ISSN 2616-5767 Vol.6, Issue 8: 37-43, DOI: 10.25236/AJETS.2023.060806

Research on Lane Detection Method Based on Machine Vision

Zhaoxiang Wang[1,a,*]

[1]School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou, Jiangxi, China
[a]wzx111666@163.com
[*]Corresponding author

Abstract: *With the continuous increase in the number of vehicles, traffic safety issues have attracted widespread attention from all sectors of society. In this paper, a lane information extraction algorithm based on Gabor filters, which have good recognition characteristics for road texture information, is proposed for lane extraction. Using a well-established test set for detection experiments, the lane detection aspect of the algorithm was tested with a false detection rate of 1.9%, which is 0.9% higher than the existing RANSAC lane fitting algorithm. The algorithm used in this paper is feasible.*

Keywords: *Lane Detection, Gabor Filter, Data Experiment, Lane Extraction*

1. Introduction

Due to China's growing economy and improved living standards, there has been a gradual increase in the number of motor vehicles. ADAS is a driving assistance system that utilizes on-board sensors to collect and analyze environmental information, providing drivers with additional driving information and warnings to help avoid driving risks. Lane detection is a crucial component of the ADAS system, as an accurate and efficient system can provide pre-warning to drivers and assist in ensuring driving safety. Therefore, it has significant research value and wide application.

Lane detection is a crucial task in advanced driver assistance systems (ADAS) and autonomous vehicles. In recent years, many lane detection algorithms based on various techniques have been proposed. Shieh and Wang [1] proposed lane detection and tracking algorithm using image and LiDAR fusion. Chiu and Liang [2] developed a lightweight convolutional neural network-based algorithm for lane detection. Wu et al. [3] presented a deep learning-based approach for lane detection and tracking using road markings. Zhang et al. [4] proposed a fuzzy clustering and road surface models-based lane marking detection algorithm. Tan et al. [5] used deep learning and trajectory prediction for vehicle lane detection. Chang and Wang [6] developed an adaptive lane detection algorithm based on multi-scale fusion. Chen et al. [7] presented a Bayesian convolutional neural network-based lane marking extraction algorithm. Zhang et al. [8] proposed a deep learning-based lane detection approach for intelligent vehicles. Huang et al. [9] used semantic segmentation for accurate and efficient lane detection. Jiang et al. [10] developed a vision-based robust lane detection algorithm in adverse weather conditions using deep learning. Wei et al. [11] proposed a fast lane detection algorithm using a cascaded fusion network. Yu et al. [12] presented a novel deep learning-based method for lane marking extraction. Kim et al. [13] developed a vision-based lane detection algorithm using a dual-stage convolutional neural network. Song et al. [14] proposed a lane detection algorithm based on convolutional neural network for night road environment. Ren et al. [15] developed an efficient lane detection algorithm with improved edge information and spatio-temporal clustering. These studies have contributed to the development of various lane detection algorithms for ADAS and autonomous vehicles using different techniques.

This paper proposes a lane information extraction algorithm based on Gabor filters, which can effectively recognize road texture information.

2. Road Vanishing Point Search Based on Road Surface Texture Orientation Features

2.1. Texture Feature Extraction Based on Gabor Filters

The Gabor transform is defined as follows: Assuming that a signal f(t) exists in the time domain and f(t)∈L2(R), the following transform exists.

$$G(a,b,\omega)=\int_{-\infty}^{\infty} f(t)g_a(t-b)e^{-j\omega t}dt \tag{1}$$

Where ga(t) is a Gaussian window function defined as follows:

$$g_a(t)=\frac{1}{2\sqrt{\pi a}}\exp\left(-\frac{t^2}{4a}\right) \tag{2}$$

Here a>0, b>0.

The above is the definition of one-dimensional Gabor transform. In order to facilitate the processing of two-dimensional image signals, it is necessary to extend the Gabor filter from one dimension to two dimensions. In 1985, Daugman proposed the two-dimensional form of Gabor transform, the mathematical expression of the two-dimensional Gabor function is as follows: "Two-Dimensional Cortical Visual Filter".

$$\begin{cases} g(x,y,\theta_k,\lambda,\sigma_x,\sigma_y)=\dfrac{1}{2\pi\sigma_x\sigma_y}\exp\{-\pi[(\dfrac{x'}{\sigma_x})^2+(\dfrac{y'}{\sigma_y})^2]\}\cdot\exp(\dfrac{2\pi jx'}{\lambda}) \\ x'=x\cos\theta_k+y\sin\theta_k \\ y'=y\cos\theta_k-x\sin\theta_k \end{cases} \tag{3}$$

In the equation, θk represents the orientation of the Gabor filter, and λ represents the wavelength of the sine wave. By changing θk, Gabor filters with multiple directions can be obtained. The definition of θk is as follows:

$$\theta_k=\frac{k\pi}{n},(k=0,1,\ldots,n-1) \tag{4}$$

Where n represents the number of direction categories of the filter.

In the two-dimensional Gabor filter, σx and σy represent the standard deviation of the filter in the x and y coordinates axis directions, respectively, which determines the coverage range of the filter in the image.

2.2. Road Vanishing Point Detection Based on Gabor Filters

The detection of the vanishing point is divided into two steps: first, the estimation of the vanishing point, and then the voting and screening of the vanishing point.

(1) Vanishing Point Estimation

In terms of vanishing point estimation, this paper uses Gabor filters to estimate and detect the vanishing point. During normal driving, the lane angle will always be in the range of 20° to 165° with the vertical direction of the vehicle's travel direction as the x-axis and the vehicle's travel direction as the y-axis. Therefore, this paper divides the range [20°, 165°] into 30 parts with a unit value of 5°.

Since Gabor wavelets have both real and imaginary parts, convolving the Gabor convolution kernel with the point a(x,y) on the image also yields two response results. The convolution operation of the Gabor kernel with the pixel point a(x,y) on the image is defined as g(a). Therefore, it is necessary to square the convolution response of each pixel point in each direction and define the corresponding

square value as follows:

$$I_k(a) = \mathrm{Re}\,(g(a))^2 + \mathrm{Im}\,(g(a))^2 \tag{5}$$

The average response of all scales in that direction is used as the criterion for determining the main direction of the image texture.

$$R_k(a) = \frac{1}{30}\sum I_k(a) \tag{6}$$

After using the above method to obtain Rk(a) for all directions, the main texture direction of the pixel point is defined as the direction θ(a) with the maximum convolution response value.

$$\theta(a) = \theta_k(\max R_k(a)) \tag{7}$$

Based on the definition formula of the two-dimensional Gabor filter, 30 different Gabor filters can be defined for different directions. By convolving the pixel point a with different directional filters, the convolution response rk(a) of the pixel point a in different directions can be obtained, which can be sorted according to size. The maximum response value r1(a) is defined, with r1(a)>r2(a)>…>r30(a), from which the main texture direction of the pixel point can be determined. In order to ensure the reliability of the texture main direction screened in this way, a confidence function needs to be defined to judge the main texture direction. In the experiment, it is determined that when the maximum convolution response r1(a) is significantly different from the response of other texture directions, the screened texture direction is considered reliable, otherwise the pixel point is considered as an invalid point and screened out. The confidence function is defined as follows:

$$\mathrm{Conf}\,(a) = 1 - \frac{\mathrm{Max}\,(r_2(a),\ldots,r_{30}(a))}{r_1(a)} \tag{8}$$

After calculation, a set containing the confidence of all pixel points is obtained. Low-confidence texture directions are screened out in the set by setting a threshold. Based on experimental results analysis, a threshold of Conf(a)=0.8 is selected here. By using this method to traverse the image, a candidate voting point map containing different texture direction candidates can be obtained. Since the road is integral, the texture direction of the vanishing point should tend to be consistent. Therefore, the voting points in the image need to be filtered. In this paper, the number of pixel points with consistent main texture directions in all directions is counted, and the texture direction with more voting points is selected for subsequent voting operations.

(2) Vanishing Point Voting and Screening

Assume that there is a voting point P with a texture direction vector P', then all pixel points V above point P can be used as voting points for the vanishing point.

During the screening of voting points, all voting points in the semi-circular area below the candidate point are considered. However, points with an angle θ greater than the threshold are obviously not selectable, which slows down the algorithm. At the same time, candidate points for the road will not appear on the two sides and lower edges of the image, and are less likely to appear on the upper edge, so part of the image area is discarded before the screening of voting points, and the image edges are not processed.

Considering the directional characteristics of road surface texture, the vanishing point of the road should be consistent with the texture direction of most pixel points on the road surface. Therefore, when the candidate point V is collinear with the voting point P, it should receive more votes. Conversely, when the voting point and the candidate point are not collinear, the votes received by the candidate point should be suppressed. Therefore, this paper adds a weighting coefficient cosθ to the local soft voting method, where θ is the angle between PV and P'.

$$\mathrm{Vote}\,(P,V) = \begin{cases} \dfrac{\cos\theta}{1+D^2}\exp\left(-\dfrac{d}{\theta}\right) & V \in R_V \\ 0 & \text{else} \end{cases} \tag{9}$$

3. Lane Fitting Method Using Catmull-Rom Splines

3.1. Inverse Perspective Transformation of the Image

Assuming the world coordinate system is (XW,YW,ZW), and the pixel coordinate system is (u,v), the entire imaging process of the camera is equivalent to mapping a point PW=[XPW,YPW,ZPW]T in the world coordinate system to a point PC=[XPC,YPC,ZPC]T in the camera coordinate system, and then transforming it to a point P=[up,vp,1]T in the pixel coordinate system. To achieve this transformation, the world coordinate system can be translated by a distance of T first, followed by a rotation matrix R to obtain the expression of point P in the camera coordinate system. The specific transformation process can be represented by the following equation.

$$\begin{bmatrix} X_{PC} \\ Y_{PC} \\ Z_{PC} \end{bmatrix} = A \cdot R \cdot \begin{bmatrix} X_{PW} \\ Y_{PW} \\ Z_{PW} \end{bmatrix} + T \tag{10}$$

Combining the above equation into a homogeneous matrix yields:

$$[X_{PW}, Y_{PW}, Z_{PW}, 1]^T = A \cdot \begin{bmatrix} R & T \\ \vec{O} & 1 \end{bmatrix} \cdot [X_{PC}, Y_{PC}, Z_{PC}, 1]^T \tag{11}$$

Where A is the camera's intrinsic matrix.

Since the internal parameters of each lens are different, the degree of distortion produced by the captured photos is also different. Therefore, it is necessary to use camera calibration methods to obtain the camera's intrinsic parameters. After calibration, the final intrinsic matrix is determined as follows:

$$A = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

In the equation, fu represents the focal length of the camera in the x-axis direction, fv represents the focal length of the camera in the y-axis direction, and u0 and v0 represent the actual coordinates of the camera's optical center.

Then, the camera coordinate system is transformed into the pixel coordinate system, and the following formula is obtained:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z_{PC}} \begin{bmatrix} \frac{f_u}{dx} & 0 & u_0 \\ 0 & \frac{f_v}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R & T \\ \vec{O} & 1 \end{bmatrix} \cdot \begin{bmatrix} X_{PW} \\ Y_{PW} \\ Z_{PW} \end{bmatrix} \tag{13}$$

Set

$$M = \begin{bmatrix} \frac{f_u}{dx} & 0 & u_0 \\ 0 & \frac{f_v}{dy} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R & T \\ \vec{O} & 1 \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix} \tag{14}$$

Then we will get that

$$\begin{cases} u = \dfrac{m_{00}X_{PW} + m_{01}Y_{PW} + m_{02}Z_{PW}}{m_{20}X_{PW} + m_{21}Y_{PW} + m_{22}Z_{PW}} \\ v = \dfrac{m_{10}X_{PW} + m_{11}Y_{PW} + m_{12}Z_{PW}}{m_{20}X_{PW} + m_{21}Y_{PW} + m_{22}Z_{PW}} \end{cases} \tag{15}$$

Given a line L in space with direction D=[dx,dy,dz]T, any point A=[xa,ya,za]T on the line can be represented as:

$$A'(\mu) = A + \mu D \tag{16}$$

As μ approaches infinity, the point A'(μ) tends to the infinite point. According to the perspective effect, parallel road boundary lines will intersect at a point at infinity in the image. This point projects to the pixel coordinate system, which is the vanishing point of the road mentioned earlier. Therefore, the coordinates of the vanishing point can be obtained as follows:

$$\begin{cases} u_{\infty} = \dfrac{m_{00}d_x + m_{01}d_y + m_{02}d_z}{m_{20}d_x + m_{21}d_y + m_{22}d_z} \\ v_{\infty} = \dfrac{m_{10}d_x + m_{11}d_y + m_{12}d_z}{m_{20}d_x + m_{21}d_y + m_{22}d_z} \end{cases} \tag{17}$$

Using this known information, the correspondence between any point on the image and its corresponding point in the spatial coordinate system can be inferred. Figure 1 shows a schematic diagram of the road area and vanishing point found in the previous search.

Figure1: Lane Extraction and Vanishing Point Calculation

3.2. Curve Fitting Algorithm Based on Catmull-Rom Splines

Catmull-Rom spline curve is an interpolating spline model. The solving method of Catmull-Rom spline curve is as follows. Assuming that we need to use Catmull-Rom spline to obtain the curve between two points P1 and P2 on the plane.

$$P(t) = \alpha t^3 + \beta t^2 + \gamma t + \mu \tag{18}$$

Then, two additional control points P0 and P3 need to be selected on the plane, and the control points must satisfy:

$$\begin{cases} P(0) = P_1 \\ P(1) = P_2 \\ P'(0) = \delta(P_1 - P_0) \\ P'(1) = \delta(P_3 - P_2) \end{cases} \tag{19}$$

Here

$$P'(t) = 3\alpha t^2 + 2\beta t + \gamma \tag{20}$$

Taking δ=0.5 and solving yields:

$$P(t)=\begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}\begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \tag{21}$$

As seen, at least four control points are needed to fit a curve using the Catmull-Rom spline model. To test the fitting effect of the Catmull-Rom spline model on the curve, the fitting result of the Catmull-Rom spline for the given four control points P0, P1, P2, P3 is shown in Figure 2.

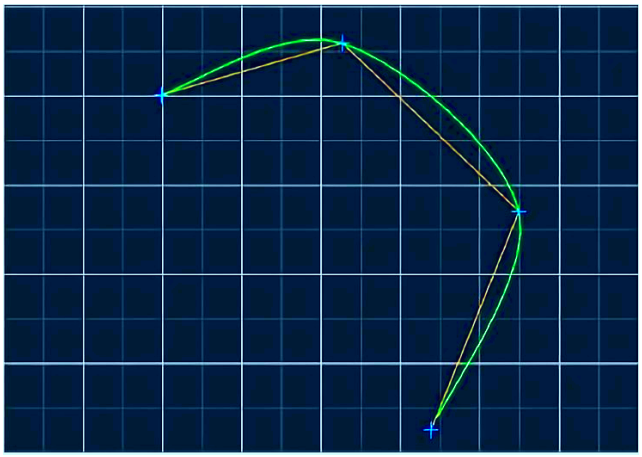

Figure 2: Fitting Result of Catmull-Rom Spline Curve

The detection results of the partial image using the Catmull-Rom lane fitting algorithm are shown in Figure 3. To ensure the continuity of the lane lines, in the process of fitting the dashed lane lines, this paper integrates the segmented dashed lane lines based on the texture direction characteristics of each lane line, using a criterion that the horizontal pixel difference is less than 20 and the texture direction angle is less than 5°. Segmented lane lines that do not exceed the threshold are clustered into one. At the same time, clustering analysis is also performed on closely spaced double yellow line contours in the image, and the two yellow lines are arranged into one. The center line of the cluster is then taken as the fitting line for curve fitting.

Figure 3: Lane fitting results

Using two different algorithms for comparative testing, the false positive rate of the RanSac algorithm was 3.8%, the false negative rate was 7.3%, and the accuracy rate was 88.9%. The false positive rate of the Catmull-Rom algorithm was 1.9%, the false negative rate was 9.5%, and the accuracy rate was 88.8%. The advantage of the Catmull-Rom algorithm over the RanSac algorithm is that its false positive rate is lower, meaning that the algorithm more accurately classifies negative samples as negative. Although the false negative rate is slightly higher, the difference is not significant, and both algorithms have almost the same accuracy rate.

References

[1] Shieh, J.J., Wang, C.H. Lane detection and tracking based on fusion of image and LiDAR data.

Journal of Intelligent & Robotic Systems (2020).
[2] Chiu, W.C., Liang, T.C. An Effective Lane Detection Algorithm Based on a Lightweight Convolutional Neural Network. IEEE Access (2021).
[3] Wu, S., Li, C., Zhang, Y., Liu, Y., Yao, Q. Lane detection and tracking via deep learning with road markings. Journal of Real-Time Image Processing (2020).
[4] Zhang, Y., Feng, C., Cui, Z., Zhu, J., Xiong, Z. A novel approach to lane marking detection based on fuzzy clustering and road surface models. International Journal of Advanced Robotic Systems (2020).
[5] Tan, W., Guo, L., Yu, J., Liu, B., Wei, J., Wang, Y. Vehicle Lane Detection Method Based on Deep Learning and Trajectory Prediction. IEEE Transactions on Intelligent Transportation Systems (2021).
[6] Chang, Y.J., Wang, Y.H. Adaptive Lane Detection in Challenging Scenarios Based on Multi-scale Fusion. IEEE Transactions on Intelligent Transportation Systems (2021).
[7] Chen, C., Zhang, B., Zhang, C., Zhang, Z. Lane Marking Extraction Based on Bayesian Convolutional Neural Network. IEEE Transactions on Intelligent Transportation Systems (2021).
[8] Zhang, S., Wang, H., Huang, M., Zhang, J., Liu, W. A novel deep learning-based lane detection approach for intelligent vehicles. Applied Sciences (2021).
[9] Huang, L., Liu, W., Liu, Y., Zheng, N., Zhang, X. Accurate and Efficient Lane Detection via Semantic Segmentation. IEEE Transactions on Intelligent Transportation Systems (2020).
[10] Jiang, C., Yang, J., Wang, R., Huang, X., Zhou, J. Vision-based robust lane detection in adverse weather conditions using deep learning. Transportation Research Part C: Emerging Technologies (2021).
[11] Wei, Q., Chen, X., He, H., Fu, J., Feng, J. Fast lane detection using cascaded fusion network. IEEE Transactions on Intelligent Transportation Systems (2020).
[12] Yu, H., Liu, J., Wang, Y., Zhang, H., Yu, Q. A novel deep learning-based method for lane marking extraction. IET Intelligent Transport Systems (2021).
[13] Kim, Y., Hong, S., Kwak, S., Park, W. Vision-based lane detection using a dual-stage convolutional neural network. IET Intelligent Transport Systems (2020).
[14] Song, B., Liu, S., Zou, D., Zhang, J., Gao, Y. Lane detection algorithm based on convolutional neural network for night road environment. Journal of Ambient Intelligence and Humanized Computing (2021).
[15] Ren, Y., Huang, Q., Luan, Z., Chen, C., Xu, Y. Efficient Lane Detection with Improved Edge Information and Spatio-Temporal Clustering. IEEE Transactions on Intelligent Transportation Systems (2020).