





Lane Detection Based on Road Module and Extended Kalman Filter

Jinsheng Xiao^{1,2}(✉) , Li Luo¹, Yuan Yao³, Wentao Zou¹,
and Reinhard Klette⁴ 

¹ School of Electronic Information, Wuhan University, Wuhan, China
xiaojs@whu.edu.cn

² Collaborative Innovation Center of Geospatial Technology, Wuhan, China

³ College of Physical Science and Technology, Central China Normal University,
Wuhan, China

⁴ School of Engineering, Computer, and Mathematical Sciences,
Auckland University of Technology, Auckland, New Zealand
reinhard.klette@aut.ac.nz

Abstract. Lane detection is already a basic component in modern vehicle control systems, with satisfying accuracy on labeled roads. There are still occasional problems of low accuracy and robustness in cases of challenging lighting, shadows, or in cases that road marking is missing. The paper proposes a new algorithm combining a model of road structure with an extended Kalman filter. Lane borders are detected by an adaptive edge detection operator based on scan lines. A new parameter space is defined to adjust the algorithm to the current lane model. All candidate lanes are extracted by voting of edge points. Road boundaries are obtained by considering various constraints. A new driveway model is specified according to roadway geometry and vehicle dynamics. The estimation of parameters is expanded for also covering driveway information. Coordinates of lane border points are tracked and estimated using an extended Kalman filter. Special attention is paid to enhancing stability and robustness of the algorithm. Results indicate that the proposed algorithm is robust under various lighting conditions and road scenarios; it is also of low computational complexity.

1 Introduction

Advanced driver assistance systems are gradually being incorporated into vehicles. Such a system can either alert the driver in dangerous situations, or take on an active part in the driving process. The systems are expected to become more and more complex towards full autonomy during the next decade. The perception problems are the main bottleneck in the development of such systems.

One of the perception problems is road and lane detection, tracking, and analysis, and another one is obstacle detection [1–3]. We consider the first one in this paper. The main perceptual cues for human driving include road color and texture, road boundaries, and lane markings. Autonomous vehicles are expected to share the road with human drivers, at least for some time, and will therefore

most likely continue to rely on the same perceptual cues that human drivers do. It is unrealistic to expect the huge investments required to construct and maintain special infrastructure for autonomous vehicles only. Road and lane perception via traditional cues remains the most likely scenario for autonomous driving; see, for example, [1, 3, 4].

The road and lane detection task can be broken down into five functional modules. They are image pre-processing, feature extraction, road/lane model fitting, temporal integration, and image-to-world correspondence; see [1, 5, 6]. Yu and Jain [7] proposed in 1997 a method based on multi-resolution Hough transform to detect lane boundaries. The geometric information of the lane was used to limit the parameter range of the Hough transform. Hough-transform based methods are still in use today for well-marked highway driving. Lee [8] proposed a lane departure warning system to estimate the subsequent direction of a lane through an *edge distribution function* (EDF) and direction changes of vehicle movement. The EDF failed on roads with curved and dashed lanes. Kang and Jung [9] estimated the initial position of a road boundary based on edge direction and amplitude of the lane based on assuming a flat road surface.

The analysis of connected components and an energy density function in dynamic programming are used to obtain the best location of a lane. Wang et al. [10] provided an initial position for a B-snake model in 2004. The lane detection problem can be changed into the problem of control points to determine a spline curve following a road model. In 2010, Zhou and Jiang [11] estimated the parameters from the main direction and the edge direction, and the best lane model was selected by a Gabor filter. In 2012, Mechat [12] detected a lane using a support vector machine (SVM) based method; the model of the lane was defined by a Catmull-Rom curve, and the standard Kalman filter was adopted to estimate and track the parameters of control points. Kortli et al. [13] established a region of interest (ROI) in road images, used a Gauss filter for data pre-processing, and then applied the Canny edge detector to enhance lane boundaries; a method was proposed to extract lane boundaries based on color information and image segmentation by using a histogram threshold, Hough transform; the current vehicle position was obtained. Shin et al. [14] extended a particle-filter-based approach for lane detection, also addressing challenging road situations. Lee et al. [15] proposed a real-time lane detection and tracking algorithm using filters (e.g. a Kalman filter) for an embedded lane departure warning system (LDWS). Lane detection modules provide currently already stable results in general, but their performance under special (i.e. challenging) conditions is still a research topic; those conditions might be defined by strong sunlight, hard to identify lanes (e.g. missing lane marks, confusing marks on the road, or cars parked along a road), shadows caused by trees or other objects, sidewalks, zebra crossings, or text logos on the road.

In view of this, we propose a lane detection algorithm by combining a road structure model with an extended Kalman filter. Considering the characteristics of the lane and according to the roadway geometry and vehicle dynamics, a new lane model is proposed which enhances the stability and anti-jamming of the lane-detection system. The parameter space is defined to accommodate the

algorithm for the lane model. Due to algorithmic developments in the area of Hough transform [16], there is good progress to improve the processing speed. The extended Kalman filter is used to estimate and track the lane, which is the major factor for improved accuracy in the proposed lane detection. The effectiveness and robustness of the algorithm are demonstrated in this paper.

The paper is structured as follows. Section 2 reports about the used lane (or road) and vehicle model. Section 3 describes our combination of road model and of an extended Kalman filter. Section 4 informs about our experiments and the performed evaluation. Section 5 concludes.

2 Processing in the Basic Layer

In this paper, estimated parameters for lane detection are related to lane shape and the pose of the *ego-vehicle* (i.e. the vehicle where the lane-detector operates in). These parameters involve lane width, lane curvature, and the motion of the ego-vehicle (described, e.g., by the pitch angle or horizontal angle).

2.1 Algorithm Outline

Figure 1 shows the flowchart of our lane detection algorithm. Parameter estimation is divided into two parts. The first part is the *pose* (i.e. position and viewing

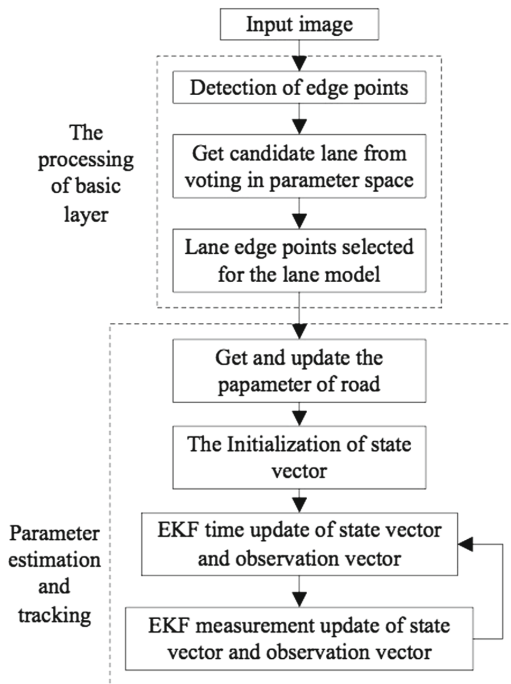


Fig. 1. Schematic diagram of lane detection algorithm

direction) of the camera and the road environment information obtained from the in-vehicle camera. The second part estimates the parameters of the road using the road lane information. The flowchart of our algorithm, presented in this paper, begins with reading a frame of the recorded video sequence. The adaptive edge detection operator, based on scan line processing, is used to extract the edge points from the R channel of the image. We select a region of interest in the algorithm. The image pixel coordinate system is mapped to the road plane coordinate system. Edge points are voted in a customized parameter space to obtain candidate lines. We collect geometrical characteristics and estimate spatial continuity of the lane. The information of the lane in the image pixel coordinate system, and the lane road plane coordinate system is used to exclude non-lane lines. Inner boundary points and an extended Kalman filter are combined to estimate and update lane model parameters.

2.2 Detection of Edge Points

Due to the perspective effect of the camera, visible information differs for different ranges in the image of the road plane. We divide the regions of interest into several areas. A preset area is selected according to the image coordinates which are converted using the vanishing point coordinates and the real coordinates. The customized edge-detection operator is based on scan-line processing for reducing the amount of computations. An image row forms a scanning line. If the number of calculated scan lines is greater than the maximum value, the scan lines are set under the real world coordinates with the same distance of each line. These lines are translated into the image pixel coordinate system. A set of scan lines is shown in Fig. 2, left. By scanning through each pixel in a selected scan line, the edge strength of each pixel is calculated by the following equation. $E(i)$ is the edge strength of a pixel in the scanning line, $I(i)$ represents



Fig. 2. *Left:* a set of scan lines. *Right:* examples of detected edge points; input image is as shown on the left

the image value of the i -th pixel, and L_i defines the neighborhood of the pixel at i :

$$E(i) = - \sum_{k=-L_i}^{-1} I(i+k) + \sum_{k=1}^{L_i} I(i+k) \tag{1}$$

Our method is not using the same neighborhood for all the pixels; the pixel neighborhood parameter L_i , also called the *scale* here, is chosen adaptively for edge detection; it may change based on the position of the pixel in the image.

We consider the perspective effect of the lane in the road image. The length that each pixel represents in world coordinates on a scanning line is computed after coordinate conversion. The width of the lane is used to be divided by the proportion for each scan line. Then, the pixels can be identified which are needed to compute the edge strength. Pixels which have a maximum or minimum value of edge strength are *edge points*. This customized edge detection operator can greatly reduce the processing time. Edge detection results are shown in Fig. 2, right. The figure shows that the operator can detect edge points in all the lanes; accuracy and efficiency can be called “very good” in general.

2.3 Obtain Candidate Lane from Voting in Parameter Space

After edge points are detected, they need to be aggregated into a candidate lane. The amount of computations for the traditional Hough transform [17] is large but can be reduced [16]. Aiming at the lane model, the parameter space is defined conveniently for our algorithm. The edge points are voting in the parameter space. A diagram of the customized parameter space is shown in Fig. 3. The parameter space is spanned by two parameters p and q . The lane in the region of interest is defined by $x = p + y(q/d)$. Parameter p represents the position of the line on the x axis; parameter q identifies the slope, i.e. the lateral position. The flowchart of our voting algorithm includes three parts:

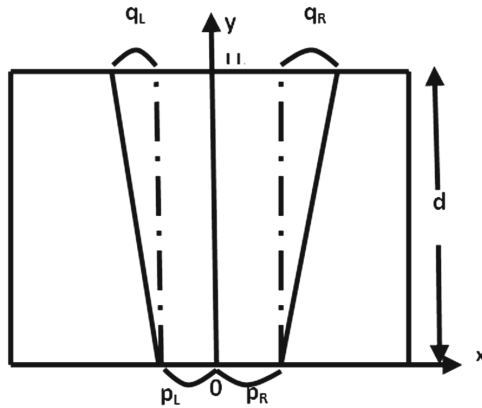


Fig. 3. Parameter space

1. The coordinates of edge points are transformed. The edge points obtained from the edge detection are in the image pixel coordinate system. The edge points need to be converted to the world coordinate system, in which the perspective effect is removed.
2. All the edge points in the selected area are traversed by parameter q . The corresponding p was obtained through the look-up table. The accumulator increases by 1. q is translated from the tilt angle of the lanes. q is discretized and the range of value is $[-88^\circ, 88^\circ]$. p is discretized and the range of value is $[-5\text{ m}, 5\text{ m}]$ in the world coordinate system.
3. Different p, q denote different lines. From the mapping relationship of the parameter space, the larger the value of the accumulator of p, q , there are more edge points in the lane which are represented by this parameter. All possible candidate lines are identified by searching for the maximum value.

The wrong candidate lines are parallel to the right lane boundary in many cases. After searching for all local maximum value, it needs to check whether any pair of candidate line is crossed. If a pair of candidate lines is crossed, the line in which has a smaller value in the parameter space will be discarded. The judgment for determining cross-line is as follows:

$$(p_A - p_B) \cdot (p_B - p_A + q_B - q_A) \geq 0 \quad (2)$$

The candidate lines can be gotten by the voting algorithm in the parameter space. The geometrical characteristics and spatial continuity of the lane are still needed to exclude the unwanted points and unwanted lanes. The dual information of lane in the image pixel coordinates system and the lane road plane coordinate system is useful too. The final candidate lanes are shown in Fig. 4 left, which detect the current two lanes accurately.

The lane edge points need to be selected for the lane model after the above steps. All the points should be selected, but only ten points are selected if the total number of edge points is greater than ten. The ten points are selected by same distances. The selected points are shown in Fig. 4 right. The perspective



Fig. 4. *Left:* candidates of lane boundary lines. *Right:* selected boundary points

effect in the road line image should be considered in the selection of boundary points which can ensure the reasonableness and accuracy of the lane tracking and estimating in the later.

3 Combining Lane Model and Extended Kalman Filter

Vehicle shaking, light changing and vehicle interference, all can lead to the jitter of the collected images. One or several frames cannot identify the lane lines due to the reasons of broken lane, dirty road, shadows, and so on. The filter algorithm is used to track the lane, which can greatly improve the ability of stability and anti-jamming for the system [18]. Therefore, the information of road structure model is combined with extended Kalman filter. The extended Kalman filter is used to track after the lane lines were detected, which greatly improved the accuracy of lane identification. The effectiveness and robustness of the algorithm are ensured too.

3.1 Lane Model

The lane model is built according to the roadway geometry and vehicle dynamics [19,20]. The road boundary model is defined as follows:

$$x_{t,k}(z) = \frac{1}{2}kW_t + e_t + \theta_t z + \frac{1}{2}c_{0,t}z^2 + \frac{1}{6}c_{1,t}z^3 \quad (3)$$

$x_{t,k}(z)$ is the position for the point of lane boundary which has a distance of z from the vehicle at time t . k denotes the left or right side of the lane with $k = -1$ or $k = 1$. e_t is the lateral offset between the center of the vehicle and the center of the lane. θ_t is the yaw angle between driving direction and the lane. $c_{0,t}$, $c_{1,t}$ are the curvature and the change rate of curvature. φ_t is the pitch angle between the optical axis and the plane of the road plane. The estimated parameters are described as follows:

$$x(t) = \left(\dot{\theta}_t, \theta_t, \dot{e}_t, e_t, c_{1,t}, c_{0,t}, \varphi_t, W_t \right) \quad (4)$$

The “.” on the letters in the formula expresses the change rate of corresponding parameters. Compared with the models [19,20], the model in this paper increases two new estimated parameters, that is, the change rate of the yaw angle $\dot{\theta}_t$ and the lateral offset \dot{e}_t . The tracking estimation parameters of lane information increase, the stability, anti-jamming and detection rate of the lane detection system can be corresponding improved.

3.2 Parameter Estimation

When the candidate lanes have been detected preliminarily, they are tracked and estimated by Kalman filter. The traditional Kalman filter is useful when the state equation and measurement equation of system are both linear system. The

system noise and measurement noise are consistent with the gaussian distribution. The parameters can be estimated by the principle of minimum mean square error. But the state equation and measurement equation of the lane model in this paper are nonlinear. The traditional Kalman filter is not practicable in this system. So the nonlinear system should be approximated by a linear problem. The Taylor series expansion is the most commonly used linearization method, which ignores the higher order term. So the Kalman filter is replaced by the extended Kalman filtering (EKF) [21]. The update processing of extended Kalman filter is shown as following:

The process has a state vector $x \in \mathbb{R}^n$. It is now governed by the non-linear stochastic difference equation:

$$\begin{cases} x(t) = f(x(t-1), u(t-1), w(t-1)) \\ y(t) = h(x(t), v(t)) \end{cases} \tag{5}$$

where the random variables $w(\cdot)$ and $v(\cdot)$ represent the process and measurement noise with zero-mean. The $u(t-1)$ is the driving function. The time update of state vector and the observation vector for extended Kalman filter are as following:

$$\begin{cases} \hat{x}_p(t) = f(\hat{x}(t-1), u(t-1), 0) \\ P_p(t) = A(t)P(t-1)A(t)^\top + W(t)Q(t-1)W(t)^\top \end{cases} \tag{6}$$

where $\hat{x}_p(t)$ is the priori state estimate at time t . $\hat{x}(t)$ is the posteriori state estimate at time t . $A(t)$ and $W(t)$ are the process Jacobians at time t . Where $P_p(t)$ is the priori estimate error covariance. $P(t)$ is the posteriori estimate error covariance. $Q(t)$ is the process noise covariance. The Kalman gain in the update of measurement is shown as following:

$$K(t) = [P_p(t)H(t)^\top] [H(t)P_p(t)H(t)^\top + V(t)R(t)V(t)^\top]^{-1} \tag{7}$$

where $H(t)$ and $V(t)$ are the measurement Jacobians at time t . $R(t)$ is the measurement noise covariance. The covariance matrix of posteriori estimation error in the measurement update is shown as:

$$P(t) = P_p(t) - K(t)H(t)P_p(t) \tag{8}$$

The updated estimation of observation vector in the measurement update is the following expression:

$$\hat{x}(t) = \hat{x}_p(t) + K(t) [y(t) - h(\hat{x}_p(t), 0)] \tag{9}$$

According to the change of traffic variables and vehicle dynamics, the system matrix can be obtained as follows [19]:

$$A(t) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Delta t & 1 & 0 & 0 & \frac{\Delta t^2 \cdot V^2}{2} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \Delta t^2 \cdot V^3 & 0 & 0 & 0 \\ 0 & 0 & \Delta t & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Delta t \cdot V & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$

where V is the vehicle’s speed. The observation vector is set as x_{road} . Because the estimation points which are selected to estimate parameters are image pixels, the points need to be converted:

$$\begin{cases} y_{img} = -f \frac{h}{z_{road}} - f \cdot \varphi_t \\ x_{img} = f \frac{x_{road}}{z_{road}} \end{cases} \tag{11}$$

In keeping with road plane, x_{img} is image’s ordinate and y_{img} is abscissa. The Jacobian matrix for the partial derivatives of h respect to x is defined as follows:

$$\begin{aligned} H_{[i,j]} &= \frac{\partial h_{[i]}}{\partial x_{[j]}} (\hat{x}_k, 0) \\ &= \left(\frac{\partial x_{img}}{\partial x[0]}, \frac{\partial x_{img}}{\partial x[1]}, \frac{\partial x_{img}}{\partial x[2]}, \frac{\partial x_{img}}{\partial x[3]}, \frac{\partial x_{img}}{\partial x[4]}, \frac{\partial x_{img}}{\partial x[5]}, \frac{\partial x_{img}}{\partial x[6]}, \frac{\partial x_{img}}{\partial x[7]} \right) \end{aligned} \tag{12}$$

So, from (11) there are:

$$\begin{cases} \frac{\partial x_{img}}{\partial x[i]} = \frac{f}{z_{road}} \cdot \frac{\partial x_{road}}{\partial x[i]}, & i = 0, 1, \dots, 5, 7 \\ \frac{\partial x_{img}}{\partial x[6]} = \frac{\partial x_{img}}{\partial z_{road}} \cdot \frac{\partial z_{road}}{\partial x[6]} \end{cases} \tag{13}$$

From (3), we obtain that

$$x_{road} = \frac{1}{2}k W_t + e_t + \theta_t z_{road} + \frac{1}{2}c_{0,t} z_{road}^2 + \frac{1}{6}c_{1,t} z_{road}^3 \tag{14}$$

Note the definition of $x(t)$ in (4), we have $x(t) = (\dot{\theta}_t, \theta_t, \dot{e}_t, e_t, c_{1,t}, c_{0,t}, \varphi_t, W_t)$, so:

$$\begin{aligned} &\left(\frac{\partial x_{road}}{\partial x[0]}, \frac{\partial x_{road}}{\partial x[1]}, \frac{\partial x_{road}}{\partial x[2]}, \frac{\partial x_{road}}{\partial x[3]}, \frac{\partial x_{road}}{\partial x[4]}, \frac{\partial x_{road}}{\partial x[5]}, \cdot, \frac{\partial x_{road}}{\partial x[7]} \right) \\ &= \left(0, z_{road}, 0, 1, \frac{1}{6} z_{road}^3, \frac{1}{2} z_{road}^2, \cdot, \frac{1}{2} k \right) \end{aligned} \tag{15}$$

with

$$\frac{\partial x_{road}}{\partial z_{road}} = \theta_t + c_{0,t}z_{road} + \frac{1}{2}c_{1,t}z_{road}^2 \quad (16)$$

So, from Eqs. (11), (14), and (16) we have that

$$\begin{aligned} \frac{\partial x_{img}}{\partial z_{road}} &= \frac{f}{z_{road}^2} \cdot \left(z_{road} \cdot \frac{\partial x_{road}}{\partial z_{road}} - x_{road} \right) \\ &= \frac{f}{z_{road}^2} \cdot \left(\frac{1}{2}c_{0,t}z_{road}^2 + \frac{1}{3}c_{1,t}z_{road}^3 - \frac{1}{2}kW_t - e_t \right) \end{aligned} \quad (17)$$

According to the relationship between angle and ordinate in image, from (11) it can be gotten that $z_{road} = \frac{-f \cdot h}{y_{img} + f \cdot \phi_t}$. Note that $x[6]$ is ϕ_t in (4), so:

$$\frac{\partial z_{road}}{\partial x[6]} = \frac{\partial z_{road}}{\partial \phi_t} = \frac{f^2 \cdot h}{(y_{img} + f \cdot \phi_t)^2} \quad (18)$$

Then

$$H_{[i,j]} = \left(0, f, 0, \frac{f}{z_{road}}, \frac{f \cdot z_{road}^2}{6}, \frac{f \cdot z_{road}}{2}, \frac{\partial x_{img}}{\partial z_{road}} \frac{\partial z_{road}}{\partial x[6]}, \frac{k \cdot f}{2 \cdot z_{road}} \right) \quad (19)$$

The 7th item in the Jacobian matrix H can be obtained by the above formula. The parameters of lane model can be expressed by the state vector and the covariance matrix of estimation error. The abscissas of the lane boundary points in the original input image selected from the above steps, are gotten by the extended Kalman filter to conduct iterative estimation. If the extracted lane candidate boundary points include the noise introduced by observation, but not include wrong identification points, then the unit of standard deviation for noise is pixel. If the value of the variance for estimation error is σ^2 , then the maximum change of each parameter is 3σ . Assuming that the standard deviation of x_{img} is σ , the search area of candidate points is $(x_{img} - 3\sigma, y_{img}) \sim (x_{img} + 3\sigma, y_{img})$ for the next time. The selection of lane curvature can be narrowed down, too.

4 Experiments and Analysis

The numerous experiments were implemented to verify the validity of the proposed algorithm with the programming development environment VS2005 on the platform of Intel Core i5-3470 K processor at 3.2 GHz 4 GB cache. The operating system is Windows XP. The algorithm is based on monocular vision, and the road images obtained via an industrial camera: Basler pia1900-32 gm/gc. The focal length is 8 mm. The maximum frame rate of image is 32 frames per second. The size of image is $1,920 \times 1,080$ in pixels. Images obtained from camera are transferred to the computer through a Gigabit Ethernet port.

To verify the effectiveness and robustness of the proposed algorithm, the road video data captured from Erdos in Inner Mongolia is selected to test. These data

Table 1. Detection efficiency for RANSAC and proposed algorithm

Video index	Total number of frames	Number of correctly detected frames		Recognition rate	
		Proposed algorithm	RANSAC algorithm	Proposed algorithm	RANSAC algorithm
1	4060	4028	3947	99.21%	97.22%
2	3660	3610	3519	98.63%	96.15%
3	3120	3101	3017	99.39%	96.70%
4	3540	3509	3368	99.12%	95.14%

include a variety of road conditions, such as trees shadow, pedestrian, vehicle interference, strong light and shadow, curve and so on. Test results of four video sequences are shown as example. The comparison algorithm is RANSAC algorithm based on template [3]. Table 1 is the statistics result for the lane detection, which include the total number of the test frames and the recognition rate. From the data in Table 1, the recognition rate of the proposed algorithm in this paper is 2% to 4% higher than the RANSAC algorithm. The algorithm has good stability and high recognition rate up to 99.39%. The proposed algorithm in this paper has good accuracy and robustness for the structural road environment. The state equation and measurement equation of the lane model here are nonlinear, which are not suitable for the traditional Kalman filter. So the nonlinear system should be approximated by a linear problem. The extended Kalman Filter is used to estimate and track the road lane, which improved the accuracy of the lane detection. Due to the extended Kalman filter, the detection result is very stable and the problems of jitter and low fitting precision in traditional algorithm are missed.

Next, it is necessary to discuss the robustness of the lane detection algorithm under different road environment. The comparative analysis of lane detection effect for complex road environment are shown in Figs. 5, 6, 7 and 8. It includes lane-changing, curve, zebra crossing, strong light and shadow trees, vehicles, etc.

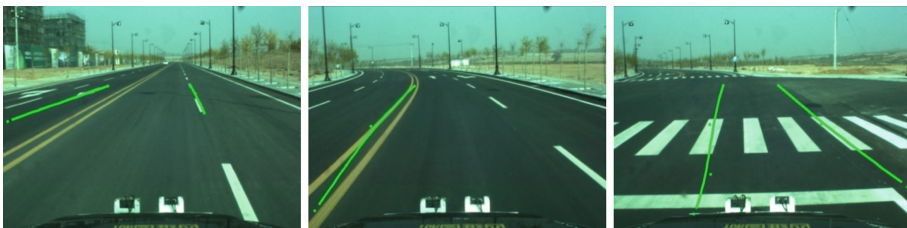


Fig. 5. Detection results for RANSAC. *Left to right:* lane-changing, curve, zebra crossing

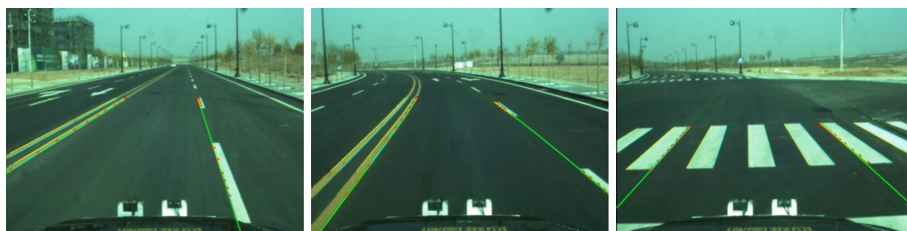


Fig. 6. Detection results of the proposed method. *Left to right:* lane-changing, curve, zebra crossing



Fig. 7. Detection results of RANSAC. *Left to right:* vehicle, strong light, shadow

In Fig. 5, the RANSAC algorithm has mistaken identification during changing lanes, while the proposed algorithm solves the problem well in Fig. 6. The RANSAC algorithm has the problem of instability and inaccuracy fitting in curve lane, while the proposed algorithm still has high stability and robustness. In the case of closing to the zebra crossing, the RANSAC algorithm makes a mistake in detection, but the proposed algorithm has a better result.

In Fig. 7 the length of right lane detected by RANSAC algorithm is greatly reduced because of the vehicle, while the detection result of the proposed algorithm is still very accurate in Fig. 8, although there are lots of mistaken edge points. With the strong light, the RANSAC algorithm detected the right lane only, while both lanes are detected by the proposed algorithm. RANSAC algorithm can detect the left lane correctly which has less shadow, but it is unable to identify right lane with serious shadow. The proposed algorithm can accurately detect it with high robustness.



Fig. 8. Detection results for the proposed method. *Left to right:* vehicle, strong light, shadow

From the comparison of detection results above in various complex road conditions, such as lane-changing, curve, the zebra crossing, strong light, shadow trees and vehicles, etc. The efficient and stability of the proposed algorithm are higher than the RANSAC algorithm. The proposed algorithm in this paper can accurately identify the lane for various special road conditions. At the same time, it can achieve real-time processing speed (20 frames per second), which has higher recognition rate and reliability compared with RANSAC algorithm.

Of course, the proposed algorithm also has some shortcomings. For example, in Fig. 9, when the vehicle changes the lane and there are long interference lines which are parallel to the lanes at the same time, there will be a mistaken detection. Further research will be needed later.



Fig. 9. False detection results for the proposed method

5 Conclusions

The paper proposes a new lane detection algorithm based on monocular vision. The algorithm uses the structure of the lane, roadway geometry, and vehicle dynamics; those components have not been fully combined before by traditional methods. A new driveway model is introduced with an increased number of parameters of driveway information to be evaluated. A customized parameter space, which is suitable for the proposed algorithm, is established. The traditional Hough transform is improved in the proposed algorithm for improved processing speed. The combination of the lane model with an extended Kalman filter for lane detection guarantees effectively the stability of the algorithm. This also enhances accuracy for lane fitting. Experiments show that the algorithm has good recognition rates and robustness in various challenging lane environments. The algorithm does not yet provide high confidence levels for some of the challenging lane detection situations. This requires further studies.

Acknowledgement. This work is supported by National Natural Science Foundation of China (Grant No. 61471272), Natural Science Foundation of Hubei Province, China (Grant No. 2016CFB499).

References

1. Hillel, A.B., Lerner, R., Levi, D., Raz, G.: Recent progress in road and lane detection: a survey. *Mach. Vis. Appl.* **25**(3), 727–745 (2014)
2. Huang, A.S., Moore, D., Antone, M., Olson, E., Teller, S.: Finding multiple lanes in urban road networks with vision and LIDAR. *Auton. Robots* **26**, 103–122 (2009)
3. Aly, M.: Real time detection of lane markers in urban streets. In: *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 7–12 (2008)
4. Xiao, J.-S., Cheng, X., Li, B.-J., et al.: Lane detection algorithm based on beamlet transformation and k-means clustering. *J. Sichuan Univ. (Eng. Sci. Ed.)* **47**(4), 98–103 (2015). (in Chinese)
5. Meng, L.-X., Sun, F.-C., Shao, Y.: Survey on road image interpretation based on monocular vision. *J. Comput. Appl.* **30**(6), 1552–1555 (2010). (in Chinese)
6. Xu, H.-R., Wang, X.-D., Fang, Q.: Structure road detection algorithm based on B-spline curve model. *Acta Automatica Sinica* **37**(3), 270–275 (2011). (in Chinese)
7. Yu, B., Jain, A.K.: Lane boundary detection using a multiresolution Hough transform. In: *Proceedings of IEEE International Conference on Image Processing*, pp. 748–751 (1997)
8. Lee, J.W.: A machine vision system for lane departure detection. *Comput. Vis. Image Underst.* **86**(1), 52–78 (2002)
9. Kang, D.J., Jung, M.H.: Road lane segmentation using dynamic programming for active safety vehicles. *Pattern Recogn. Lett.* **24**, 3177–3185 (2003)
10. Wang, Y., Teoh, E., Shen, D.: Lane detection and tracking using B-Snake. *Image Vis. Comput.* **22**, 269–280 (2004)
11. Zhou, S., Jiang, Y.: A novel lane detection based on geometrical model and Gabor filter. In: *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 59–64 (2010)
12. Mechat, N., Saadia, N., M'Sirdi, N.K., Djelal, N.: Lane detection and tracking by monocular vision system in road vehicle. In: *Proceedings of IEEE International Conference on Image, Signal Processing*, pp. 1276–1282 (2012)
13. Kortli, Y., Marzougui, M., Atri, M.: Efficient implementation of a real-time lane departure warning system. In: *Proceedings of IEEE International Conference on Image Processing Applications Systems*, pp. 1–6 (2016)
14. Shin, B., Tao, J., Klette, R.: A superparticle filter for lane detection. *Pattern Recogn.* **48**, 3333–3345 (2014)
15. Lee, D., Shin, J., Jung, J., et al.: Real-time lane detection and tracking system using simple filter and Kalman filter. In: *Proceedings of IEEE International Conference Ubiquitous Future Networks*, pp. 275–277 (2017)
16. Xu, Z., Sin, B., Klette, R.: Closed form line-segment extraction using the Hough transform. *Pattern Recogn.* **48**, 4012–4023 (2015)
17. Hough, P.V.C.: Method and means for recognizing complex patterns. US Patent, pp. 77–79 (1962)
18. Xiao, J., Liu, T., Zhang, Y., et al.: Multi-focus image fusion based on depth extraction with inhomogeneous diffusion equation. *Signal Process.* **125**, 171–186 (2016)
19. Dickmanns, E.D., Mysliwetz, B.D.: Recursive 3D road and relative ego-state recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 199–213 (1992)
20. Watanabe, A., Naito, T., Ninomiya, Y.: Lane detection with roadside structure using on board monocular camera. In: *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 191–196 (2009)
21. Welch, G., Bishop, G.: An introduction to the Kalman filter. UNC, Chapel Hill, TR (2006)