

Robust and Real-Time Lane Marking Detection for Embedded System

Yueting Guo¹, Yongjun Zhang², Sha Liu¹, Jun Liu¹,
and Yong zhao¹(✉)

¹ School of Electronic and Computer Engineering,
Shenzhen Graduate School of Peking University, Shenzhen, China
{guoyueting, liusha, liujun}@sz.pku.edu.cn,
zhaoyong@pkusz.edu.cn

² College of Computer Science and Technology,
Guizhou University, Guiyang 550025, China
zyj6667@126.com

Abstract. Lane marking detection is part of most advanced driver assistance systems (ADAS) as an important component of computer vision. This paper presents a lane detection system based on a novel lane feature extraction approach. The robustness and real-time of algorithm enable different configurations of embedded solutions. The system is divided into three phases. Firstly, using the Prewitt operator we can get the rich useful details and using Shen Jun operator we can get step edge, on the other hand Shen Jun operator is the best filter to detect the symmetrical markings according to the maximum signal noise ratio (SNR) criterion. So we introduce the best compromise method between noise smoothing and edge locating that combining the Prewitt operator with Shen Jun operator to extract lane markings. Then a fast Hough transform based on image pyramid is applied to get the lane lines. The posterior algorithm of reasonably refining the Lane lines angle is introduced to correct to error caused by Hough transform. Finally, robust detection of vehicle's departure warning is also discussed. Experiment results on real road will be presented to prove the robustness and effectiveness of the proposed lane detection algorithm.

Keywords: Shen Jun operator · Fast Hough transform · Warning of lane departure

1 Introduction

Traffic accidents have become one of the most serious problems in our live today because of driver careless negligence, languor and distraction. Therefore many works have been done to investigate improved ways of driving-assistance system to enhance vehicle driving safety. Driver assistance system (DAS) being capable of detecting the lane markings and warning the driver when the vehicle begins to move out of its lane is desperately needed to eliminate these accidents.

Throughout the last two decades, different approaches for lane analysis were proposed, demonstrated and improved. This topic can be separated into three essential blocks: lane marking detection, lane tracking and lane departure warning. For the lane

marking detection algorithm, the existing solutions can be classified as feature-based [1, 3, 4], model-based [2, 5] and region-based [1, 6] methods. Among these methods, different feature-based methods for Lane detection are still the most popular and commonly used ways which contribute to its efficiency in multiple algorithms. There are many related research works on the lane detection methods. Firstly, the determination of the region of interest (ROI) can reduce the complexity and the unnecessary information caused by noises. Isolating the ROI from other region before the computation [6] can reduce processing time while keeping a good performance. The fixed size of ROI may not work well, so some adaptive road ROI determination algorithms are proposed to detect the road region using the positional information of a vanishing point [7] or the position of the heading vehicles [4]. Secondly, extraction algorithms of the lane features have been widely studied based on color [9], template matching [13] or edge features [8, 12, 13]. Edge-based techniques not only work well with solid and segmented lines but also can compensate for circular reflectors. However, many of the proposed methods are not suitable for the accurate lane marking points extraction in the situations that contain many extraneous lines. Thirdly, various shapes of mathematical models such as parabolas [2, 14, 15] or splines [10, 11] have been used to match the lane lines. While road modeling can eliminate false positions via outlier removal, they might not be a suitable choice for variety of road conditions. Not to mention the fact that a stable control system only require approximate 10-m lookahead, making a linear road model satisfactory. Then, a sensing system [1] such as GPS or other sensors may be used to gather information about the vehicle's environment or augment lane-position estimates. In this paper, we only focus on the camera to get the images of lane information. Finally, various of tracking methods such as kalman filter [8, 12], particle filter [16, 17] have been used to track the lane line.

However, many of these algorithms do not work well in realtime and robustness or the results are shown on a limited scenarios with no obstacles present. Considering that DAS are implemented on battery powered embedded platforms inside a car, we present a robust and real-time lane detection system which shows robust results in a variety of road conditions.

Following the introduction, the algorithm used to detection lane marking is detailed organized as follows. Section 2 presents the image preprocessing method including selection of ROI and image gray-enhancing conversion method. Section 3 proposes lane feature extraction algorithm. Section 4 illustrates the lane lines detection algorithm and vehicle's Departure warning algorithm. Section 5 describes some experimental results under various illumination conditions to confirm the robustness of the proposed algorithm. Section 6 concludes the paper and discusses future works.

2 The Preprocessing of Image

The selection of ROI minimizes the detection of extraneous objects while maximizing the likelihood of detecting lane markers. Both the speed and the accuracy can be improved by cropping and shrinking the original image. At present, a portion of the image that is above the hood of the vehicle and below the vanishing point is selected as the ROI. In order to reduce the compute time we just choose the G channel of the RGB

to get the gray image. This method save much time but has no influence to the final result of experiment. Then histogram equalization is applied After getting the gray scale image to enhance the contrast of image. The images are obtained by applying Histogram equalization as illustrated in Fig. 1. Then shrinking the region of interest to half of the original ROI by taking the maximum value of each four-pixel box as the current pixel value. Trough shrinking the ROI, we find that the accuracy of detection does not decrease in the final experimental results, but the detection time reduces.

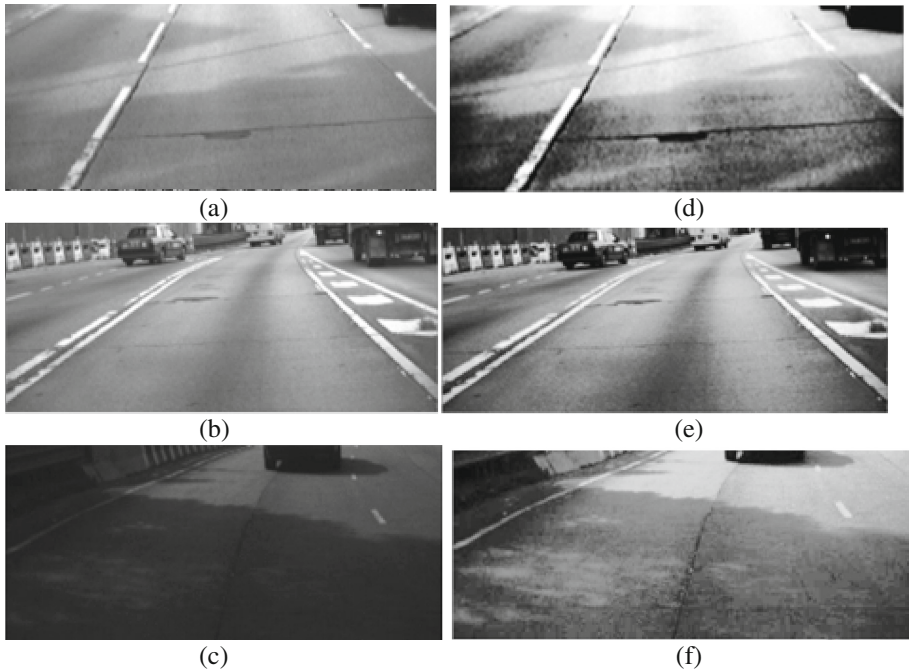


Fig. 1. The ROI of original gray image and histogram equalization image. (a, b, c) Are ROI gray images, (d, e, f) are histogram equalization images.

3 Lane Marking Points Extraction

Road lane points extraction is one of the key steps in real-time lane analysis, which decides the accuracy of the entire lane analysis system. The robustness, and hence the efficiency, of these points extraction step is dependent on vehicle surround conditions such as road types, shadows, road surface etc. Figure 2 shows the framework of lane points extraction. Firstly, the input image (Orgimg) is smoothed with a 5×5 Prewitt operator to remove background noises and get image (Grandimg). In order to eliminate the influence of illumination, a adaptive threshold is used on the smoothed image to get image (Mskimg). Image (ShunJunimg) that got by Shun Jun edge operator is used to restrict the threshold image (Mskimg) and get image (Laneimg). The following sections are the detail component of this improved system.

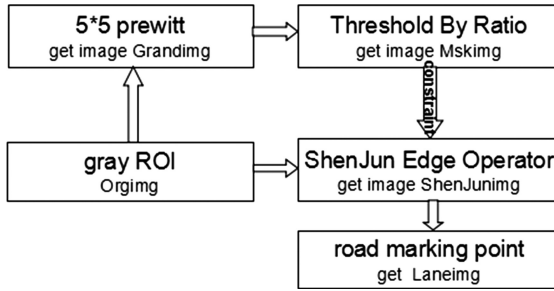


Fig. 2. Framework of lane points extraction.

3.1 Prewitt Operator and Adaptive Threshold

A 5×5 Prewitt operator is used to smooth the gray image and find initial edge pixels inside the ROI. Then the adaptive threshold is used to extract the lane marking points. In order to reduce the shadows influence, before the adaptive threshold, we segment the smoothed image Grandimg into six small blocks whose width is the three-fifth of the length of ROI width and height is one-third of the length of ROI height. $(x1, y1)$ and $(x2, y2)$ are one block's starting coordinates and end coordinates, as shown in Fig. 3(b). we propose the adaptive threshold method shown in Algorithm 1.

Algorithm1 Adaptive Threshold

```

1: read input image Grandimg block
2: get the 256-level gray histogram
3: Histogram[i]=the pixel count of the i level
4: P0=pixel value after threshold
5: Count = 0; i = 255
6: If (i > 0) then
7: Count = count + histogram[i]
8: If (count>sum) then
9: Threshold = i ;break
10: else
11: i=i-1
12: end if
13: end if
14: For(each pixel) then
15: If (the pixel value>threshold ) then
16: Po=255
17: else
18: P0=0
19: end if
20: end for
  
```

where $\text{sum} = (x_2 - x_1)(y_2 - y_1) \times R$, R is the ratio of the number of possible road lane points unit area. This value can be got through experiment.

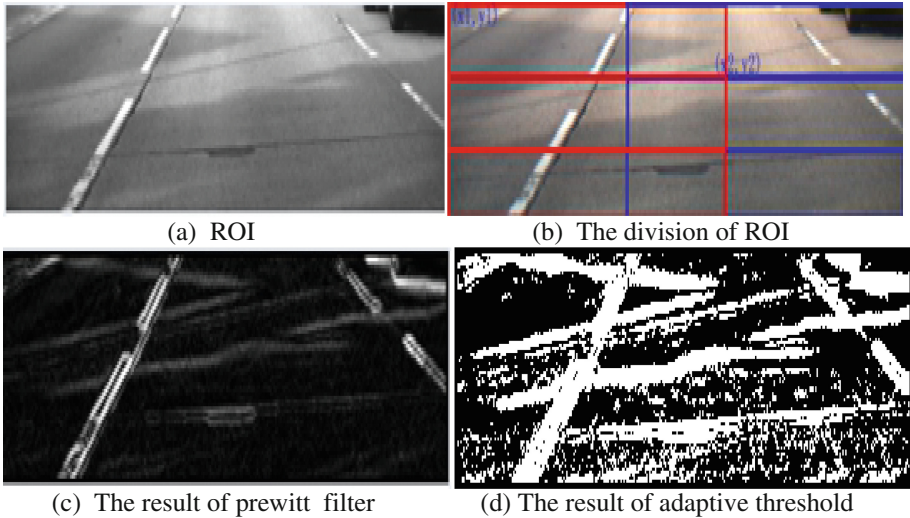


Fig. 3. The division of image and the result of threshold.

3.2 Shen Jun Edge Operator

Shen Jun edge operator is similar to LoG edge operator which smooths the image first, then find the second derivative of the zero-crossing points. But the difference is that the LoG operator selects a two-dimensional Gaussian low-pass function to smooth image, while Shen Jun edge detection operator uses symmetric exponential function to smooth image. Shen Jun edge detection operator is proved to be the best symmetrical exponential filter for step edge with noise according to the maximum SNR criterion. Shen Jun edge operator only need one parameter, the details of the image will have a great change when we adjust this parameter. So if the appropriate parameter is selected, the rich useful details can be got. The function of symmetrical exponential filter as follows:

$$h(x, y) = C_1^2 + C_2^{|x|+|y|} \tag{4.2.1}$$

$$\text{where } C_1 = \frac{a_0}{2 - a_0}, C_2 = 1 - a_0, a_0 \in (0, 1)$$

a_0 is the only parameter. When a_0 is closer to 1, the $h(x, y)$ becomes narrower, at the same time the filter will be less resistant, but the edge position will be more accurate.

Shen Jun proved that Using (4.2.1) function to smooth the image $f_0(x, y)$ which can be implemented through recursive smoothing the image in positive and negative direction according to the rows and columns. The detail smoothing function as follows:

Firstly, smooth the image $f_0(x, y)$ according to rows in positive direction ($y = 1, 2, \dots, n$).

$$f_1(x, y) = f_1(x, y - 1) + a_0 \times [f_0(x, y) - f_1(x, y - 1)], \quad y = 1, 2, \dots, n \quad (4.2.2)$$

Secondly, smooth the image $f_1(x, y)$ according to rows in negative direction ($y = n, n - 1, \dots, 1$).

$$f_2(x, y) = f_1(x, y + 1) + a_0 \times [f_1(x, y) - f_1(x, y + 1)], \quad y = n, n - 1, \dots, 1 \quad (4.2.3)$$

$f_2(x, y)$ is the smoothing output in positive and negative directions according to the rows, then smooth image $f_2(x, y)$ according to the columns in positive direction ($x = 1, 2, \dots, n$).

$$f_3(x, y) = f_3(x - 1, y) + a_0 \times [f_2(x, y) - f_3(x - 1, y)], \quad x = 1, 2, \dots, n \quad (4.2.4)$$

Finally, smooth image according to the columns in negative direction ($x = n, n - 1, \dots, 1$).

$$f_4(x, y) = f_4(x + 1, y) + a_0 \times [f_3(x, y) - f_4(x + 1, y)], \quad x = n, n - 1, \dots, 1 \quad (4.2.5)$$

$f_4(x, y)$ is the smoothing output in positive and negative directions according to the columns, using $f_4(x, y)$ subtract $f_0(x, y)$ to get the difference image. The final output of the filter is the threshold image of the difference image, as shown in Fig. 6(a).

3.3 The Constraint Between Prewitt and Shen Jun Edge Detection

After get the edge points from Shen jun edge detection, we can restrict the edge points based on edge points from Prewitt edge detection in order to eliminate the inaccurate points and noise points.

Firstly, in a starting row (near the top) of the image, a search is conducted to find edge pixel positions x_1 and x_2 . Figure 4 are two illustrations of possible edge position. According to the possible width of the lane markings, the width between x_1 and x_2 is set bigger than minRL (which is set 2 pixel width) smaller than maxRL (which is set 35 pixel width). a, b are the pixel values bigger than 0.

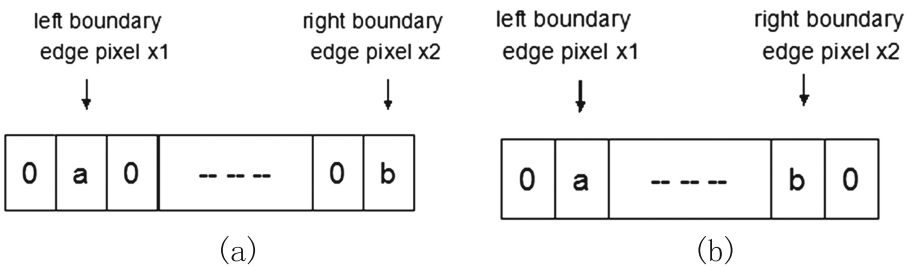


Fig. 4. x_1 and x_2 in (a, b) are illustrations of the edge points

The constraint algorithm is composed of three major steps: gratitude and gray value restriction, elimination of holes caused by Shen Jun edge detection and removing of wider continuous horizontal lines than lane markings. The flowchart is shown in Fig. 5.

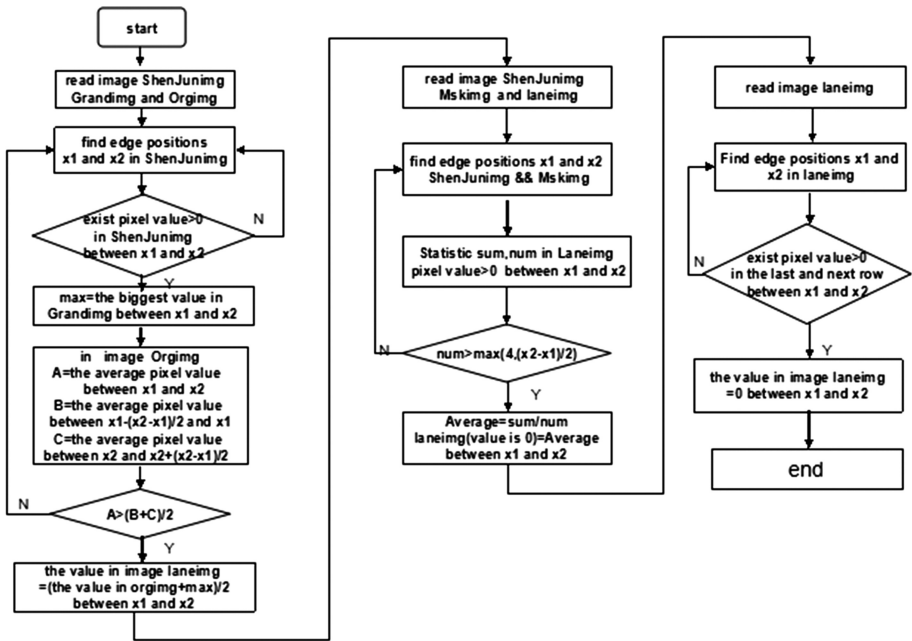


Fig. 5. Flowchart of eliminate incorrect lane markings

At first, when the edge points are found in ShenJunimg, a comparison is carried out in the same position of image Orgimg. If the average gray value in the both sides of the edge positions is bigger than the average gray value in middle of the edge positions, we can make the value in image Laneimg equal to the pixel value in Orgimg plus the maximum of gratitude in image Grandimg, which is shown in Fig. 6(b).

Then, after Shen Jun edge detection, there are some holes whose pixel-values are zero in image ShenJunimg. In order to fill the holes that belong to lane markings, the edge points that not only belong to image ShenJunimg but also belong to image Mskimg are found. When the distance between the edge points satisfy the threshold, the value in image Laneimg which are zero can be replaced by the average value in image Laneimg between edge points, the result is shown in Fig. 6(c).

Finally, sometimes there are some wider continuous horizontal lines than the actual width of the lane marking. We eliminate these continuous horizontal lines through making their pixel-values to be zero value, the image Laneimg is shown in Fig. 6(d).

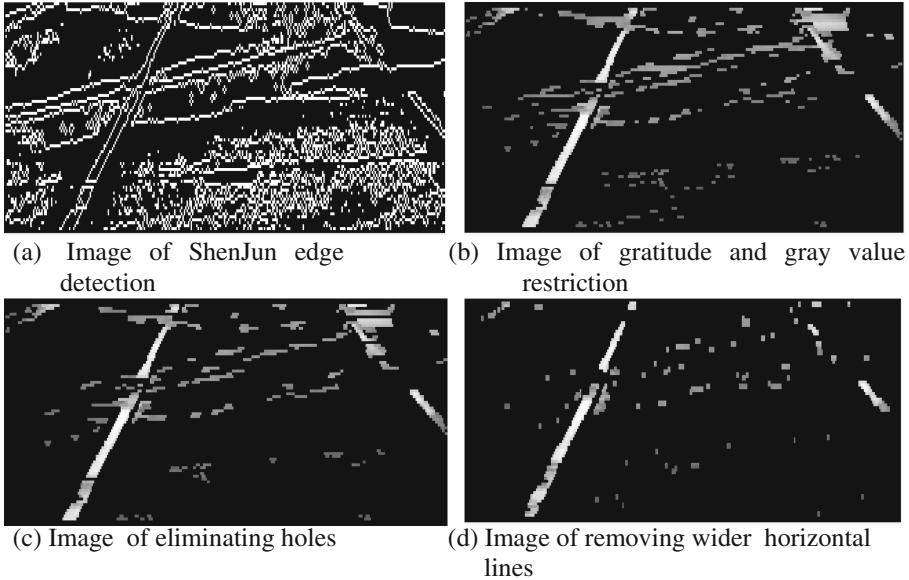


Fig. 6. The result of constraint between Prewitt and Shen Jun operator

4 Fast Hough Transform and Posterior Algorithm

Hierarchical Hough transform can reduce the computing time and improve the accuracy of detection. The procedure as follows: firstly, do three times down-sample processes to get a group of pyramid-resolution images. After that using the Hough transform at the lowest-resolution image to get the rough lean angle. Through Refining the lean angel at a higher-resolution image to get more accurate lane lines. At last, the most accurate lane lines can be got in the original image. As can be seen from the above process, the overall calculation is not big because of the pyramid-resolution Hough transform, but we can get higher precision. Suitable down-sample process also helps to improve the ability to resist noise.

After Hough transform, we can find several candidate lines at the left and right half of the image. The posterior algorithm removing the incorrect lane lines is carried out according to the following two steps. Firstly, each of the candidate lines is sampled at the equal interval sampling rows. At each sampled points, the average pixel value the middle rectangular window is compared with the each left and right area around the middle window. If the difference between them bigger than the threshold, the sample window is qualified. Only when the rate of the qualified sample window is large enough, the lane line will be reserved. Secondly, After two fittest lines are found in ROI, as shown in Fig. 7(a). Posterior algorithm of road marking lines is carried out to refine the lane angel. Firstly, each of the lines is sampled at the equal interval sampling rows, as shown in Fig. 7(b). The corresponding sampled points are then found, the intensity values which have a transition from dark to light to dark near the sampled point along the row would produce a rectangular pulse. An adjustable template [12] is

used to find the middle of the lane markings near the sampled points, as shown in Fig. 7 (c). Best lane markings estimation using the new middle points is shown in blue line in Fig. 7(d).

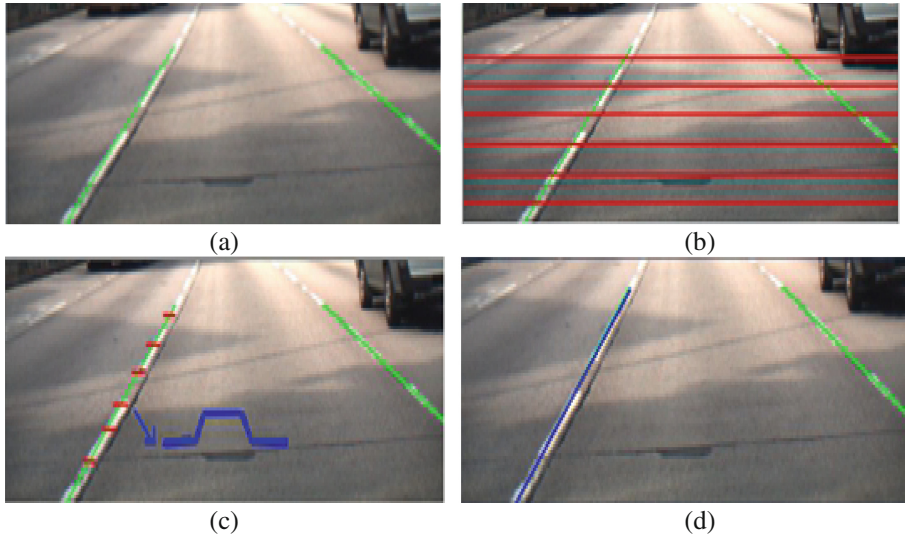


Fig. 7. The images in posterior algorithm of road marking lines

After the lane markings are detected and the vehicle’s lateral position are calculated, the departure warning system must interact with the driver to prevent road departure. A measure of roadway departure based on the vehicle’s lateral displacement and lateral velocity can work well. The vehicle’s lateral displacement is usually small because of the corrective driving, as shown in Fig. 8. To computer lateral velocity, we uses the lateral position of recent 1/2 s equal to fifteen frames. The function is as follows:

$$TTC = \text{distance} / \text{averagespeed} \tag{4.1}$$

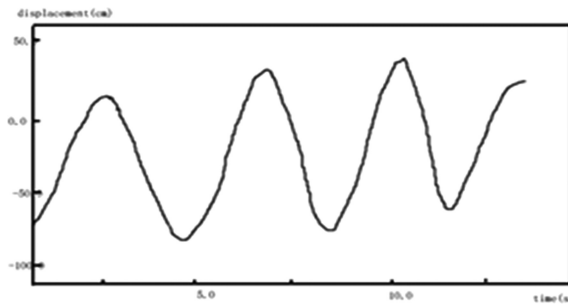


Fig. 8. The illustration of vehicle’s displacement from the road center



Fig. 9. The result of the proposed algorithm

$$\text{Averagespeed} = (\text{Speed } 1 + \text{speed } 2 + \dots + \text{speed } m) / m \quad (4.2)$$

$$\text{Speed}m = (\text{leftPoint}[m + 8] - \text{leftPoint}[m]) / dTm(m + 8) \quad (4.3)$$

Where $\text{leftPoint}[m]$ is the vehicle's displacement from the road center in the frame m , $dTab$ is the time difference between frame a and b .

5 Experimental Result

Our test vehicle equipped with a camera installed under the rear-view mirror is used to recorded videos of the roadways. Videos are recorded resolution at 30 frames/s. The algorithm has been implement in visual studio 2010. For images the resolution is 640×480 , ROI is 640×240 resolution, shrinking the ROI to 320×120 resolution. Although images down-sampling will cause edge pixels loss, this does not affect the accuracy of the detection algorithm very much, as edge pixels can be restored by using the constraint between Shen Jun edge detection and Prewitt operator edge detection. The algorithm can achieve an average speed of 20 frames per second on an Intel Core2 2.53 GHZ processor.

We tested our algorithm for many different marked roads. It fast and successfully detected road lane even under conditions including different shadows. Figure 9 Shows the result of the proposed algorithm under various illumination conditions to confirm the robustness of the proposed algorithm. Where the green lines is the lane lines found by the Hough Transform, the blue lines represent the results of posterior algorithm, the green blocks at the upper of the images represent there no lane departure warnings,

The red characters at the below of the images such as TTC time 6.0303 represent the collision time.

6 Conclusion and Future Work

In this paper, a robust and real-time lane markings detection system is presented. The major part of the system is the lane points extraction module based on the constraint between Prewitt and Shen Jun operator. Compared with other model-based lane detection methods, the proposed algorithm simplifies the modeling process so that it satisfies real-time detection. At constraint step, Shen Jun edge detection images are checked to select edge points which belong to the images of Prewitt operator detection. Most noises can be removed at this step, only edge points which belong to lane markings are used to Hough transform. Additionally, in the proposed algorithm, there are no special requirements for background models, camera parameters, or any other road surface models. This makes the algorithm more adaptive to different road environment.

For future works, we will develop the fast tracking algorithm. Now this algorithm has a real-time and robust detection of road markings every frame. So a tracking algorithm will be need to reduce the detection time and enhance the detection robustness, so that it can be carried out in a embedded system.

Acknowledgment. This work was supported by the Joint Fund of Department of Science and Technology of Guizhou Province and guizhou university under Grant: LH [2014] 7635. We would like to thank our colleagues for helping to this paper and the anonymous reviewers and the associate editor for their valuable suggestions.

References

1. McCall, J.C., Trivedi, M.M.: Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Trans. Intell. Transp. Syst.* **7**, 20–37 (2006). doi:[10.1109/TITS.2006.869595](https://doi.org/10.1109/TITS.2006.869595)
2. Zhou, S., Jiang, Y., Xi, J., et al.: A novel lane detection based on geometrical model and Gabor filter. In: *IEEE Intelligent Vehicles Symposium* (2010)
3. Liu, W., Zhang, H., Duan, B., et al.: Vision-based real-time lane marking detection and tracking. In: *IEEE Transactions on Intelligent Transportation Systems*, pp. 49–54 (2008)
4. Satzoda, R.K., Trivedi, M.M.: Efficient lane and vehicle detection with integrated synergies (ELVIS). In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 708–713 (2014)
5. Satzoda, R.K., Trivedi, M.M.: Vision-based lane analysis: exploration of issues and approaches for embedded realization. In: Kisačanin, B., Gelautz, M. (eds.) *Advances in Embedded Computer Vision*, pp. 217–235. Springer International Publishing, Switzerland (2014)
6. Lin, Q., Han, Y., Han, H.: Real-time lane departure detection based on extended edge-linking algorithm. In: *Second International Conference Computer Research and Development*, pp. 725–730. IEEE (2010)
7. Ding, D., Lee, C., Lee, K.: An adaptive road ROI determination algorithm for lane detection. In: *TENCON 2013–2013 IEEE Region 10 Conference* (31194), pp. 1–4. IEEE (2013)
8. Li, W., Gong, X., Wang, Y., et al.: A lane marking detection and tracking algorithm based on sub-regions. In: *International Conference on Informative and Cybernetics for Computational Social Systems (ICSS)*, pp. 68–73. IEEE (2014)
9. Grisman, J.D., Thorpe, C.E.: SCARF, A color vision system that tracks roads and intersections. *IEEE Trans. Robot. Autom.* **9**, 49–58 (1993)
10. Wang, Y., Teoh, E.K., Shen, D.: Lane detection and tracking using B-Snake. *Image Vis. Comput.* **22**(4), 269–280 (2004)
11. Ruyi, J., Reinhard, K., Tobi, V., et al.: Lane detection and tracking using a new lane model and distance transform. *Mach. Vis. Appl.* **22**(4), 721–737 (2011)
12. Borkar, A., Hayes, M., Smith, M.T.: A novel lane detection system with efficient ground truth generation. *IEEE Trans. Intell. Transp. Syst.* **13**(1), 365–374 (2012)
13. Chen, M., Jochem, T., Pomerleau, D.: AURORA, A vision-based roadway departure warning system. In: *1995 IEEE/RSJ International Conference on Intelligent Robots and Systems 95, Human Robot Interaction and Cooperative Robots, Proceedings*, vol. 1, pp. 243–248. IEEE (1995)
14. Lu, W., Rodriguez, F., Sergio, A., et al.: Monocular multi-kernel based lane marking detection. In: *2014 IEEE 4th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 123–128. IEEE (2014)
15. Kluge, K., Lakshmanan, S.: A deformable-template approach to lane detection. In: *Proceedings of the Intelligent Vehicles 1995 Symposium*, pp. 54–59. IEEE (1995)

16. Álvarez, J.M., Lopez, A.: Novel index for objective evaluation of road detection algorithms. In: 11th International IEEE Conference on Intelligent Transportation Systems, ITSC 2008, pp. 815–820. IEEE (2008)
17. Kim, Z.: Realtime lane tracking of curved local road. In: Intelligent Transportation Systems Conference, ITSC 2006, pp. 1149–1155. IEEE (2006)