An Efficient Lane Line Detection Method Based on Computer Vision

HG Zhu
China Highway Engineering Consulting Corporation, Beijing, China

*Corresponding author: hug_zhu@163.com

Abstract. Current approaches of using neural networks in lane departure warning systems are expensive. And it is difficult for neural networks to process 2K and 4K images. In this paper, we use a series of image preprocessing techniques such as perspective transformation, threshold processing and mask operation to process high-resolution images and the optimized sliding window method to fit the lane lines. Compared with neural network method, we can not only reduce hardware cost, but also quickly process high-resolution images. In addition, compared with the traditional lane line detection algorithm, we extract the region of interest through perspective transformation, which not only greatly reduces computation, but also converts images into an aerial view for subsequent processing. Especially, we carry out threshold operation and mask operation after perspective transformation, which greatly improves the performance of our algorithm in a strongly interfering environment. As can be seen from the experimental results, our method has good detection effect and can be applied to various road sections in different environments.

Keywords: Lane line detection, high resolution, real time, perspective transformation.

1. Introduction
With the rapid economic development, vehicles are increasing year by year and there are more and more traffic accidents. Statistics show that many traffic accidents are caused by drivers not having enough time to deal with complicated road conditions. The lane line deviation detection system [1] can give early warning when the vehicle deviates from the driving lane line and effectively reduce the accident rate. In this article, our contributions are as follows: (i) Use perspective transformation [2] to extract the ROI region, which greatly reduce computational requirements in the post-processing; (ii) Design a new kind of mask to eliminate a lot of interference, which improves the environmental adaptability of the algorithm; (iii) We have optimized the sliding window algorithm [3] to greatly improve the detection speed; (iv) Design a kind of quick and easy lane line migration algorithm. Through the above operations, our lane line deviation warning system can achieve excellent detection effect in different environments.

2. Lane line detection
Lane line detection is the most fundamental and critical step in the lane line deviation warning system. Accurate and rapid detection of lane lines is essential for subsequent operations. Although current
A popular neural network algorithm is in a leading position in accuracy such as LaneNet [4], it still has obvious deficiencies in hardware cost processing speed and high-resolution image processing [5]. In addition, different from other traffic signs, lane lines have fixed positions and single types, and convey little information. Therefore, lane lines can be detected quickly and accurately by some traditional methods.

2.1. Image preprocessing
The collected images will be subject to a lot of interference during vehicle movement including trees on both sides of the road, houses and other vehicles in front of the road. The presence of these areas will not only interfere with lane detection, but also makes subsequent processing even more difficult. Therefore, we used some methods to preprocess images. The main steps are as follows.

2.1.1. Using perspective transformation to extract ROI region. We used the idea of extracting region of interest [6] during image preprocessing to reduce subsequent processing. We chose perspective transformation instead of the traditional mask operation to determine ROI. [7] The essence of perspective transformation is to project an image onto a new visual plane, also known as mapping transformation [8]. The general transformation formula is:

\[
\begin{bmatrix}
x', y', w' \\
u, v, w
\end{bmatrix} =
\begin{bmatrix}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{bmatrix}
\begin{bmatrix}
x, y, w
\end{bmatrix}
\]

(1)

where \( u \) and \( v \) represent the original image, and the coordinates \( x \) and \( y \) of the transformed image are obtained corresponding to the left side of the equation. The transformation matrix can be divided into four parts: represents linear transformation, \([a_{31} a_{32}]\) used for translation, \([a_{13} a_{23}]^T\) produces perspective transformation. The previous transformation formula can be obtained by rewriting:

\[
x = \frac{x'}{w'} = \frac{a_{13}u + a_{23}v + a_{33}}{a_{33}u + a_{32}v + a_{31}}
\]

(2)

\[
y = \frac{y'}{w'} = \frac{a_{13}u + a_{23}v + a_{33}}{a_{33}u + a_{32}v + a_{31}}
\]

(3)

We can compute the transformation formula by using the corresponding points before and after the transformation. At the same time, the specific transformation formula can also produce a new transformed image. Here, we only need to select the four vertex coordinates of the region to be processed from the original drawing and set the coordinates of these four points after perspective transformation. Thus, perspective transformation can be easily implemented. The rendering of perspective transformation is as follows:
As shown in the figure above, we use perspective transformation to re-project the selected quadrilateral area to a new plane in an aerial view. Through perspective transformation, we remove most of the interference in the image and project the lane lines to a relatively parallel position to make subsequent processing less difficult.

2.1.2. Image graying and threshold processing. The image is obtained after perspective transformation. First, the image is converted to a grayscale image which has the following two advantages: (i) The three-channel image is converted into a single-channel image; (ii) The grayscale image can be converted into a binary image directly by setting a threshold or using adaptive threshold [9], and part of the interference can be removed during conversion.

After obtaining the grayscale image, we conducted threshold processing on the grayscale image. The processing results indicate that most of the interference has been removed after threshold processing.
2.1.3. **Mask operation removes large areas of interference.** In this article, we use masks to remove interference areas in contrast to the usual masks [10]. First, a white image of the same size as the original image is generated. Second, according to the approximate position of the lane lines, mask area is demarcated in advance between the lane lines and filled with black color to generate a mask image. Finally, the mask image is superimposed with the image after threshold processing to obtain the result image. When there is a large amount of interference in the lane line, mask operation can effectively remove the interference and ensure the accuracy of lane line detection.

2.1.4. **Improved sliding window polynomial fitting lane line.** Traditional lane line search algorithm uses Hof linear transformation [11], which requires high quality of input image when detecting lane line. But it is difficult to adapt to real scenes with a lot of interference due to its poor recognition effect and low recognition rate. Therefore, the sliding window algorithm was used to identify and track the lane line [12]. In addition, considering that a section of lane line will appear repeatedly in consecutive multiple frames, it is wasteful for the algorithm to process each frame. In order to solve this problem, we improved the algorithm according to the actual situation.

According to the experimental test, the length of the lane line covered by each frame is about 5 meters, which means the sliding window algorithm will repeatedly detect lane lines within a distance of five meters. In order to avoid repeated detection and improve the process speed, we set a threshold, that is, only when the driving distance of the vehicle is greater than this threshold, we will carry out detection again. After many experiments, we set the threshold at 3.5m, at which the detection speed
has been significantly improved without affecting its accuracy. Assuming that the vehicle speed is 70km/h and the camera collects 30 frames per second, the improved sliding window algorithm detects the lane lines approximately every 5 frames, which means the calculation cost in this section is reduced by about 80%.

2.2. Lane line deviation warning

After obtaining the fitting image of lane lines, we took the middle line of the image as the Y-axis and the bottom of the image as the X-axis to establish the coordinate system of the road, as shown in Figure 6. In order to calculate the degree of deviation in the left and right directions, we took the distance between the left lane line and the Y-axis as $d_1$ and that between the right lane line and the Y-axis as $d_2$. Here, $d_1$ and $d_2$ represent the number of pixels. If the actual width of the lane line is known to be $d_3$, we can calculate the offset distance $d_3$. The formula is as follows:

$$d_3 = \frac{d(d_1 - d_2)}{2 \times (d_1 + d_2)}$$

If $d_3 > 0$, then the vehicle will deviate to the right from the middle lane for $d_3$ meters, otherwise it will deviate to the left from the middle lane for $d_3$ meters. According to the actual condition of the road, we set the threshold as 0.25m, that is, once $d_3 > 0.25$, the system will warn the driver of the risk so that they can timely leave the lane line. In addition, we also calculated the radius of curvature according to the fitting lane lines.
2.3. The inverse perspective transform \[13\] fits the results to the original diagram

The fitting result on the perspective transformation chart was obtained after the above steps. In order to get the final result, we only need to transform the graph of perspective transformation into the original image. In perspective transformation, we do get a transformation matrix, and we only need to use the inverse transformation of the matrix to fit the final result into the original image.

3. Experiment Result

After fixing the camera position and setting the parameters, we tested the detection effect on newly built roads, urban roads with large traffic flow, national highway with good road conditions and roads with fuzzy lane lines and a lot of interference.

The detection effects in the case of massive disturbances in urban roads are shown in the following figures.

The detection effects under normal road conditions with the interference of speed bumps are shown in the following figures.
Figure 8. Detection effect for normal roads.

The detection effect of newly built roads with good conditions is shown in the following figures.

Figure 9. Detection effect for newly built roads.

The detection effect for roads where lane lines are fuzzy and there is a lot of interference is shown in the following figures.

Figure 10. Detection effect for Poor road conditions.

4. Conclusions
According to theoretical and experimental analysis, our method can not only achieve high detection accuracy in various harsh environments, but also achieve fast, accurate and real-time detection in processing high-resolution images, which cannot be achieved by many neural network methods. But our algorithm also has some defects. Firstly, the perspective transformation parameters need to be
adjusted according to the installation angle of the camera. Second, the calculation of lane line deviation distance is not accurate enough.

References
[1] A Kullack, I Ehrenpfordt, K Lemmer, F Eggert, "ReflektAS: lane departure prevention system based on behavioural control" IET Intelligent Transport Systems (2008).
[2] Huang, Yingping; Li, Yangwei; Hu, Xing, "Lane Detection Based on Inverse Perspective Transformation and Kalman Filter" KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS 12 643-661 (2018).
[3] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, Philip Torr,"BING: Binarized Normed Gradients for Objectness Estimation at 300fps" CVPR (2014).
[4] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, Luc Van Gool, "Towards End-to-End Lane Detection: an Instance Segmentation Approach" CVPR (2018).
[5] Zou, Qin; Jiang, Hanwen; Dai, Qiyu; "Robust Lane Detection From Continuous Driving Scenes Using Deep Neural Networks" IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY 69 41-54 (2020).
[6] M Heikkilä, M Pietikäinen, C Schmid, "Description of interest regions with local binary patterns" Pattern recognition (2009).
[7] H Watanabe, T Ogita, "Transcoding by automatic roi extraction from jpeg2000 bitstream" Proceedings of Picture Coding Symposium (2003).
[8] Z Hong, B Cheng-Lin, "Extended mapping transformation method and its applications to nonlinear partial differential equation" Communications in Theoretical Physics (2005).
[9] G Jie, L Ning, "An improved adaptive threshold canny edge detection algorithm" International Conference on Computer Science and Electronics Engineering (2012).
[10] BL Toll, R Valentine, JC San Adrian, "Packed data operation mask concatenation processors, methods, systems and instructions" US Patent (2017).
[11] Cao, Xiaohua; Liu, Daofan; Ren, Xiaoyu, "Detection method for auto guide vehicle's walking deviation based on image thinning and Hough transform" MEASUREMENT & CONTROL 52, 252-261 (2019).
[12] He, Jianye; Sun, Sha; Zhang, Debing, "Lane Detection for Track-following Based on Histogram Statistics" IEEE INTERNATIONAL CONFERENCE ON ELECTRON DEVICES AND SOLID-STATE CIRCUITS (2019).
[13] Y Huang, Y Li, X Hu, W Ci, "Lane Detection Based on Inverse Perspective Transformation and Kalman Filter." KSII Transactions on Internet & Information (2018).