Soft Computing (2019) 23:9161–9174
https://doi.org/10.1007/s00500-018-3607-x

FOCUS

A robust lane detection method based on hyperbolic model

Wenhui Li[1] · Feng Qu[1] · Ying Wang[1] · Lei Wang[1] · Yuhao Chen[1]

Published online: 8 November 2018

Abstract

Lane detection is an essential part of safety assurance in intelligent vehicle and advanced driver assistance systems. Despite many methods having been proposed, there still remain challenges such as complex road surface and large curvature. In this paper, we present a robust lane detection method under structured roads to solve these issues. The method contains two parts: straight line detection in near field and curve matching in far field. Instead of generating top-view image by inverse perspective mapping (IPM), we propose a new form of IPM application to reduce noise that we only take advantage of sub-pixel-level spatial relations and project line segments obtained by line segments detector to top-view image. Then, we apply density-based spatial clustering of applications with noise to clustering segments and design a fusion method to extract the optimal lines combination from clusters. Finally, a weighted hyperbolic model is proposed to finish curve fitting. The results of experiment indicate that the method has robust performance in complex environment.

Keywords Lane detection · LSD · DBSCAN clustering · Curve fitting

1 Introduction

Intelligent vehicle has become a commanding height of the current automobile development. Advanced driver assistance systems based on visual sensors need to be able to effectively identify and detect lanes, vehicles and other obstacles on the structured road, and optimize vehicle conditions while moving(Yang et al. 2018). As an indispensable part of vision-based lane departure warning system (LDWS) and adaptive cruise control, the main purpose of lane detection is to detect line markings on road, so as to determine whether there is a trend that the car is going to deviate from the current lane. The accuracy of detection and mathematical fitting is the basis of lane keeping, which ensures the stability and the safety of the systems. According to Insurance Institute for Highway Safety (IIHS), about 50% of the car traffic accidents are caused by cars deviation from their normal driving lane, and lane departures are also regarded as the main cause of the vehicle rollover accidents. Vision-based LDWS with lane detection can reduce the accident mortality by 86% (Cicchino 2017). Visual representation of lane detection is shown in Fig. 1, in which the blue lines are our targets.

So far, many lane detection algorithms have been proposed. According to Narote et al. (2018), lane detection contains two approaches, feature-based methods (Aly 2008; Taubel et al. 2014; Niu et al. 2016) and model-based methods (Ruyi et al. 2011; Wang et al. 2011; Shin et al. 2015). The former approach aims to extract low-level features from the images and the latter uses geometry to describe the shape of lanes. Though different ideas and methods are adopted in the two types of detection, some common strategies are introduced in both approaches. Aly (2008), Taubel et al. (2014), Ruyi et al. (2011) and Shin et al. (2015) apply inverse perspective mapping (IPM) to generate top-view image and remove perspective effect. Features such as parallel and vertical can be used on top-view image after edge detection. The effect of IPM usually depends on the mathematical modeling of the mapping relations between the two planes and interpolation for generating IPM images. Besides IPM, Hough transform (HT) is another widely used algorithm mainly for straight line detection or linear modeling (Wang et al. 2011; Niu et al. 2016). Because of the voting procedure in HT, the edge of shadows and other lane marks would be strong noise. In addition to that, linear model is hard to describe the lanes with large curvature. So, challenges still exist as follows.

Communicated by A. K. Sangaiah, H. Pham, M.-Y. Chen, H. Lu, F. Mercaldo.

✉ Feng Qu
qufeng_jlu@163.com

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China

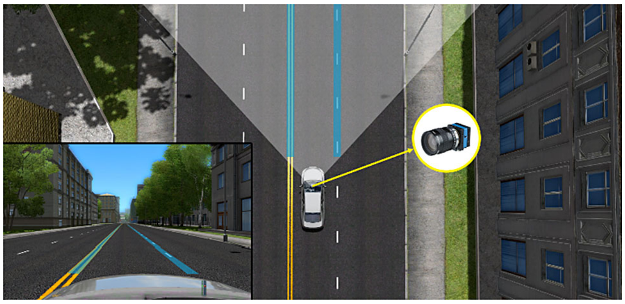


Fig. 1 Visual representation of lane detection

1. Noise interference. Lane detection is greatly affected by traffic conditions on urban streets. Unclear lanes, complex road marks and shadows further increase the difficulty of detection. Methods based on IPM are greatly affected by these noises. Interpolation algorithms in IPM also produce sawtooth-type noise. The farther the space is from camera, the worse the quality of interpolation result is.
2. Environment factors. Weather and light will also affect lane feature extraction, and overexposure will lead to unclear lane, resulting in missed detection.
3. Curvature. Lanes usually have a certain curvature. When the curvature of the lane become relatively large, simple models cannot fit the shape of the lane effectively.

To solve the influence of IPM on line detection and reduce the noise generated by road markings and shadows, we consider applying a two-step method to overcome the challenges above. In this paper, a lane detection algorithm based on line segment detector (LSD) (Grompone von Gioi et al. 2012) and weighted hyperbolic model is proposed. Algorithm flow is shown in Fig. 2. Therefore, the lane detection contains two parts: straight line detection in near field and curve fitting in far field.

In the near field, we abandon the idea of generating IPM image but project line segments to top-view plane with mapping matrix. The line segments $\{L\}$ represented as endpoint coordinates are extracted from the gray image by the LSD. The endpoint coordinates in $\{L\}$ are projected to the top-view image with IPM matrix and get a new set $\{L\} \rightarrow \{L_{\text{IPM}}\}$. Then, the segments in $\{L_{\text{IPM}}\}$ are represented by point $p(\rho, \theta) \in \{P_{\text{IPM}}\}$ with the idea of polar space. We cluster $\{P_{\text{IPM}}\}$ through density-based spatial clustering of applications with noise (DBSCAN). After that, a method for fusion is designed to fuse clusters and generate candidate lines. Because the lanes on top-view plane are nearly parallel and the distance is similar, a straight lines combination that meets the characteristics is selected from candidate lines.

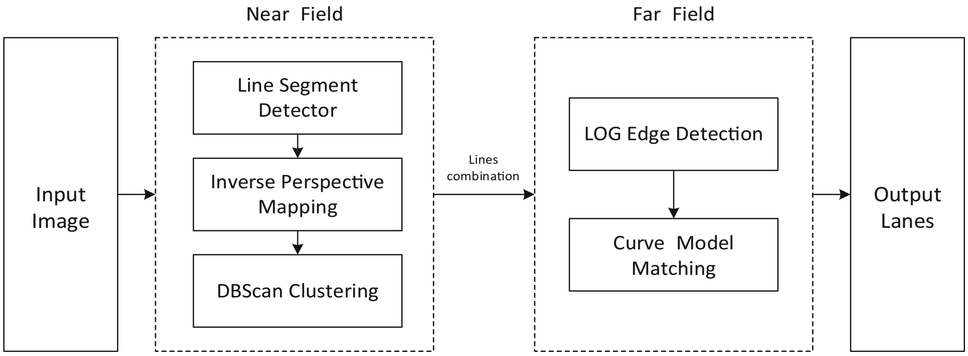
In far field, we reproject the line combination in near field to the source image coordinate system and get the parameters of these lines. After that, a 5×5 convolution kernel for edge detection is obtained by a simplified LOG operator. With this kernel, the edge is extracted from the gray image by convolution. Then, we match the weighted hyperbolic model on edge image to obtain the optimal curvature of the hyperbolic. At last, the optimal lane is obtained.

The main contributions of this paper are as follows:

1. We propose a new form of IPM application. Instead of generating top-view images, we consider processing the line detection on the original image and only analyzing the spatial relationship of line segments on the top-view image with sub-pixel-level mapping matrices between the two images. It can effectively correct the interpolation error caused by IPM.
2. We use DBSCAN to cluster lane edge segments and design a method to get the fusion result of the elements in the cluster and extract the best combination among the clusters under the constraints to obtain a set of candidate lines in near field.
3. We propose a weighted hyperbolic model to fit the lane lines in far field, which can adapt to the curvature of the lane line and get more accurate results on the lanes with large curvature.

The rest of this article is organized as follows. Section 2 introduces the existing relevant studies. Section 3 introduces the methods of processing, clustering and screening straight

Fig. 2 The framework of the proposed method



lines in near field part. In Sect. 4, we describe the process of edge detection and weighted hyperbolic model matching. In Sect. 5, the experimental results are given and analyzed. Finally, the inference is given in Sect. 6.

2 Related work

At present, the popular traditional lane detection algorithms are divided into two steps, first is to extract lane features [color (Chiu and Lin 2005; Mammeri et al. 2016), edge (Lan et al. 2009; An et al. 2013; Tu et al. 2013; Wu et al. 2014; Niu et al. 2016), etc.] directly, and then fit the lane with default models (straight line model (Saudi et al. 2008), parabola (McCall and Trivedi 2006; Zhao et al. 2012; Yu et al. 2017), polynomial, spline (Wang et al. 2000, 2004; Hsiao et al. 2009; Wang and An 2010), etc.). Since the road environment is usually very complex, the algorithms based on feature extraction directly are affected by noise from non-lane road, which means the result is unstable. The difficulty of the latter is whether the default model matches the lane and whether the parameters of the model can be acquired correctly.

In the preprocessing stage, it is necessary to denoise (Liu et al. 2017) and extract lane features in frames such as color, gradient, edge, lane width and vanishing point in a complex road environment. Chiu and Lin (2005) propose a lane mark detection method based on color segmentation and obtain thresholds by using gray-level histogram. Mammeri et al. (2016) apply the HSV space to extract the yellow and white features of the lanes for lane detection. But the detection performance is affected by night environment, different lighting conditions or passing cars. In a complete sequence of images, the feature information of the target or the background may change in different situations, such as illumination variation, occlusion and scale variation (Liu et al. 2018). Since most color models are sensitive to light, it is usually necessary to use edge detection to extract lane features. The common edge detection methods include Sobel (Lan et al. 2009; Tu et al. 2013; Ozgunalp et al. 2017; Song et al. 2017), Canny (Wang et al. 2000; Wu et al. 2014; Andrade et al. 2017), steerable filter (McCall and Trivedi 2006), etc. Niu et al (2016) use Canny operator to obtain edge image and then use DBSCAN to cluster the small line segments obtained by modified Hough transform to get straight line equations of the left and right lanes. This method performs well in most cases, but the performance decreases obviously in the case of large curvature. Andrade et al. (2017) use oriented gradient filter to remove unnecessary information and introduce the yaw angle of the vehicle to correct the filtering direction. Vanishing point (Ozgunalp et al. 2017; Piao and Shin 2017; Tian et al. 2018) is used to control ROI region and multi-lane detection parameters. Lee et al. (2018) use Λ-ROI to constrain the lane line in perspective images.

The representation of the default model is achieved from simple linear models to complex spline models. Saudi et al. (2008) apply random Hough transform to detect straight lines in the literature. This method has good performance on the straight lanes, but cannot detect the lane with curve. McCall and Trivedi (2006) propose a simple parabolic model, which includes lane location, angle and curvature. The authors (Yu et al. 2017) get the control points by getting the straight line model first, and use the control points to fit the curve in the parabola model, and then apply RANSAC method to optimize the parameters. Although the parabolic model fits the geometric features of lanes, the connection part between the straight line and the curve is usually bad. To solve this problem, a parabolic model of second order is proposed (Zhao et al. 2012). However, with the order of the model increasing, it is more sensitive to noise. Wang et al. (2004) propose a new spline-based lane model, which uses cubic B-splines to fit the midline on both sides of the lane. In their method, both sides of the driveway are assumed to be parallel. A lane model based on CabMull-ROM spline is proposed (Wang et al. 2000). It is also called Overhauser spline, that is, local interpolating splines. Vanishing point is used to calculate B-spline control points (Hsiao et al. 2009). S Jung et al. (Jung et al. 2015) accumulate pixels along the time axis on the scanning line in the sequence images and then use a cubic polynomial and least-square method to fit the lane line.

3 Straight line combination extraction

The main purpose of this part is to achieve primary localization of lanes by extracting the optimal line combination that meet the feature, and provide parameters for subsequent curve fitting. After the internal and external parameters of the camera are calibrated, we can get the vanishing point of the image. The ROI area of near field is selected as the bottom 3/4 part of the area below the vanishing point. The process is mainly divided into three steps: line extraction based on LSD in near field, inverse perspective transformation of lines, clustering and selection of straight line segments.

As shown in Fig. 3, the algorithm in near field starts by setting the ROI of the grayscale image. Then, we detect line segments by LSD, and simultaneously the perspective and inverse perspective mapping matrices are calculated. After we project the segments to the ground plane with the matrix, a histogram is applied to obtain the lane direction. DBSCAN is used for clustering the segments, and we proposed a fusion and extraction method to process the clusters and get optimal lines combination of results. At last, the combination is reprojected to the original image.

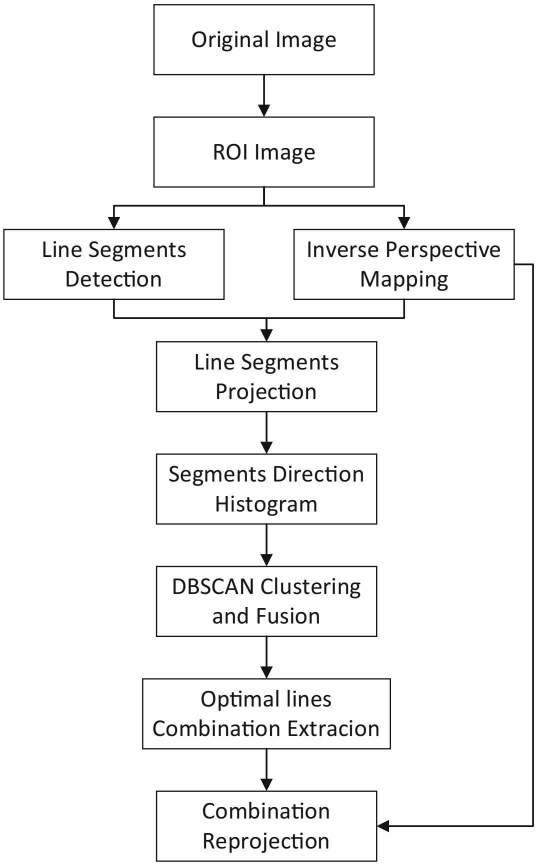

Original Image
ROI Image
Line Segments Detection
Inverse Perspective Mapping
Line Segments Projection
Segments Direction Histogram
DBSCAN Clustering and Fusion
Optimal lines Combination Extracion
Combination Reprojection

Fig. 3 Processes of straight line extraction in the near field

3.1 Line segments detection

The first step for straight line extraction in near field is to detect line segments in the ROI by LSD algorithm. Line segments detector is an algorithm for extracting local straight lines, and it can get sub-pixel accuracy result in linear time (Grompone von Gioi et al. 2012). Although Hough transform is widely used in line detection, it still has some limitations, such as the shortcoming of slow calculation rate and the large amount of computation and storage. Another reason we choose LSD is that we aim to use the mapping relations between perspective image and top-view image, which means we need endpoint coordinates but not straight line equations. However, the voting procedure in HT determines that the output of HT must be a straight line. The advantage is that it can extract straight line segments accurately on any digital image, and does not need to set parameters manually. The algorithm calculates the level-line angle of each pixel in the image, which is perpendicular to the gradient angle at the current pixel point. The level lines which are adjacent and within a certain threshold form a line-support region, as shown in Fig. 4.

Considering the characteristic that LSD is based on the gradient direction, one edge is usually divided into two segments on the image, which may affect the complexity of computation. Therefore, we do LSD straight line detection directly on the gray image and make preliminary screening according to the line angle. In the world coordinate system, the object perpendicular to the ground is at an angle of 90° on the perspective plane (relative to the X axis of the image). And the edge of the object parallel to the ground is parallel to the X axis in the perspective plane. Therefore, the straight line segments approaching these two angles can be identified as noise and filtered out. In the actual filtering process, we choose to retain the straight line at the angle of $[5°, 85°] \cup [95°, 175°]$.

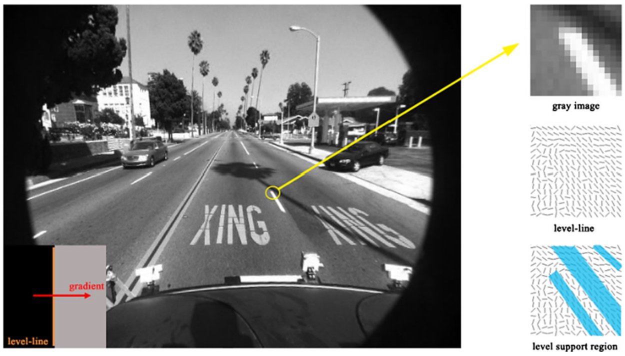


Fig. 4 Level line and gradient are perpendicular to each other. The orange line represents level line and the red one is gradient. On the right side, the three images represent gray image, level lines and level-support region, respectively

3.2 Inverse perspective mapping (IPM)

To obtain the information of the lane marks on the road better, it can effectively highlight the geometric features of the lanes by converting the image to bird view, and eliminate the interference and error of the perspective for detection and recognition. On the perspective image, lanes which are not obviously relevant can show geometric characteristics such as parallel and equidistant after inverse perspective mapping.

For the following considerations, we decide to detect the lines first and then proceed to IPM:

1. LSD algorithm is sensitive to the sawtooth noise. Because of the perspective transformation, the closer to the vanishing point (the farther away from the visual distance), the less the image feature is. After IPM, a large number of sawtooth noise will appear on the far ground, resulting in excessive noise generated by line detection, which will affect subsequent calculations.
2. When the vehicle is moving, the pitch angle of the camera changes because of the uneven road or sudden change in speed. The pitch angle measured under static condition is not accurate in motion, which also causes the error of IPM image. This error may cause the edge of some

obstacles (such as vehicles) on the source image to be detected after the inverse perspective transformation.

3. Because we do not do the straight line detection on the top-view image, we only need to find the coordinate mapping matrix from the original image to the top-view image. This process only needs to be calculated once. The parameter equation of line segment is converted to the ground plane coordinate directly by the mapping matrix, instead of using inverse perspective transformation and pixel interpolation for the whole image, which will greatly reduce the computation.

According to Zhang's theory (2000), a 3D point $\tilde{M} = [X, Y, Z, 1]^{\mathrm{T}}$ to its image projection $\tilde{m} = [u, v, 1]^{\mathrm{T}}$ can be represented by the following expression:

$$s\tilde{m} = A\begin{bmatrix} R & T \end{bmatrix}\tilde{M} \tag{1}$$

where s is an arbitrary scale factor, A is the camera intrinsic matrix, and $\begin{bmatrix} R & T \end{bmatrix}$ is the extrinsic matrix of the camera. R and T are the rotation matrix and translation matrix, respectively. We assume the world coordinate system $\left(X_w \ Y_w \ Z_w \right)$, camera coordinates system $\left(X_c \ Y_c \ Z_c \right)$ and image coordinates system (X_i, Y_i). As shown in Fig. 5, we can solve the upper expression based on the intrinsic matrix and the extrinsic matrix of the camera to get the mapping matrices (Aly 2008).

The transformation from the world coordinate system to the camera coordinate system is mainly represented by the rotation and translation of the coordinate system. We define that the pitch angle of the camera coordinate system relative to the world coordinate system is alpha. During the installation of the camera, we set the yaw angle as yaw $= 0$, the roll angle as roll $= 0$, the height of camera as h. And we assume that the origin points of two coordinate systems coincide. The intrinsic matrix of the camera is defined as $A = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}$, where (f_u, f_v) is the focal length of the camera, (c_u, c_v) is the optical axis center of the camera on the image. We define that α is pitch angle,$^{i}P = \{u, v, 1, 1\}$ represents image plane, and $^{g}P = \left\{x_g, y_g, -h, 1\right\}$ represents ground plane.

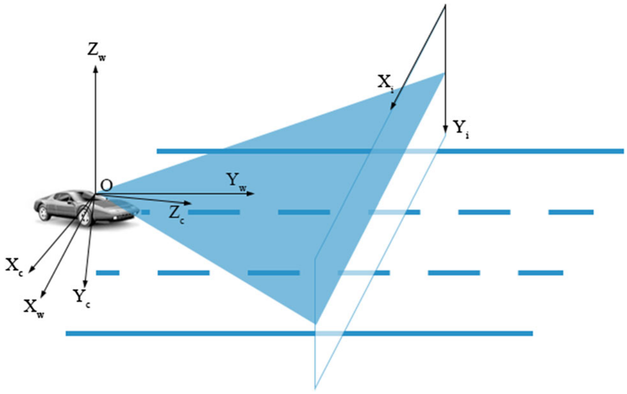


Fig. 5 Relations among systems

Here, we can get relations as follows.

1. Projection from the source frame to the ground plane, $P = {}^{g}_{i}T^{i}P$:

$$ {}^{g}_{i}T = h\begin{bmatrix} -\frac{1}{f_u} & 0 & \frac{1}{f_u}c_u & 0 \\ 0 & \frac{1}{f_v}\sin\alpha\cos\alpha & -\frac{1}{f_v}c_v\sin\alpha - \cos\alpha & 0 \\ 0 & \frac{1}{f_v}\cos\alpha & -\frac{1}{f_u}c_v\cos\alpha + \sin\alpha & 0 \\ 0 & -\frac{1}{hf_v}\cos\alpha & \frac{1}{hf_u}c_u\cos\alpha - \frac{1}{h}\sin\alpha & 0 \end{bmatrix} \tag{2}$$

2. The inverse of the transform from the ground plane to the source frame, $P = {}^{i}_{g}T^{g}P$:

$$ {}^{i}_{g}T = \begin{bmatrix} f_u & c_u\cos\alpha & -c_u\sin\alpha & 0 \\ 0 & c_v\cos\alpha - f_v\sin\alpha & -f_v\cos\alpha - c_v\sin\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \end{bmatrix} \tag{3}$$

By the two transformations above, we can get the sub-pixel points mapping matrix from the original image to the IPM image, and their inverse mapping matrix.

According to the set of line segments we get on the source image, we project these line endpoint coordinates to the bird's-eye view through the matrix and get the set of straight lines $\{L_{\mathrm{IPM}}\} = \{(P_{\mathrm{start}}, P_{\mathrm{end}})\}$ after the projection. Among them, P_{start} stands for the starting point of a line segment, and P_{end} stands for the ending point.

As shown in Table 1, if we apply IPM and image interpolation before LSD straight line detection, we will take an extra 7.8 ms per frame, which means another 24.84% time cost for generating top-view image. As shown in Fig. 6, the straight line segments obtained from the source image are more consistent with features and spatial relations, and have less false positive results due to sawtooth noise. In Fig. 6b, c, the method IPM with interpolation extracts the boundary of the tree shade wrongly.

Table 1 Average detection time of two methods

Method	Total time (ms/frame)
IPM with interpolation	39.2
IPM without interpolation	31.4

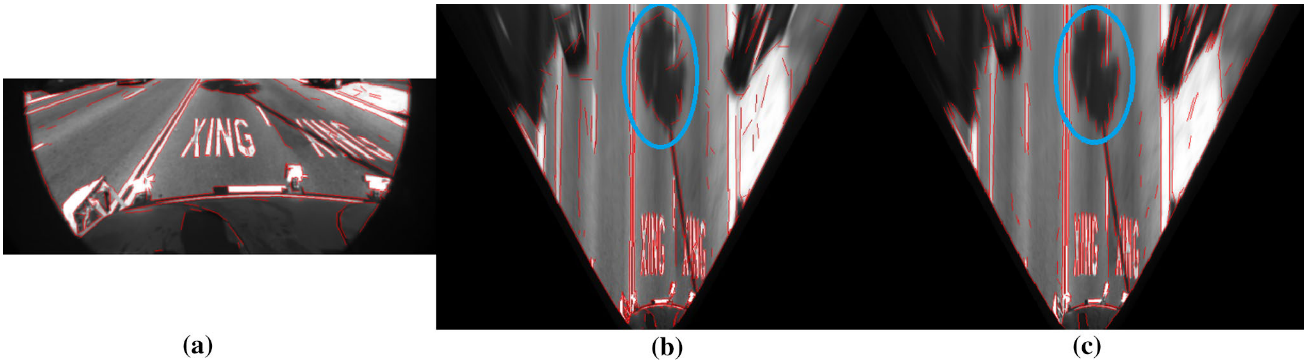


Fig. 6 **a** LSD on the ROI of the source image, **b** results of projecting the segments on ROI to IPM image, **c** results of applying LSD on the IPM image. It can be seen that the segments in **b** are much more accurate than that in **c**

3.3 Line segments clustering

Though LSD algorithm performs well in the detection speed, it also has obvious disadvantages in some cases.

1. In the case of line intersection, LSD algorithm needs to find the unused pixels from a sorted list as seed points, which means a point can only belong to a straight line segment. If two straight lines intersect, there must be at least one straight line cut to two. Because the algorithm is based on the gradient and the gradient of the intersection point is often small (not detected as the edge point), it is very likely that the two intersected lines are split into four lines segments at the intersection point.
2. The ranking algorithm in LSD is a pseudo-sorting algorithm based on greedy algorithm. So when a long straight line is obscured or partially blurred, it will often be split into several straight lines segments. These shortcomings do not exist in Hough transformation.

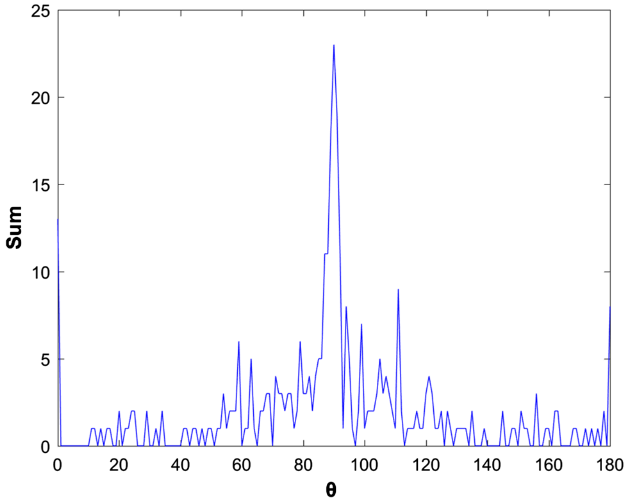


Fig. 7 Histogram of lines directions. The angle θ is defined as $\theta = \arctan(\Delta y/\Delta x)$; Δx and Δy are the absolute value of the two end coordinates of a line segment

Therefore, relative to Hough transform, LSD can get more straight line segments in an image. In the procedure of extracting lanes, one lane edge may be divided into multiple line segments. For this reason, we consider applying clustering method to merge multiple segments into a straight line. With the idea of Hough transformation, we transform the set of $\{L_{\text{IPM}}\}$ from the image coordinate system to the polar coordinate system. Each point (ρ, θ) in a polar coordinate system represents a specific line segment.

The parameter equation is as follows:

$$\rho = x \cdot \sin\theta_P + y \cdot \cos\theta_P, \quad 0^\circ < \theta_P < 180^\circ \qquad (4)$$

To simplify the computation, we make histograms projection according to the angle between the line segment and the X axis in Cartesian coordinate system. Here, θ_D meets $\theta_D = \theta_D + 90^\circ$. Because lanes and road edges are parallel, we think that the theta of the peak region in the histogram is the lane direction in near field. Considering that the change of pitch in vehicle driving will affect the inverse perspective transformation, we focus on the lines of 5° near the peak point. Figure 7 shows the angle of line segments on IPM image, and the peak point is nearly 90°. The segments in polar coordinate form near peak value are shown in Fig. 8.

The advantage of DBSCAN algorithm is that clustering does not depend on the order of traversing points. It can identify various shapes of clusters fast and eliminate noise points effectively (Hahsler and Piekenbrock 2017). More importantly, unlike K-means (Kant et al. 2018) and other clustering methods based on distance (Liu et al. 2017), DBSCAN does not need to set the number of clusters which satisfies our needs (Sahu et al. 2018). In polar coordinates system, the line segments like the same straight line are close. In this paper, we use DBSCAN to cluster the line segments and use the length of the line segment in the cluster as weight. The segments in the same cluster are fused and represented by one

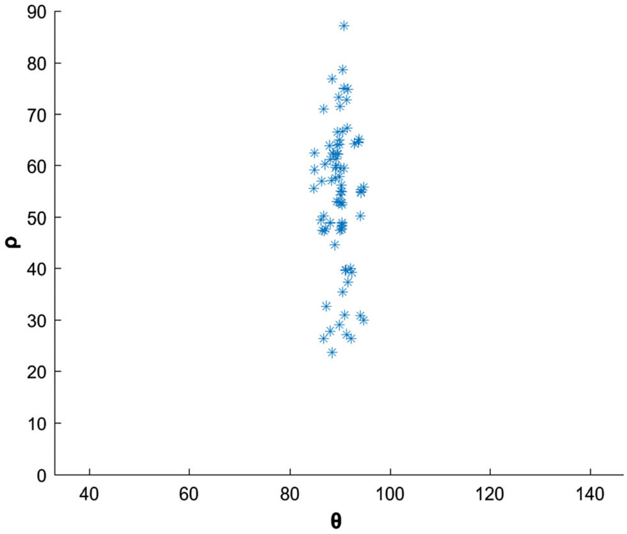


Fig. 8 Distribution of line segments in polar coordinates after histogram filter

point. By this method, we can effectively reduce the number of straight lines to get the set S of the clusters and fuse the straight lines $s_k \in S$ in each cluster.

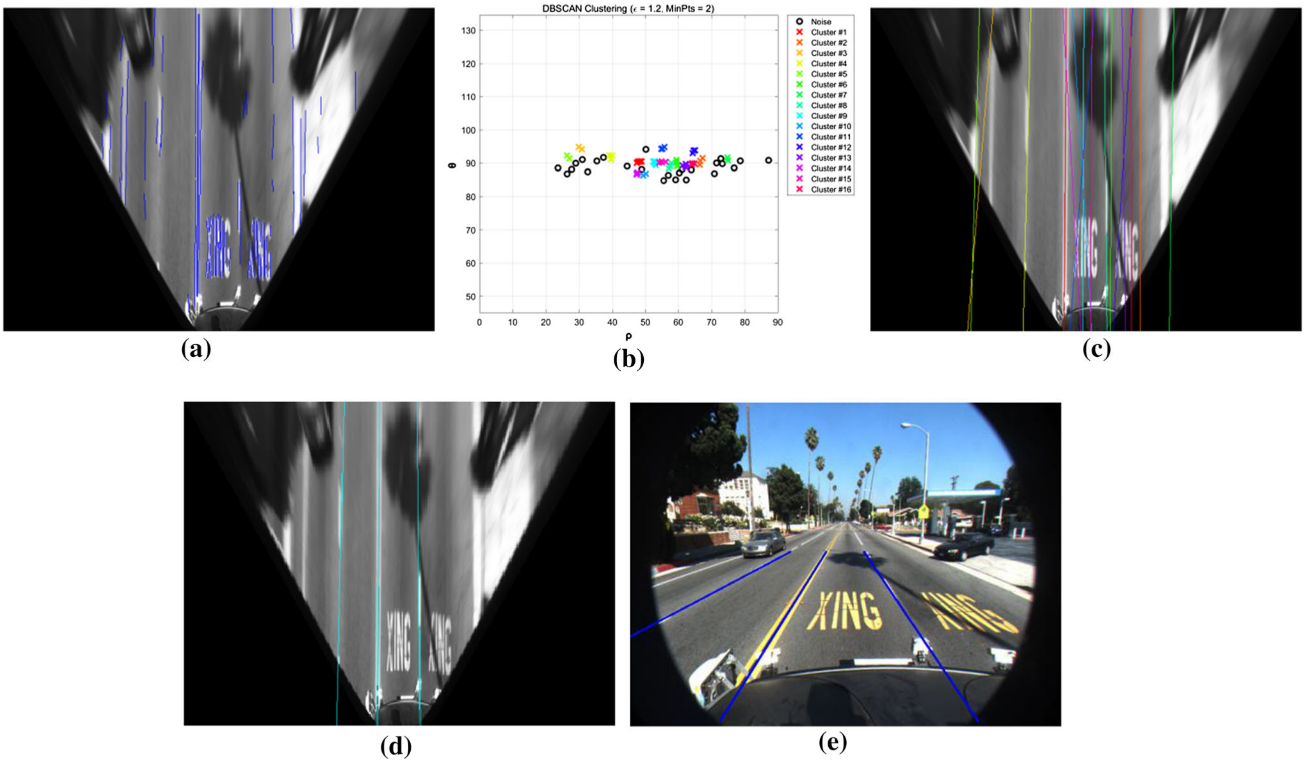
$$w_i = l_i / \sum_{j=0}^{n} l_j \tag{5}$$

$$\begin{cases} \rho_c = \sum_{i=0}^{n} w_i * \rho_i \\ \theta_c = \sum_{i=0}^{n} w_i * \theta_i \end{cases}, \quad (\rho_i, \theta_i) \in s_k \tag{6}$$

We define l_i as the length of a line segment in cluster s_k after clustering, and w_i is the weight of the line segment. According to the weight w_i, the fused straight lines of the clusters are calculated $(\rho_c, \theta_c) \in L_c$. Not all fused straight lines in set L_c belong to lanes. It is necessary to find an optimal line combination that meets the geometric characteristics of lanes through some constraint conditions. The distance between the lanes is usually fixed, so the distance of the adjacent lines in combination should meet $d_1 \le \Delta\rho \le d_2$, in which $[d_1, d_2]$ is the threshold interval of lane width.

Here, we design a line combination extraction method: the fused lines in L_c are ordered in ascending order by $|\rho|$, and then the sequence is reformed to a directed graph according to the condition of the threshold interval. So that each path from top to bottom of the graph is a candidate combination. Among all the candidate combinations, we consider that the combination which has maximum sum of length is the optimal one. After projecting the combination back to the perspective image, we will further deal with the final straight lines in next section.

Figure 9a is a collection of unclassified line segments on IPM image. Figure 9b is the result of DBSCAN clustering, in which the black circles represent the non-classified noise



(a)

(b)

(c)

(d)

(e)

Fig. 9 **a** Line segments projected on IPM image, **b** results of DBSCAN clustering, one color represents one cluster, **c** results of cluster fusing, and colors of lines correspond to the colors of clusters, **d** optimal combination extraction from candidate lines, **e** lines reprojected to original image and different colors correspond to different clusters. The X axis represents the direction of straight line, and the Y axis is a scaling rho. Figure 9c shows the fused results of clusters on IPM image, and the color of line corresponds to the color of each cluster in Fig. 9b. Figure 9e is the best line combination obtained after our proposed filtering method. Figure 9e maps the line combination back to the source image by using the previous obtained mapping matrix.

4 Lane fitting

The shape of the lane is not usually straight but has a certain curvature. The line combination obtained at the previous stage cannot satisfy the actual road conditions in far field. To improve the accuracy of lane detection, we need to fit the curve of the lane. In this paper, we use a weighted hyperbolic model to match pixels on edge images to get the highest matching rate curvature.

4.1 Edge detection

At present, the widely used edge detection operators on space domain such as Sobel, Canny, Prewitt operators are sensitive to noise. Some new theory such as fractal shows potential for edge detection (Liu et al. 2014). Laplacian operator is also a commonly used edge detection operator based on the two derivatives (Maini and Aggarwal 2009). To reduce the sensitivity to noise, Marr combined Gaussian filtering with Laplacian edge detection (Fig. 10).

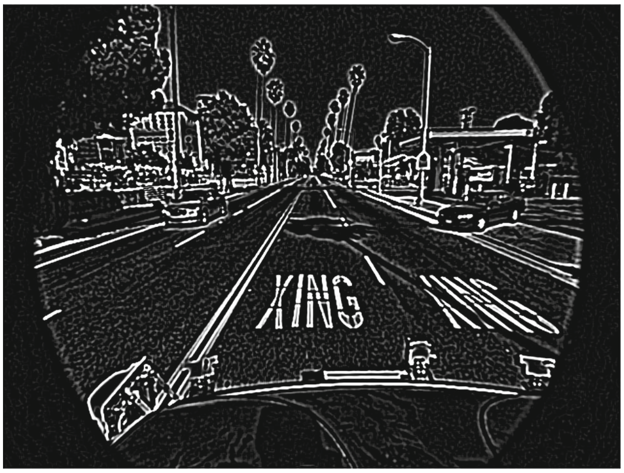


Fig. 10 Edge detection result

First, the original image is processed by Gaussian smoothing; thus, the noise is restrained maximally. Then, the edge is extracted from the smooth image, and the probability of the false edge is minimized. The LOG operator has the following form:

$$G(x, y) = \left(-\frac{1}{\pi\sigma^4}\right)\left(1 - \frac{x^2 + y^2}{2\sigma^2}\right)e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} \tag{7}$$

To simplify the operation, we designed a convolution kernel of 5 × 5 to deal with the image. The convolution kernel of the design is as follows:

$$k = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & 2 & 0 & 2 & -1 \\ -1 & 2 & 2 & 2 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

The kernel is used to convolve the gray image to get the edge image.

4.2 Curve fitting with weighted hyperbolic model

Because curvatures of the left and right lanes are basically the same in the perspective image, the hyperbolic model is therefore used to fit the curves. The equation is $y = A \cdot \frac{\tan\alpha}{x-h} + k \cdot x + b$, where α is the curvature of the hyperbola and A is a parameter for controlling changes of the curvature. Setting a certain range of the hyperbola's curvature, and with the parameters obtained by former stage, we can calculate all curvatures in the range. The step size of the curvature is 1. After that, the optimal matching ratio can be calculated by matching ratio formula and weighting formula. The curve with the highest matching rate is taken as the final fitting result, as shown in relations (8) and (9):

$$M = \sum_{i=0, j=0}^{W,H} w(i, j) \cdot p(i, j) \tag{8}$$

$$w(i, j) = \begin{cases} (i - h)/h & (i > h) \\ 0 & (i \le h) \end{cases} \tag{9}$$

where M is the matching rate, W and H are the width and height of the edge image, (i, j) represents the coordinates of the pixels, and $w(i, j)$ is a weight function. It can easily be seen that the edge pixel has a larger weight if it is closer to the observer, while the weight of the pixel that is out of the range is zero. $p(i, j)$ represents the distance from the point (i, j) of the binary edge image to the curve. The curvature that has the largest matching rate is taken as the curvature of the current lane (Fig. 11).

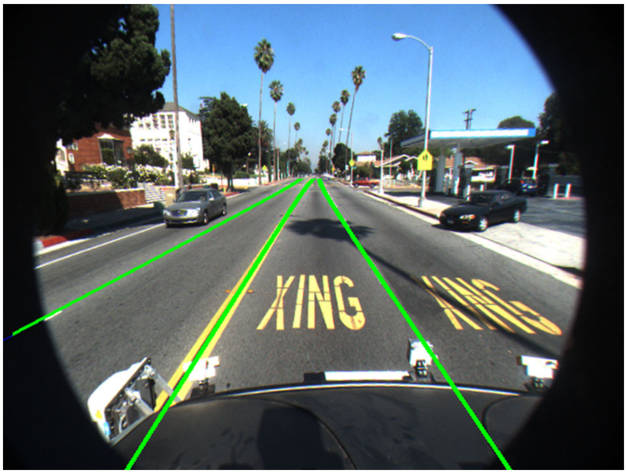


Fig. 11 Results of curve fitting

5 Experiments

5.1 Setup

5.1.1 Environment

The algorithm we use is tested on an Intel I7 4850HQ 2.30 GHz CPU, which is equipped with 16 GB RAM for experiment. The whole system is coded in C++ on Microsoft VS 2013 environment with open source OpenCV image library. We use open source database to conduct experiments. All frame sizes are 640 × 480 pixels, and the program runs at 31 fps.

5.1.2 Dataset

Caltech dataset (Aly 2011): the dataset consists of four sequences of frames in urban city under structured road, totaling 1225 frames with resolution of 640 × 480. The scene includes a variety of complex conditions, including shadow, straight/curve lane, ground crack/road signs interference, occlusion, overexposure and other challenges. The lane edges in *Clip Cordova1* are clear, but the lane curvature is large, and there is road marking interference. In *Clip Cordova2*, the borders of lanes are not clear and some frames are exposed. There are tree shades and passing cars in *Clip Washington1*. Also, there are a lot of shadows and noise in *Clip Washington2*. The dataset also provides a tool to label the ground truth of lanes manually, as well as lane detection results using method described in Aly (2008).

5.1.3 Baseline

The experiment was tested in two-lane mode (detecting only the two lines of the lane on which the vehicle is driving). We compared our results with the methods described by Aly (2008), by Ruyi et al. (2011), by Shin et al. (2015) and by LDTFE (Niu et al. 2016). The reasons we choose these papers are that we try to solve the disadvantages of IPM interpolation. Aly, Ruyi and Shin all applied top-view image to achieve lane detection. Furthermore, all the four methods use Caltech dataset to evaluate performance. Though Niu did not apply IPM, he is the state of art on Caltech dataset.

5.1.4 Metric

The main performance metrics of our algorithm are: accuracy rate AR $= \frac{N_t}{N_{gt}}$, false positive rate FP $= \frac{N_f}{N_{all}}$ and false negative rate FN $= \frac{N_m}{N_{gt}}$. Here, we define the number of true detections as N_t, the number of false detections as N_f, the number of missed lanes as N_m, the number of all ground truth lanes as N_{gt} and the number of all detections as N_{all}.

5.2 Parameters

The main parameters of our algorithm are concentrated in two parts. One is the processing of clustering data before DBSCAN and the setting of some parameters for DBSCAN clustering. The other part is the parameters for screening fused results after clustering.

Because the direction of the lane in the IPM image is about 90° relative to the X axis, the value of θ represented by polar coordinates is around 0° and 180°. If we apply DBSCAN clustering with direct polar coordinates, it may lead to a situation that two straight line segments belonging to the same cluster cannot be clustered together because of θ. So we consider the angle of line direction instead of the normal direction θ in the polar coordinates (Fig. 12).

The IPM image size is 640 × 480, so the range of ρ should be $(-800, 800)$, while the acceptable angle range of lane direction in cluster is about 1°. Obviously, we should reduce ρ with a right scale to ensure the success of clustering. The width of the lane line on the IPM image is usually 3–5 pixels, so ρ is modified by a factor of 1/6.

DBSCAN clustering has two main parameters, one is the radius *eps* and the other is the minimum number *MinPts* specified in the cluster. The two parameters in this paper also determine the effectiveness and performance of the algorithm. The remaining line segments on the image are not very dense after the lines are screened by angle on the source image and the peak of the straight line direction is obtained by histogram on the IPM image. For example, there may be only the left or the right edges in a lane line. So, if *MinPts* is too large and *eps* is too small, many lines will not be clustered successfully and then will be mistaken for outliers, which results in an increase in missed detection. And if *eps* is too large, too many line segments would be clustered together.

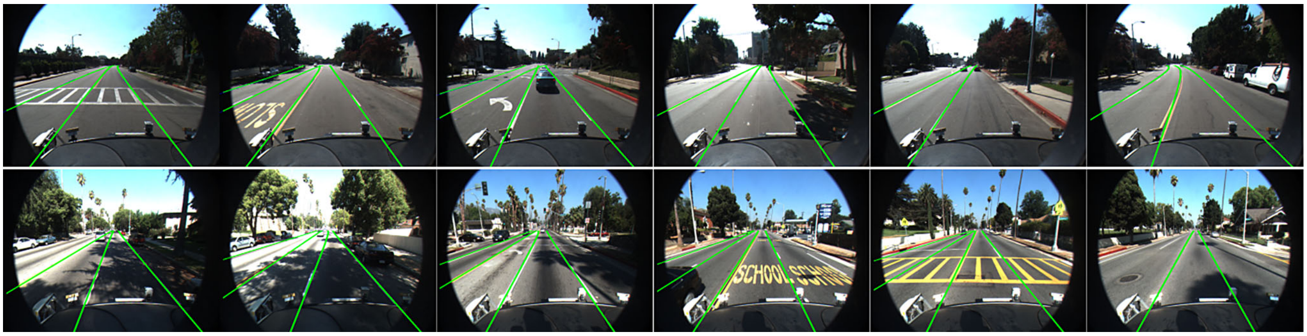


Fig. 12 Examples of correct lane detection

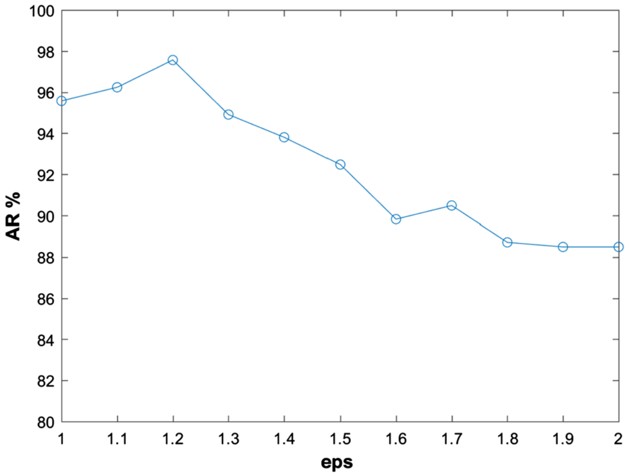


Fig. 13 Impact of different *eps* on accuracy

We take the *Clip Washington2* as the samples to test the impact of different *eps* values on accuracy (AR%), in order to produce the optimal *eps* value. As shown in Fig. 13, when $n = 2$, the value of abscissa *eps* is [1.0, 2.0], the step size is 0.1. The ordinate is the accuracy. Experiments show that when $eps = 1.2$, the accuracy rate is the highest.

In the process of combination extraction, we believe that lane width should be between [50, 80]. Then, the fused lines are placed in a directed graph by the distance among each other. Two lines that meet the threshold interval have a directed path between them. By searching the path from the top to the bottom, the best combination is obtained with the highest confidence. If this combination does not exist, then the two lines which are nearest to the image center and within the threshold will be considered as lane lines.

5.3 Experiment results

As shown in Table 2, LSD costs the most computing time of the whole operation. That is limited by the algorithm performance, which we cannot change. We also find that clustering and fusion part only costs 0.19 ms/frame, which means our method is simple but logical. Because this step determines whether we could extract the best straight line combination, it is the key to build linear model in near field. The total computing time is 31.87 ms/frame, which means the program runs at 31 fps.

We tested the performance of our algorithm on the Caltech dataset. As shown in Figs. 12 and 14, the scenarios include challenges such as curvature, lane marks, shadows, overexpose, etc. In Fig. 14, some intermediate results are listed. Column (a) is the original images. Column (b) is the line segments which are extracted from ROI image and projected to top-view plane. Column (c) is the results after DBSCAN clustering and fusion, and Column (d) is the final combination of clustered lines. Column (e) shows that our method has a good performance on these challenges.

In Table 3, the experimental results are displayed. We get better performance in *Clip Cordova1* and total. The average detection accuracy is 97.5%. The reason is that we abandon generating top-view images, which reduces much noise caused by interpolation and false lines that are not on the ground. The IPM matrices can provide sub-pixel-level accuracy of spatial relations, as well as less computing time for interpolation. In addition to that, the weighted hyperbolic model performs well on curve fitting, so we get a better result in *Clip Cordova1*.

Both Ruyi's and Shin's methods are based on IPM and particle filter, so their results have something in common. They

Table 2 Computing time of each step

Step	Preprocessing	LSD	Clustering and fusion	Curve fitting	Total
Time (ms/frame)	2.12	22.76	0.19	6.80	31.87

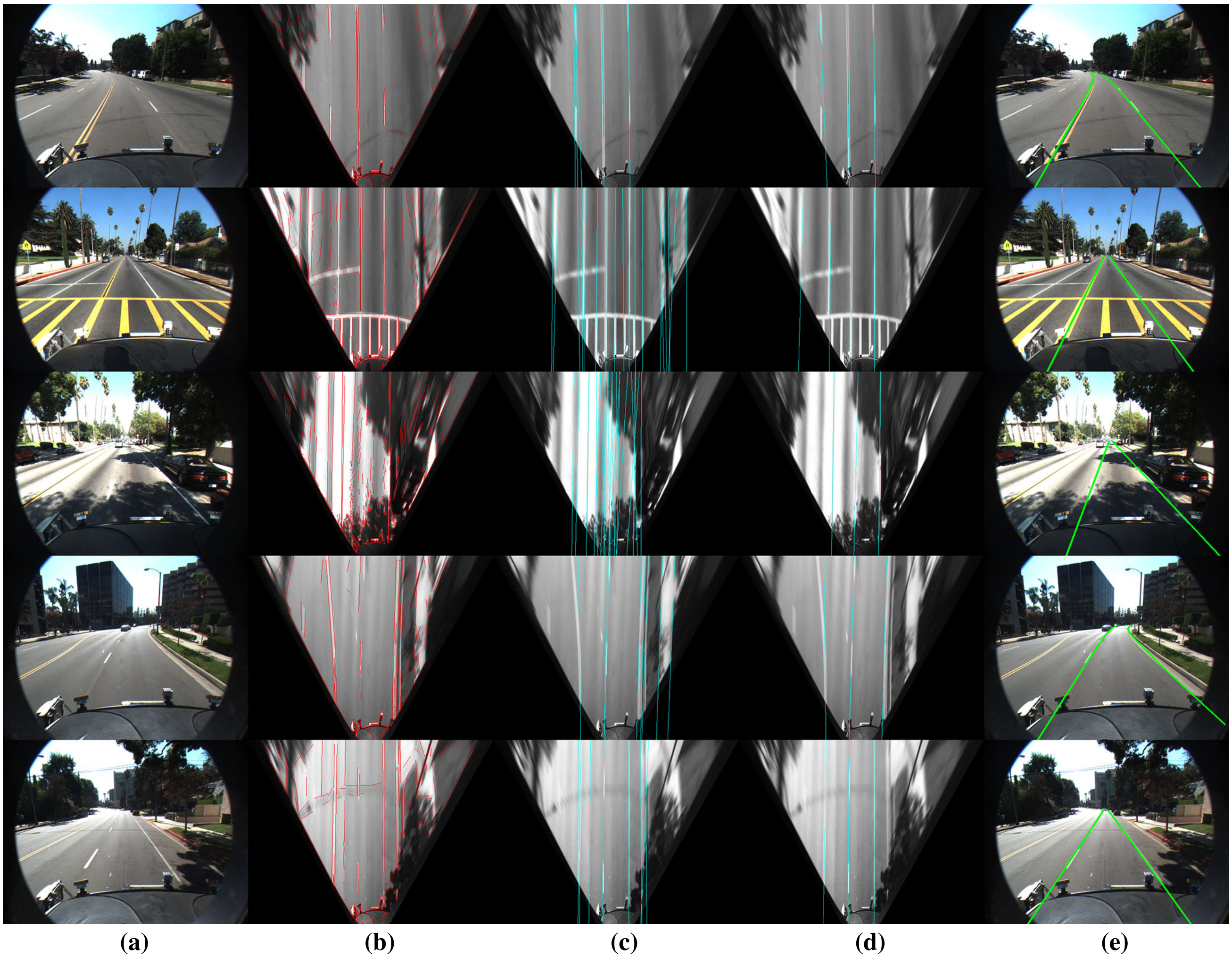


Fig. 14 Some examples of results in various scenarios, including curvature, lane marks, shadows, overexpose, etc. The columns are **a** original images, **b** segments projected to top-view planes, **c** results after DBSCAN clustering, **d** curve fitting, **e** results

both get good results in *Clip Cordova1* but perform not well in *Clip Cordova2*. Because there are many different pavement types and the sun is facing the vehicle, the performance of IPM and initial parameters are the keys to correct detection. Once the surface of road is painted with other marks or the extrinsic parameters of camera are changed, the accuracy of detection would decrease.

Compared with LDTFE, our method applies hyperbolic model to further fit the obtained parametric equations of lines. Because of a lot of curvatures in *Clip Cordova1*, our method performs better in these curvilinear lanes. Like Aly's (2008) problems in *Clip Cordova2*, a lot of FP in the experiment is due to the fact that there is no lane line but curb on the right side of the vehicle in images. In the process of line detection, road shoulder stone is mistaken for lane because its direction and curvature are the same as lane and it has strong edge, which leads to the false detection as the lane line. We think this kind of false detection is acceptable. In terms of total data, we have less FN than LDTFE.

As shown in Fig. 15, the first row is some false detection in LDTFE. LDTFE algorithm performs not very well in lane detection for *Clip Cordova1*, in which the lanes have a lot of curvature. LDTFE method cannot generate very accurate small segments, because it will be more difficult to process the curved part as a straight line. The core of LDTFE is mainly based on modified Hough transform line detection, so it is not very good at matching the lane with large curvature. In our method, the weighted hyperbolic model is used to match the lane lines, which is more suitable for dealing with large curvature. Secondly, LDTFE only focuses on the clustering of the left and the right lanes in the clustering process. When one lane line is a dashed line and the line outside it is a solid line, which may have more small line segments on the edge. Its feature may be more suitable for the constraints proposed in Niu's paper, so the inner dashed line is filtered out.

Table 3 The performances of different detection methods under Caltech dataset

Clips	Frames	Aly (2008)			Ruyi et al. (2011)			Shin et al. (2015)			LDTFE Niu et al. (2016)			Ours		
		AR	FP	FN	AR	FP	FN	AR	FP	FN	AR	FP	FN	AR	FP	FN
Cordova1	250	97.2	3.0	–	97.4	–	1.3	**98.3**	–	–	92.7	–	5.4	**98.3**	1.9	0.6
Cordova2	406	96.2	38.4	–	91.1	–	5.7	91.3	–	–	**97.7**	–	1.8	96.6	19.4	0.2
Washington1	337	96.7	4.7	–	**97.8**	–	1.2	88.2	–	–	96.9	–	2.5	97.1	2.2	1.4
Washington2	232	95.1	2.2	–	97.3	–	1.4	93.3	–	–	**98.5**	–	1.7	97.6	1.5	0.9
Total	1225	96.3	11.6	–	95.9	–	–	92.8	–	–	96.5	–	2.7	**97.5**	6.02	0.8

Bold values indicate the best results

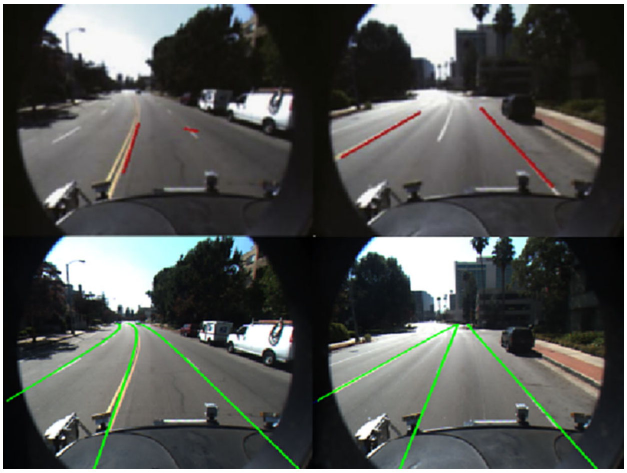


Fig. 15 Comparison between LDTFE and our method. The first row is LDTFE's false detection. The second row is our performance on these frames

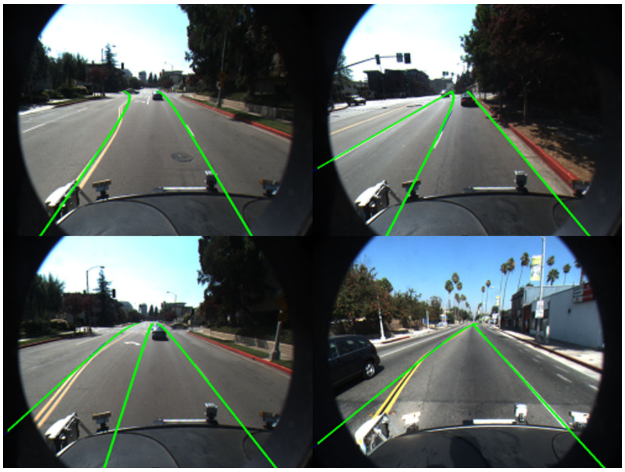


Fig. 16 Some examples of false detection of our method

5.4 Limitations

Of course, our method has some limitations. Figure 16 shows some false detections in our method. Because the lane width sometimes changes when vehicles enter or exit junctions, the lane on one side forms into S-type. However, the hyperbola cannot fit it well. This is the defect of the model. But in most cases hyperbola can fit the lane shape well. Secondly, since the color features are not applied, we cannot filter out the edge of the road shoulder stone, which leads to a large number of false positive detection. But we think that this kind of mistake is acceptable, because the edge of the road also reflects the shape of the lane.

6 Conclusion

We propose a robust lane detection method, which uses straight line detection in near field and curve matching in far field to get the lane line. In near field, the line segments of the region of interest on the source image are obtained by LSD algorithm and projected to IPM image. The line segments are clustered by DBSCAN using the idea of polar coordinates. We design an extraction method to get the optimal fused line combination and project it back to the source image. In far field part, we design a weighted hyperbolic model to fit the lane line. The experimental results achieved an average accuracy of 97.5% with a speed of 31 frames per second, which means the proposed algorithm can achieve good detection results and satisfy real-time requirement.

In future work, we consider further optimizing the performance of line segments clustering. Some filtering methods are also considered to apply to solve the interference of curb. We also plan to use other curve models to better describe the lane shape. The next challenge is to recognize the type of lanes.

Acknowledgements The work described in this paper was funded by Science and Technology Development Plan of Jilin Province (20170204020GX) and National Natural Science Foundation of China under Grant U1564211 and 51805203.

Compliance with ethical standards

Conflict of interest We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

Aly M (2008) Real time detection of lane markers in urban streets. In: IEEE intelligent vehicles symposium, proceedings, pp 7–12

Aly M (2009) Caltech lanes dataset. http://vision.caltech.edu/malaa/datasets/caltech-lanes. Accessed 10 May 2017

An X, Shang E, Song J et al (2013) Real-time lane departure warning system based on a single FPGA. Eurasip J Image Video Process. https://doi.org/10.1186/1687-5281-2013-38

Andrade D, Adamshuk R, Omoto W et al (2017) Lane detection using orientation of gradient and vehicle network signals. SAE Tech Pap. https://doi.org/10.4271/2017-01-0042.copyright

Chiu KY, Lin SF (2005) Lane detection using color-based segmentation. In: IEEE intelligent vehicles symposium, proceedings, pp 706–711

Cicchino JB (2017) Effects of lane departure warning on police-reported crash rates. J Saf Res. https://doi.org/10.1016/j.jsr.2018.05.006

Grompone von Gioi R, Jakubowicz J, Morel J-M, Randall G (2012) LSD: a line segment detector. Image Process Line. https://doi.org/10.5201/ipol.2012.gjmr-lsd

Hahsler M, Piekenbrock M (2017) dbscan: density based clustering of applications with noise (DBSCAN) and related algorithms. R Packag. version

Hsiao PY, Yeh CW, Huang SS, Fu LC (2009) A portable vision-based real-time lane departure warning system: day and night. IEEE Trans Veh Technol. https://doi.org/10.1109/tvt.2008.2006618

Jung S, Youn J, Sull S (2015) Efficient lane detection based on spatiotemporal images. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/tits.2015.2464253

Kant S, Mahara T, Kumar Jain V et al (2018) LeaderRank based k-means clustering initialization method for collaborative filtering. Comput Electr Eng. https://doi.org/10.1016/j.compeleceng.2017.12.001

Lan M, Rofouei M, Soatto S, Sarrafzadeh M (2009) SmartLDWS: a robust and scalable lane departure warning system for the smartphones. In: IEEE conference on intelligent transportation systems, proceedings, ITSC, pp 108–113

Lee C, Member S, Moon J (2018) Robust lane detection and tracking for real-time applications. IEEE Trans Intell Transp Syst 1:1–6

Liu S, Cheng X, Fu W et al (2014) Numeric characteristics of generalized M-set with its asymptote. Appl Math Comput 243:767–774. https://doi.org/10.1016/j.amc.2014.06.016

Liu S, Pan Z, Song H (2017) Digital image watermarking method based on DCT and fractal encoding. IET Image Process 11:815–821. https://doi.org/10.1049/iet-ipr.2016.0862

Liu G, Liu S, Muhammad K et al (2018) Object tracking in vary lighting conditions for fog based intelligent surveillance of public spaces. IEEE Access. https://doi.org/10.1109/access.2018.2834916

Maini R, Aggarwal DH (2009) Study and comparison of various image edge detection techniques. Int J Image Process 3:1–11

Mammeri A, Boukerche A, Tang Z (2016) A real-time lane marking localization, tracking and communication system. Comput Commun 73:132–143. https://doi.org/10.1016/j.comcom.2015.08.010

McCall JC, Trivedi MM (2006) Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. IEEE Trans Intell Transp Syst 7:20–37

Narote SP, Bhujbal PN, Narote AS, Dhane DM (2018) A review of recent advances in lane detection and departure warning system. Pattern Recognit. https://doi.org/10.1016/j.patcog.2017.08.014

Niu J, Lu J, Xu M et al (2016) Robust lane detection using two-stage feature extraction with curve fitting. Pattern Recognit 59:225–233

Ozgunalp U, Fan R, Ai X, Dahnoun N (2017) Multiple lane detection algorithm based on novel dense vanishing point estimation. IEEE Trans Intell Transp Syst 18:621–632. https://doi.org/10.1109/tits.2016.2586187

Piao J, Shin H (2017) Robust hypothesis generation method using binary blob analysis for multi-lane detection. IET Image Process 11:1210–1218. https://doi.org/10.1049/iet-ipr.2016.0506

Ruyi J, Reinhard K, Tobi V, Shigang W (2011) Lane detection and tracking using a new lane model and distance transform. Mach Vis Appl. https://doi.org/10.1007/s00138-010-0307-7

Sahu B, Kumar Sa P, Bakshi S, Sangaiah AK (2018) Reducing dense local feature key-points for faster iris recognition. Comput Electr Eng 70:939–949

Saudi A, Teo J, Sulaiman J (2008) Fast lane detection with randomized hough transform. In: 2008 international symposium on information technology, pp 1–5

Shin BS, Tao J, Klette R (2015) A superparticle filter for lane detection. Pattern Recognit. https://doi.org/10.1016/j.patcog.2014.10.011

Song W, Fu M, Yang Y et al (2017) Real-Time lane detection and forward collision warning system based on stereo vision. IEEE Intell Veh Symp Proc 18:493–498. https://doi.org/10.1109/ivs.2017.7995766

Taubel G, Sharma R, Yang J (2014) An experimental study of a lane departure warning system based on the optical flow and Hough transform methods. WSEAS Trans Syst 13:105–115

Tian Y, Cui W, Li X, Chen L (2018) A robust lane detection method based on vanishing point estimation. Procedia Comput Sci 131:354–360

Tu C, van Wyk BJ, Hamam Y et al (2013) Vehicle position monitoring using hough transform. IERI Procedia 4:316–322. https://doi.org/10.1016/j.ieri.2013.11.045

Wang X et al (2011) Robust lane detection based on gradient-pairs constraint. In: 2011 30th Chinese control conference (CCC)

Wang J, An X (2010) A multi-step curved lane detection algorithm, based on hyperbola-pair model. In: 2010 IEEE international conference on automation and logistics, ICAL 2010, pp 132–137

Wang Y, Shen D, Teoh EK (2000) Lane detection using spline model. Pattern Recognit Lett 21:677–689. https://doi.org/10.1016/s0167-8655(00)00021-0

Wang Y, Teoh EK, Shen D (2004) Lane detection and tracking using B-Snake. Image Vis Comput 22:269–280. https://doi.org/10.1016/j.imavis.2003.10.003

Wu PC, Chang CY, Lin CH (2014) Lane-mark extraction for automobiles under complex conditions. Pattern Recognit 47:2756–2767. https://doi.org/10.1016/j.patcog.2014.02.004

Yang T, Long X, Sangaiah AK et al (2018) Deep detection network for real-life traffic sign in vehicular networks. Comput Netw. https://doi.org/10.1016/j.comnet.2018.02.026

Yu G, Wang Z, Wu X et al (2017) Efficient lane detection using deep lane feature extraction method. SAE Int J Passeng Cars Electron Electr Syst. https://doi.org/10.4271/2017-01-1970

Zhang Z (2000) A flexible new technique for camera calibration. IEEE Trans Pattern Anal Mach Intell 22:1330–1334. https://doi.org/10.1109/34.888718

Zhao K, Meuter M, Nunn C et al (2012) A novel multi-lane detection and tracking system. In: IEEE intelligent vehicles symposium, proceedings, pp 1084–1089