Lane Detection Using Heuristic Search Methods Based on Color Clustering

Chao Ma, Ling Mao, Yuefei Zhang, Mei Xie
School of Electronic Engineering/UESTC/Chengdu, Sichuan, China
ziyanmach@163.com

***Abstract*—Concerning the problem of lane detection in the Lane Departure Warning (LDW) system, this paper uses Heuristic Search Methods to detect the lane boundary points based on the *CIELab* color features clustering. Color space providing us more precious information than gray scale, and Heuristic Search Methods narrowing the search region, this algorithm is robust and adaptive to the actual road conditions. According to the geometry structure of super highway, quadratic curve is adopted to match the lane. At last, use Simulink, the Signal Processing Blockset, and the Video and Image Processing Blockset developing lane detection and tracking model.**

I. INTRODUCTION

LDW is one part of Safety Driving Assist (SDA) on intelligent vehicle, which is on behalf of the development direction of smart vehicle. The prerequisite to achieve LDW function is identifying the road environment. Machine vision technology continues to evolve, with its capability of precise recognition of road traffic environment improving and relatively cost lower, which has shown a great prospect. Therefore, vision-based lane detection is becoming the focus problem which attracts a lot of researchers' attention both at home and abroad. Currently, the general methods use a model to characterize the roadside in the image. References [1-2] adopt stereovision to built 2D and 3D visual model, and [3] uses Hough-Transform to get straight line model's parameters, as to the curve model, polynomial curve [4], B-Snake spline curve [5] and Catmull-Rom spline curve [6] permit to model the roadsides. For the purpose of recognizing lane marking features, template matching [7], threshold segmentation, edge detection and other traditional digital image processing methods are used. Moreover, along with the development of machine vision technology, neural network algorithm (see [8]) based on large sample self-learning and training is used for road recognition, but such algorithm's requirement for the sample's quantity and quality is extremely rigorous. Therefore, such algorithm is doomed not quite easy to meet the road diversity requirement.

In the practice of driving, due to vehicle vibration, weather changing, day or night driving and other reasons, the acquired image's color and brightness vary widely. In seeking precise detection, these algorithms above-mentioned mostly abandon the color information, and directly process the grayscale image. However, valuable color information provides more rich information, which helps to improve detection accuracy. In this paper, a method called *K-means* color clustering based on *CIELab* color space, the closest color space to human vision, is proposed to segment the complete region of lane marking in the image. And then, we use Heuristic Search Methods to extract the marking boundary, and use quadratic curve fitting based on least square method to model the current lane's boundary.

The Video and Image Processing Blockset extends Simulink with a rich, customizable framework for the rapid design, simulation, implementation, and verification of video and image processing algorithms and systems. It includes basic primitives and advanced algorithms for designing embedded imaging systems in a wide range of applications in aerospace and defense, automotive, communications, consumer electronics, education, and medical electronics industries. Finally, we simulate our model by it. This approach expresses our design concept, and verifies this algorithm robust and adaptive to actual road conditions.

II. LANE MARKING REGION EXTRACTION

A. Saliency Calculation

CIELab color space, specified by the International Commission on Illumination (*Commission Internationale d'Eclairage*, hence its *CIE* initialism), describes all the colors visible to the human eye and serves as a device independent model to be used as a reference.

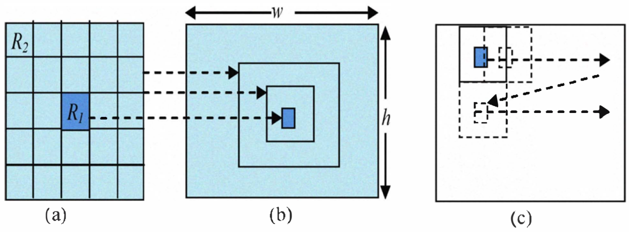


Figure 1. (a) Contrast detection filter showing inner square region R_1 and outer square region R_2. (b) The width of R_1 remains constant while that of R_2 ranges. (c) Filtering the image at one of the scales

R. Achanta [9] uses sub-region R_2, to find salient color region R_1, as shown in Fig. 1 (a), whose width w_{R_2} meets

$$\frac{w}{2} \leq w_{R_2} \leq \frac{w}{8} \tag{1}$$

where w is the whole image's width. At a given scale, the contrast based saliency value $C_{i,j}$ for a pixel at position (i, j) in the image is determined as the distance D between the average vectors of pixel features of the inner region R_1 and that of the outer region R_2 :

$$C_{i,j} = D\left[\left(\frac{1}{N_1}\sum_{p=1}^{N_1} v_p\right),\left(\frac{1}{N_2}\sum_{q=1}^{N_2} v_q\right)\right] \tag{2}$$

where N_1 and N_2 are the number of pixels in R_1 and R_2 respectively, and v is the vector of feature elements corresponding to a pixel. Considering distance D is Mahalanobis distance and in *CIELab* color space, we can calculate $C_{i,j}$:

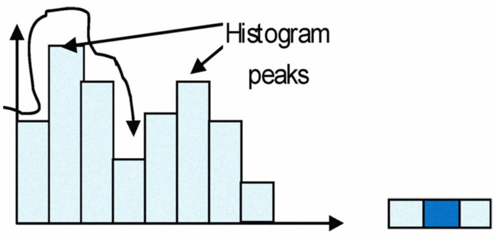
$$C_{i,j} = \left\| v_1 - v_2 \right\| \tag{3}$$

where $v_1 = [L_1, a_1, b_1]^T$, $v_2 = [L_2, a_2, b_2]^T$. In this paper, we present $w_{R_2} = w/3$, and use R_2 to filter the image at 3 scales in a raster scan fashion (see Fig. 1 (c)). The final saliency map is determined as a sum of saliency values $m_{i,j}$,

$$m_{i,j} = \sum_{3} C_{i,j} \tag{4}$$

B. Road image Segmentation

The image is over-segmented using a simple *K-means* algorithm. The K seeds for the *K-means* segmentation are automatically determined using the hill-climbing algorithm (see [10]) in the three-dimensional *CIELab* histogram of the image. The hill-climbing algorithm can be seen as a search window being run across the space of the *d*-dimensional histogram to find the largest bin within that window. Fig.2 explains the algorithm for a one-dimensional case. Since the *CIELab* feature space is three-dimensional, each bin in the color histogram has $3^d\text{-}1=26$ neighbors. The number of peaks obtained indicates the value of K, and the values of these bins form the initial seeds.

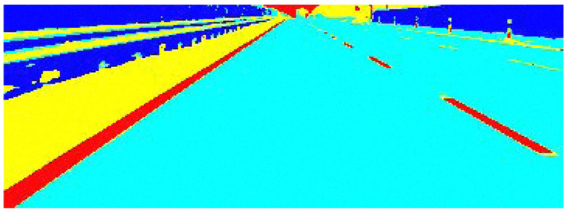


(a) (b)

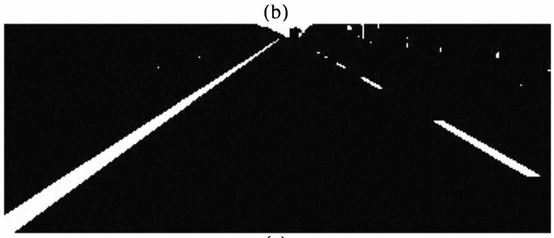
Figure 2. Finding peaks in a histogram using a search window like (b) for a one dimensional histogram

Since *K-means* algorithm clusters pixels in the *CIELab* feature space, an 8 neighbor connected-components algorithm is run to connect pixels of each cluster spatially. Hence, the segmented regions $r_i, i=1,2...K,$ are found (see Fig. 3 (b)).

(a)



(b)



(c)

Figure 3. (a) original image, (b) the color clustering region labeled by pseudo color, (c) the lane marking region

C. Lane marking region extraction

The standard colors most commonly used to mark lane are yellow and white, which are focused at R and G channels. In road image, values of R and G components contained in white color and yellow color are adjacent and higher than other colors. In accordance with equation (5), the brightness values I_i in region r_i are calculated respectively:

$$I_i = \frac{1}{2N_i}\sum_{k=1}^{N_i}\left(I_R + I_G\right) \tag{5}$$

where N_i is the number of pixels in R_i, and I_R and I_G are the R component value and G component value of pixel k (index of pixel position) in R_i. And the regions with maximum I_i are lane marking regions, as shown in Fig. 3 (c).

III. CURRENT LANE'S BOUNDARY RECOGNITION

After the marking region has been extracted by color clustering, this paper uses Heuristic Search Methods to search current lane's boundary. The heuristic search techniques earliest presented by Martelltit (see [11]) to extract the boundary points, the core thinking of which is to search for the next most likely boundary points based on context information, and the whole process mainly including: the starting point defined, boundary points chosen, as well as the termination point determined.

A. The Choice of starting point

To determine the starting point is very important, which is directly related to the search success or failure. We start searching from the bottom of image up towards, and initiate the section point at the middle of the whole image (*w/2*). Then search the section point's left and right per row, following three constraints to find a right starting point:

1) Variation of lane gray

In the road binary image extracted by color clustering, the gray value in the region of lane-line must be higher than the road area, so the gray value difference will produce mutations on both sides of the lane line, as shown in Fig.4:

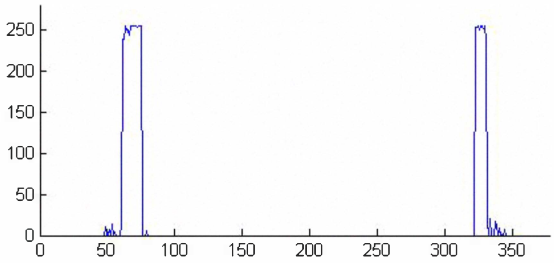


Figure 4. Row scan of the road image

2) Trapezoid template matching

When the starting lane points P_{L1}, P_{R1} on both sides are searched to meet the above condition 1), skip to the 10th row in the search direction to search two boundary points P_{L2}, P_{R2}, these four points should form a trapezoid, as shown in Fig. 5. This matching can be further selected to judge the correctness of the starting point.

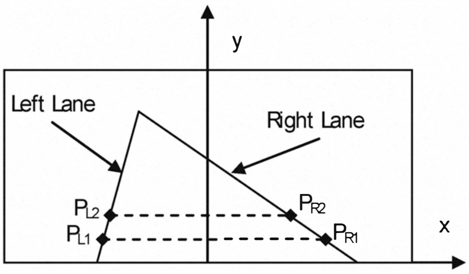


Figure 5. Trapezoid template matching sample diagram

3) Lane width

When one or both sides of the lane line are dashed, the adjacent lane may be mistaken for a starting point-point line. To avoid this from happening, lane width restrictions should be considered. In the standard lane structure, the lane width is for certain. According to the image coordinates and the road coordinate mapping formula equation (6), calculate whether the distance between the starting points on both sides permits the actual driveway width within the range. Pixel $P(X, Y, Z)$ in world coordinate mapping to the image coordinate is $p(x, y)$ complying :

$$\begin{cases} x = f\dfrac{X}{Z\cos\theta - Y\sin\theta} \\ y = f\dfrac{Z\sin\theta + Y\cos\theta}{Z\cos\theta - Y\sin\theta} \end{cases} \tag{6}$$

According to equation (6), supposing that the camera's height H to ground keeps constant, the two points p_1 and p_2 located in left lane and right lane follow equation (7):

$$\begin{cases} x_1 = f\dfrac{X_1}{Z_1\cos\theta - Y_1\sin\theta} \\ x_2 = f\dfrac{X_2}{Z_2\cos\theta - Y_2\sin\theta} \\ Z_1 = Z_2 = r \\ X_1 - X_2 = W_{lane} \\ Y_1 = Y_2 = -H \end{cases} \tag{7}$$

where (x_1, y_1), (x_2, y_2) denote pixel p_1 and p_2 in image coordinate, (X_1, Y_1, Z_1), (X_2, Y_2, Z_2) denote p_1, p_2 in world coordinate, W_{lane} denote the true road width and keep constant. Then we can solve the image road width w_{lane},

$$w_{lane} = x_1 - x_2 = c(X_1 - X_2) = cW_{lane} \tag{8}$$

The bottom of the image, closer to the vehicle, with small disturbance, is ideal for determining starting points. According to the above three constraints, we can accurately determine a starting point. As shown in Fig.7 (a), starting point marked by red dots, red lines mark the trapezoid.

B. Boundary points search

After the starting point determined, then search the next boundary point from candidate boundary points according to certain criteria. In the search path, to the *ith* pixel t_i, next possible boundary point can be determined by the 1×5 neighborhood in the next line L_{i+1} under the guidance measure, as shown in Fig.6, m_{i+1}^k denote the cost measure from pixel t_i to t_i^k ,

$$m_{i+1}^k = |G_x| + \frac{1}{1+|\theta_i - \theta_{i-1}|} \qquad k = 1,2,\cdots,5 \tag{9}$$

where G_x denote the gray gradient of pixel t_i^k in the direction of x, θ_i denote the phase angle of pixel t_i, i.e., the angle defined by pixel t_i and pixel t_{i-1} between x axis.

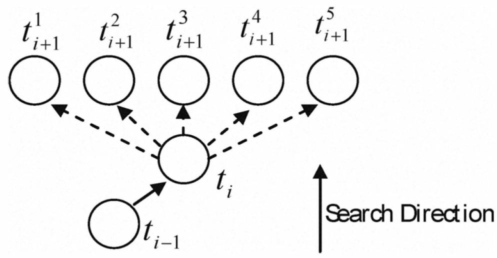


Figure 6. Boundary points search schematic diagram

According to (9), calculate the cost measure of each white point (lane line candidate point), and select the point with greatest as boundary point.

When starting point is defined, in order to search the next point t_2, initiate the value $\theta_1 = \pi/4$ to the left lane, and $\theta_1 = 3\pi/4$ to the right lane.

When point t_2 is detected, original search direction of the trajectory is acquired. According to the principle that lane direction's mutation does not happen and supposing starting point and search direction are constant, search the next boundary point. This way can help to reduce the search region and also get rid of noise points.

C. The termination point determination

Termination point is determined following principles as below: If to a point t_i's next line L_{i+1}, there is no white point in the 1×5 neighborhood or all the white spots' cost measure less than a certain threshold, then the P point selected to be end point. From the starting point to the P point, the length of the search path achieves a certain length, then lane boundary point search completes, otherwise return the above steps to continue searching until the search is completed.

From the analysis above we can see that this boundary point search algorithm narrows the search region to less than 10% of the whole image, and a lot of useless information excluded, the capability of anti-jamming improved. The search results as shown in Fig.7, where (b) is the left lane points and (c) is the right lane points.

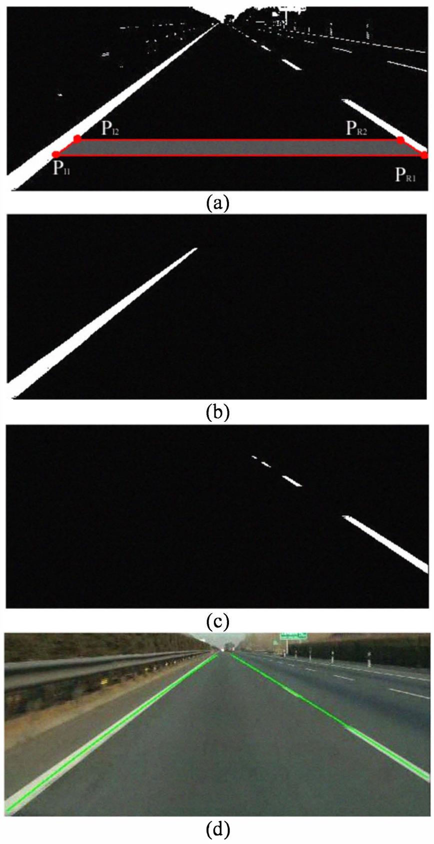


Figure 7. (a) The starting point determination, (b) the left lane detection, (c) the right lane detection, (d) lane lines marked by green lines

IV. LANE MARKING FITTING

Reference [4] adopts polynomial curve, including quadratic curve and cubic curve, to characterize the geometric road model. But in actual, especially to the high-quality roads (highway or roads between cities), lane bend is relatively small. Accordingly, quadratic curve fitting has been able to meet the requirements of most road environments. We use least square method to solve quadratic curve parameters. Suppose that the quadratic curve equation of lane model is:

$$Y = a + bx + cx^2 \tag{10}$$

And suppose *S(a,b,c)* as square sum of difference. Then

$$S(a,b,c) = \sum_{i=1}^{n} (Y_i - a - bx_i - cx_i^2)^2 \tag{11}$$

where (x_i, Y_i) is the row and column number of boundary point i. Calculate the partial derivative of (11), and let it be zero:

$$\begin{cases} \dfrac{\partial S}{\partial a} = -2\sum_{i=1}^{n}(Y_i - a - bx_i - cx_i^2)x_i^0 = 0 \\ \dfrac{\partial S}{\partial b} = -2\sum_{i=1}^{n}(Y_i - a - bx_i - cx_i^2)x_i^1 = 0 \\ \dfrac{\partial S}{\partial c} = -2\sum_{i=1}^{n}(Y_i - a - bx_i - cx_i^2)x_i^2 = 0 \end{cases} \tag{12}$$

$$\begin{cases} a\sum_{i=1}^{n} x_i^0 + b\sum_{i=1}^{n} x_i^1 + c\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} Y_i\, x_i^0 \\ a\sum_{i=1}^{n} x_i^1 + b\sum_{i=1}^{n} x_i^2 + c\sum_{i=1}^{n} x_i^3 = \sum_{i=1}^{n} Y_i\, x_i^1 \\ a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i^3 + c\sum_{i=1}^{n} x_i^4 = \sum_{i=1}^{n} Y_i\, x_i^2 \end{cases} \tag{13}$$

Finally, solve the equation (13) and get the quadratic curve parameters $(a,b,c)^T$, and use $(a,b,c)^T$ to fit the left or right boundary of lane, as shown in Fig.7 (d).

V. BUILDING SIMULINK MODEL

Using Simulink, the Signal Processing Blockset, and the Video and Image Processing Blockset, we first develop a floating-point model of the lane-detection system. We input a video stream to the simulation environment using the From Multimedia File block from the Video and Image Processing Blockset. During simulation, the video data is processed in the Lane Marker Detection and Tracking subsystem, which outputs the detection algorithm results to the To Video Display block for computer visualization (as shown in Fig. 9).

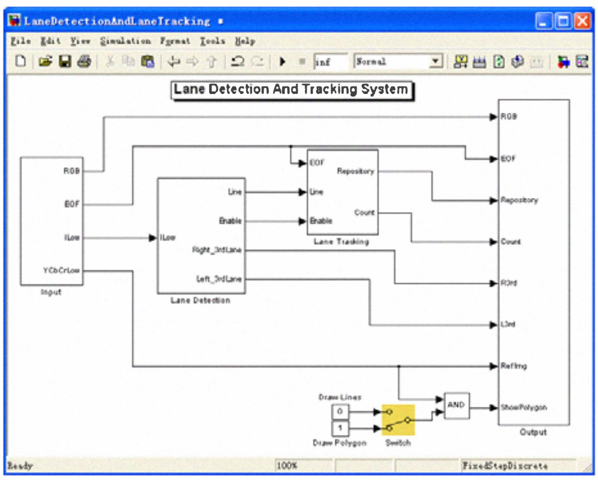


Figure 8. Simulink model's flow chart

(a)T=1.833s (b)T=10.867s

(c)T=19.256s (d) T=43.819s

Figure 9. Simulation results at different time

VI. CONCLUSION

Compared with most of other current algorithms based on using grayscale image directly to detect edge, this algorithm uses color information clustering to extract the region of lane marking, and proves its feasibility. In seeking more accurate detection, the edge detection method can be used to identify the Region Of Interest (ROI), such as the Canny/Hough Estimation of Vanishing Points (CHEVP) algorithm [5]. Along with the technological advances, it is getting easier for us to acquire colorful images that contain abundance color information, which is getting more and more important to improve the accuracy of detection. However, our algorithm requires the quality of image must be high. And being a tentative algorithm, it does not use any edge information, and without gray threshold segmentation. At the same time, we use Heuristic Search Methods to search boundary points, by which the search region can narrow 10% of the whole image. At last, the model simulation result reaches ideal to actual road condition, of course we use Kalman Filter to tracking lane marking in order to reduce processing time. Follow-up work would be combining the part of departure warning to this model, to make this model more roads adaptive and robust.

REFERENCES

[1] H. Hattori, "Stereo for 2D Visual Navigation," *Proc. of the IEEE Intelligent Vehicles Symposium*. USA , 2000:31238.

[2] S. Nedevschi, R. Schmidt and T. Graf, "3D Lane Detection System Based on Stereovision", *IEEE Intelligent Transportation Systems Conference*. USA , 2004 :1612166.

[3] J. Kang, J. Won Choi and I.S. Kweon, "Finding and tracking road lanes using line-snakes," *Proc. of Conference on Intelligent Vehicle*. Japan,1996, pp.189–194.

[4] R. Chapuis, R. Aufrere, and F. Chausse, "Accurate Road Following and Reconstruction by Computer Vision," *IEEE Transl. J. Intelligent Transportation Systems*, 2002, pp.261- 270

[5] Y. Wang, E.K. Teoh and K. Shen, "Lane Detection and Tracking Using B-Snake"*J. Image and Vision Computing*, 2004(22):269-280.

[6] Y. Wang, D. Shen and E.K. Teoh. "Lane Detection Using Catmull-Rom Spline,"*J. IEEE International Conference on Intelligent Vehicles*, 1998:51-57.

[7] K. Kluge and S. Lakshmanan, "A deformable template approach to lane detection," *Proc. IEEE Intelligent Vehicle '95, Detroit*, September 25–26, 1995, pp.54–59.

[8] R. Gopalan, T. Hong, M. Shneier and R. Chellappa, "Video-based Lane Detection using Boosting Principles," Snowbird 2009 submission paper

[9] R. Achanta, F. Estrada, P. Wils and S. SÄusstrun, "Salient Region Detection and Segmentation," International Conference on Computer Vision Systems (ICVS 2008), May 2008.

[10] T.Ohashi, Z.Aghbari and A. Makinouchi, "Hill-climbing Algorithm for Efficient Color-based Image Segmentation," IASTED International Conference On Signal Processing, Pattern Recognition, and Applications (SPPRA 2003), June 2003.

[11] A. Martelli,"An Application of Heuristic Search Methods to Edge and Contour Detection", *Communication of the ACM*, Febraury,1976