density Clustering (Master thesis), 2020, Chang'an University, Xi'an, China. https://doi.org/10.1007/978-981-15-3753-0_47
 14. Liu, S., Rahman, M. A., Lin, C. F., Wong, C. Y., Jiang, G., Liu, S., Kwok, N., Shi, H. Image Contrast Enhancement Based on Intensity Expansion-Compression. *Journal of Visual Communication and Image Representation*, 2017, 48, 169-181. <https://doi.org/10.1016/j.jvcir.2017.05.011>
 15. Masaya Y., Yasunori S. Image Enhancement Using Retinex and Image Fusion Techniques. *Electronics and Communications in Japan*, 2018, 101(8), 52-63. <https://doi.org/10.1002/ecj.12092>
 16. Moritz S. D., Martin M. Improved Automatic Computation of Hessian Matrix Spectral Bounds. *SIAM Journal on Scientific Computing*, July 2015, 38(4). <https://doi.org/10.1137/15M1025773>
 17. Pan, X., Shi, J., Luo, P., Wang, X. Spatial as Deep: Spatial CNN for Traffic Scene Understanding, 2018, arXiv, 1712.06080.
 18. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E., ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. arXiv, 1606.02147.
 19. Pi, J., Hu, K., Zhang, X., Gu, Y., Zhan, Y. Robust Object Tracking with Compressive Sensing and Patches Matching. *IEICE TRANSACTIONS on Information and Systems*, 2016, 99(6), 1720-1723. <https://doi.org/10.1587/transinf.2015EDL8235>
 20. Shin, B., Xu, Z., Klette, R. Visual Lane Analysis and Higher-Order Tasks: A Concise Review. *Machine Vision and Applications*, 2014, 25, 1519-1547. <https://link.springer.com/article/10.1007/s00138-014-0611-8> - citeas. <https://doi.org/10.1007/s00138-014-0611-8>
 21. Suvarna, S., Udayakumar, R. Lane Datasets for Lane Detection. Conference: 2019 International Conference on Communication and Signal Processing (ICCSP), <https://doi.org/10.1109/ICCSP.2019.8698065>
 22. Wang, W. Image Analysis of Particles by Modified Ferret Method-Best-Fit Rectangle. *Powder Technology*, 2006, 165(1), 1-10. <https://doi.org/10.1016/j.powtec.2006.03.017>
 23. Wang, W., Chen, W., Wang, K., Li, S. Extraction of Tunnel Centerline and Cross Sections on Fractional Calculus and 3D Invariant Moments and Best-Fit Ellipse. *Optics & Laser Technology*, 2020, 128, 1062. <https://doi.org/10.1016/j.optlastec.2020.106220>
 24. Wang, W., Li, H., Wang, K. Pavement Crack Detection on Geodesic Shadow Removal with Local Oriented Filter on LOF and Improved Level Set. *Construction Building Materials*, 2020, 237(2020), 117750. <https://doi.org/10.1016/j.conbuildmat.2019.117750>
 25. Wang, W., Li, L., Han, Y. Crack Detection in Shadowed Images on Gray Level Deviations in a Moving Window and Distance Deviations Between Connected Components. *Construction Building Materials*, 2021, 271, 121885. <https://doi.org/10.1016/j.conbuildmat.2020.121885>
 26. Wang, W., Li, R., Wang, K., Lang, F., Chen, W., Zhao, B. Crack and Fracture Central Line Delineation on Steger and Hydrodynamics with Improved Fractional Differential. *International Journal of Wavelets, Multiresolution and Information Processing*, 2020, 18 (5) 2050037 (2020) (21 pages). <https://doi.org/10.1142/S021969132050037X>
 27. Wang, W., Wang, M., Li, H., Zhao, H., Wang, K. Pavement Crack Image Acquisition Methods and Crack Extraction Algorithms: A Review. *Journal of Traffic and Transportation Engineering (English edition)*, 2019, 6 (6), 535-556. <https://doi.org/10.1016/j.jtte.2019.10.001>
 28. Wang, W., Yang, N., Zhang, Y., Wang, F., Cao, T., Eklund, P. A Review of Road Extraction from Remote Sensing Images. *Journal of Traffic and Transportation Engineering (English Edition)*, 2016, 3(3), 271-282. <https://doi.org/10.1016/j.jtte.2016.05.005>
 29. Wang, W., Zheng, S., Lian, R., Liang, Y. Rock Fissure Pattern Characterization by Combining 1-D Fractal Dimension and Statistical Analysis. *IEEE/CAA Journal of Automatica Sinica*, 2017, 1-7. <https://doi.org/10.1109/JAS.2017.7510391>
 30. Wei, X., Zhang, Z., Chai, Z., Wei, F. Research on Lane Detection and Tracking Algorithm Based on Improved Hough Transform. *IEEE International Conference of*

- Intelligent Robotic and Control Engineering (IRCE) (2018). <https://doi.org/10.1109/IRCE.2018.8492932>
31. Winserng, C., Phooi, Y. A Framework for Lane Departure Warning System for Various Lane Markings. Conference: 2017 International Workshop on Advanced Image Technology, At: Penang, Malaysia.
 32. Xu, X. Real-Time Lane Departure Warning Based on Fusion Segmentation and LASSO Regression. Journal of Wuhan University of Science and Technology, 2020, 43(1).
 33. Yi, S. C., Chen, Y. C., Chang, C. H. A Lane Detection Approach Based on Intelligent Vision. Computers & Electrical Engineering, 2015, 42, 23-29. <https://doi.org/10.1016/j.compeleceng.2015.01.002>
 34. Zhang, L. Study on Lane Departure Warning Strategy Based on Machine Vision. Dalian, Dalian University of technology, 2017.
 35. Zhang, Z., Hao, X., Liu, S., Cheng, C. Line Detection Based on Hough One-Dimensional Transform. Acta Optica Sinica, 2016, 36(4), 0412005. <https://doi.org/10.3788/AOS201636.0412005>
 36. Zhao, X. Liver Vessel Segmentation Algorithm Based on Fuzzy Enhance and Hessian Matrix. Journal of Zhejiang Industry & Trade Vocational College, 2018-02.
 37. Zheng, Q., Tian, X., Jiang, N., Yang, M., Layer-Wise Learning Based Stochastic Gradient Descent Method for the Optimization of Deep Convolutional Neural Network. Journal of Intelligent & Fuzzy Systems, 2019, 37(4), 5641-5654. <https://doi.org/10.3233/JIFS-190861>
 38. Zheng, Q., Yang, M., Tian, X., Jiang, N., Wang D. A Full Stage Data Augmentation Method in Deep Convolutional Neural Network for Natural Image Classification. Discrete Dynamics in Nature and Society, 4706576, 2020. <https://doi.org/10.1155/2020/4706576>
 39. Zheng, Q., Yang, M., Yang, J., Zhang, Q., Zhang, X. Improvement of Generalization Ability of Deep CNN via Implicit Regularization in Two-Stage Training Process. IEEE Access, 2018, 6, 15844-15869. <https://doi.org/10.1109/ACCESS.2018.2810849>
 40. Zheng, Q., Zhao, P., Li, Y., Wang, H., Yang, Y. Spectrum Interference-Based Two-Level Data Augmentation Method in Deep Learning for Automatic Modulation Classification. Neural Computing and Applications, 2020, 1-23. <https://doi.org/10.1007/s00521-020-05514-1>
 41. Zhou, S. Y., Jiang, Y. H., Xi, J. Q., Gong, J., Xiong, G., Chen, H. A Novel Lane Detection Based on Geometrical Model and Gabor Filter. IEEE Intelligent Vehicles Symposium, 2010, 59-64. <https://doi.org/10.1109/IVS.2010.5548087>

