

Lane Detection and Kalman-based Linear-Parabolic Lane Tracking

King Hann LIM, Kah Phooi SENG, Li-Minn ANG and Siew Wen CHIN

School of Electrical and Electronic Engineering
The University of Nottingham Malaysia campus
Jalan Broga, Semenyih, 43500 Selangor, Malaysia
{keyx7khl, Jasmine.Seng}@nottingham.edu.my

Abstract— This paper presents a lane detection and linear-parabolic lane tracking system using kalman filtering method. First, the image horizon is detected in a traffic scene to split the sky and road region. Road region is further analyzed with entropy method to remove the road pixels. Lane boundaries are then extracted from the region using lane markings detection. These detected boundaries are tracked in consecutive video frames with a linear-parabolic tracking model. The model parameters are updated with Kalman filtering method. Error-checking is performed iteratively to ensure the performance of the lane estimation model. Simulation results demonstrate good performance of the proposed Kalman-based linear-parabolic lane tracking system with fine parameters update.

Keywords—Lane Detection, Lane Tracking, Intelligent Vehicle, and Driver Assistance System

I. INTRODUCTION

Vision-based lane detection and tracking system has become an important mechanism in vehicular autonomous technology due to the reducing cost of optical sensors and increasing speed of microprocessors. Generally, cameras are mounted on a moving vehicle to detect the lane edges automatically in a traffic scene for road following and lane departure warning [1]. In assistance with lane tracking system, it enhances the performance of lane detection and reduces its computational cost. Hence, an effective visual scanning of lane boundaries and tracking system is highly required to rapidly identify the road geometry.

Numerous techniques for vision-based lane detection and tracking system have been developed over the past decade to localize lane boundaries in various road scenarios [1-4]. Lane-region based method basically classified road and non-road pixels regarding on the road analysis and recognition. Kim et al. [5] proposed a framework for lane detection and lane color recognition using support vector machine. In addition, Kim [4] presented a comparative study on lane marking classifiers and performed tracking system using particle filtering. However, a large amount of samples were needed in advance for the classifier training process.

Feature-based method was usually used to extract the significant features, such as lane-markings from the road. Yu and Jain [6] proposed a multi-resolution Hough Transform to detect lane boundary. Nevertheless, it required camera parameters in the calculation. Weina et al. [7] performed histogram analysis and feature extraction on road image to retrieve the lane markings. They used quadratic curve to match the road boundary with fixed-width lane region.

Moreover, Assidiq et al. [8] performed noise reduction to remove the shadows and applied Hough Transform to detect a pair of hyperbolas edges. However, lane detection without tracking system might consume high computational time.

Model-driven method, on the other hand used a geometric model to characterize the lane. An assumption that the lane geometry could be represented by straight lines, parabolic curves, or snake-like curves was made in this method [3]. Jung and Kelber [9, 10] proposed a linear-parabolic model to follow the lane boundaries and they used a fixed-width tracking mask to detect the possible edges. Furthermore, Wang et al. [11] used B-snake model to mimic the lane geometry with a hypothesis made, that the road boundaries must be parallel on the ground plane. Besides, stereo-vision system [12] was applied to recognize lane markings, but this method had matching difficulty, costly hardware and high computation [7].

In this paper, a new scheme for lane detection and lane model estimation is proposed as shown in Fig. 1. At the beginning, an incoming frame is split into sky and road region. Next, lane region analysis is performed to further distinguish road and lane markings pixels. Road edges are then detected using edge distribution function and weighted Hough transform in order to obtain the orientation and radii for left-right boundaries. With this acquired information, possible edge scanning is performed to detect the nearby edge pixels continuously. Linear-parabolic model is estimated to follow the left-right lane boundaries with the parameters update using Kalman filtering. This paper is organized as: Section II discusses about the lane boundary detection while Section III explains the lane tracking technique. Some simulation results are shown in Section VI and followed by conclusion and future works.

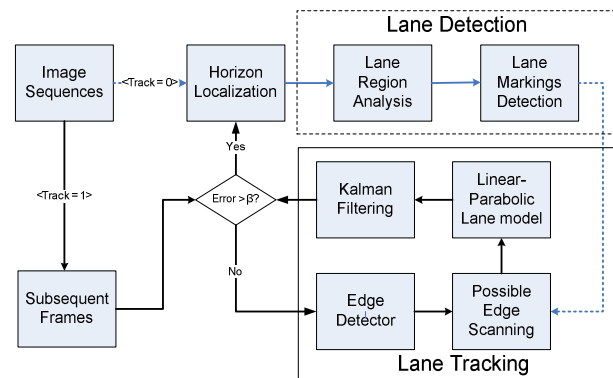


Figure 1. Block diagram of proposed lane detection and tracking system.

II. LANE DETECTION

Lane detection is a crucial task to locate the left-right edges of driving path on a traffic scene image automatically. In this section, a three-stage lane boundary detection is performed, i.e. (i) vertical mean distribution, (ii) lane region analysis, and (iii) lane marking detection.

A. Vertical Mean Distribution

At the preliminary stage, a traffic scene image $I(x,y)$ is divided into sky region and road region using vertical mean distribution [7]. Vertical mean distribution is measured by averaging the gray values of each row on $I(x,y)$ as depicted in Fig. 2(a). The threshold value of regional dividing line is acquired through a minimum search along the vertical mean curve, where the first minimum occurs from the upper curve is the horizon line. This is because sky region usually possesses higher intensity than road pixels, and it might have a big jump of intensity difference as sky pixels approaches ground. The horizon line threshold is applied to generate a road image $R_{roi}(x,y)$ as demonstrated in Figure 2(b), where all vertical coordinates below the threshold are discarded.

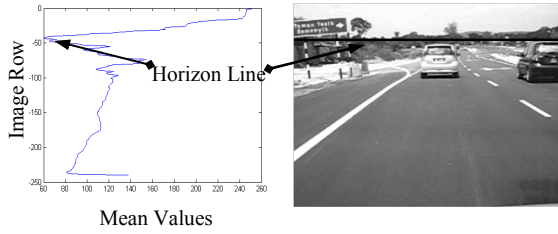


Figure 2. (a) Vertical mean distribution, (b) Road region image $R_{roi}(x,y)$.

B. Lane Region Analysis

Lane region analysis is performed to further classify road region and lane markings. Usually, the bottom region in a road image contains road pixels. By taking a few rows from the bottom of image, the gray value range of road color can easily be obtained and therefore, the color range is applied to remove road pixels from the $R_{roi}(x,y)$ map. The lane region analysis steps are shown as follows:

- (i) Pick 30-60 rows from bottom to avoid the possible existence of inner part of a vehicle at the edge of image.
- (ii) Build a global vote scheme for the selected rows and the maximum vote is recorded as $VOTE$.
- (iii) Record a local maximum $VOTE$ for each row and the maximum of all votes is given as $MAXTHRES$.
- (iv) Obtain $[VOTE - \delta; MAXTHRES + \delta]$ as the road color range where δ is a threshold. Pixels that fall within the range are denoted as road pixels and a binary map $R_{bin}(x,y)$ is formed in Fig. 3(a).
- (v) Invert $R_{bin}(x,y)$ to produce a lane markings map $L_{map}(x,y)$ as illustrated in Fig. 3(b).

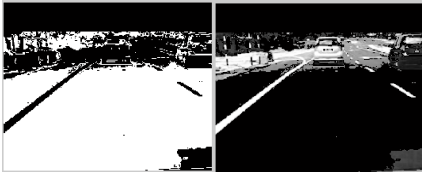


Fig. 3: (a) Lane binary map $R_{bin}(x,y)$, (b) Lane markings map $L_{map}(x,y)$.

C. Lane Markings Detection

The generation of $L_{map}(x,y)$ in the previous stage is, therefore used for lane markings detection. Lane markings are the salient features on road and they are usually used to extract the boundaries of road region. Initially, the gradient magnitude $\nabla L_{map}(x,y)$ and orientation $\theta(x,y)$ of $L_{map}(x,y)$ are

$$\text{denoted as } |\nabla L_{map}(x,y)| \approx |D_x| + |D_y| \text{ \& } \theta(x,y) = \tan^{-1}\left(\frac{D_y}{D_x}\right),$$

where D_x is the horizontal edge map and D_y is the vertical edge map.

With reference to the gradient map and its corresponding orientation, an edge distribution function [10] is constructed, with its x -axis is the orientation ranging from $[-90^\circ; 90^\circ]$ and its y -axis is the gradient value of each orientation bins. The maximum peak acquired in Fig. 4 on the negative angle denotes the right boundary angle. Conversely, the maximum peak at the positive region denotes the left boundary angle. Next, a weighted-gradient Hough Transform [10] is applied to determine the lane's radius $r(\theta)$ based on $r(\theta) = x_o \cos \theta + y_o \sin \theta$, with known $\theta(x_o, y_o)$ values for line construction. The voting bins for each radius are accumulated with the gradient edge values and the maximum vote is selected. Finally, the measured angles and radii are used to construct the left-right lane boundaries.

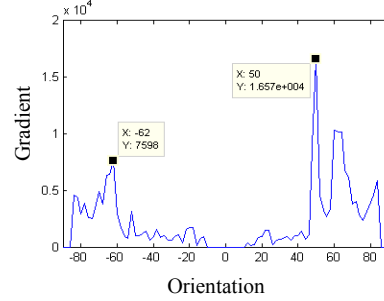


Figure 4. The edge distribution function.

III. LANE TRACKING

After the detection stage, lane tracking system is implemented to restrict the edges searching area on the subsequent frames, and at the same time estimate the lane model to follow the road boundaries. Lane tracking system has three stages, i.e. possible edge scanning, linear parabolic modeling and Kalman filtering.

A. Possible Edge Scanning

As illustrated in Fig. 5(a), the left-right lane boundaries are detected in the lane detection and they are drawn in black lines up to sky-road dividing line. The detected lines are used to scan for the nearby edge pixels on the following image after a Sobel edge detector and these lines would be updated iteratively from the previously estimated lane model. The closest pixels to the estimated lane model are considered as the possible lane edges, which are shown in Fig. 5(b). The

possible edges are useful for the linear-parabolic estimation in the following stage.

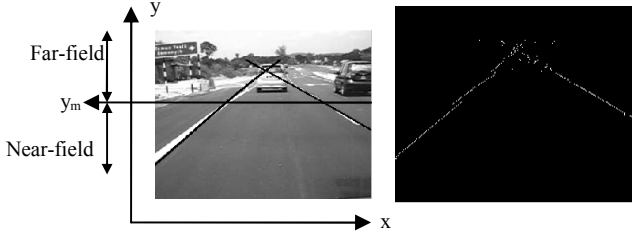


Figure 5. (a) Detected lane boundaries, (b) Possible Edges

B. Linear Parabolic Model

Lane estimation model plays an important role in lane tracking process. In this stage, a simple linear-parabolic tracking model proposed in [9, 10] is used to construct left-right tracked lines. As shown in Fig. 5(a), an image is divided into near and far-fields where y_m is the border between them. A linear model $f(y) = a + by$ whereas $y > y_m$ is applied to follow the straight line in the near-field while a parabolic model $f(y) = c + dy + ey^2$ for $y \leq y_m$ is used to mimic the far-field curvature line. After the imposition of continuity and differentiability conditions on point y_m for the function $f(y)$, $c = \frac{2a + y_m(b-d)}{2}$ and $e = \frac{b-d}{2y_m}$ can be acquired. Hence, the lane model can be written as [9],

$$f(y) = \begin{cases} a + by_{ni} = x_{ni}, & i = 1 \dots m \\ a + \frac{b}{2y_m}(y_{fj}^2 + y_m^2) - \frac{d}{2y_m}(y_{fj}^2 - y_m^2) = x_{fj}, & j = 1 \dots n \end{cases} \quad (1)$$

where (x_{ni}, y_{ni}) denotes the m coordinates of the non-zero pixels of the edge image in near-field for $i = 1, \dots, m$ and (x_{fj}, y_{fj}) represents the n coordinates in far-field for $j = 1, \dots, n$. Moreover, equation (1) can be rearranged into matrix form: $\mathbf{J} = \mathbf{H}\mathbf{A}$, whereas \mathbf{J} is the estimated output of the lane edges, \mathbf{H} is the known variable, \mathbf{A} is the unknown state parameters and x is the desired output pixels. The subscript L and R denotes to left and right lane boundaries parameters respectively.

$$\mathbf{h} = \begin{bmatrix} 1 & y_{ni} & 0 \\ \vdots & \vdots & \vdots \\ 1 & y_{nm} & 0 \\ 1 - \frac{1}{2y_m}(y_{f1}^2 + y_m^2) - \frac{1}{2y_m}(y_{f1} - y_m)^2 \\ \vdots & \vdots & \vdots \\ 1 - \frac{1}{2y_m}(y_{fn}^2 + y_m^2) - \frac{1}{2y_m}(y_{fn} - y_m)^2 \end{bmatrix} \quad (2)$$

$$\mathbf{H} = [\mathbf{h}_L^T, \mathbf{h}_R^T]^T \quad (3)$$

$$\mathbf{A} = [(a, b, d)_L, (a, b, d)_R]^T \quad (4)$$

$$\mathbf{x} = [(x_{n1}, \dots, x_{nm}, x_{f1}, \dots, x_{fn})_L, (x_{n1}, \dots, x_{nm}, x_{f1}, \dots, x_{fn})_R]^T \quad (5)$$

C. Kalman Filtering

Kalman filtering method [13] can be used to update the unknown state parameters for left-right edges. The linear state and measurement equations are defined as:

$$\mathbf{A}_{k+1} = \mathbf{F}_{k+1,k}\mathbf{A}_k + \mathbf{G}_{k+1,k}\mathbf{u}_k + \mathbf{w}_k \quad (2)$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{A}_k + \mathbf{v}_k \quad (3)$$

where the state space \mathbf{A}_k contains the linear-parabolic unknown parameters; $\mathbf{F}_{k+1,k}$ is the transition matrix bringing state $\hat{\mathbf{A}}_k$ from time k to $k+1$, $\mathbf{G}_{k+1,k}$ is the matrix connecting the vector \mathbf{u}_k to the state variables; \mathbf{w}_k is known as process noise; \mathbf{x}_k is the measurement output; \mathbf{H}_k is the observation model that maps the true state space to the observed space; and \mathbf{v}_k is the measurement noise.

As demonstrated in Fig. 6, \mathbf{P}_k^- is a priori estimate error covariance and \mathbf{P}_k is a posteriori estimate error covariance. The estimated state, $\hat{\mathbf{A}}_k$ is updated with the estimated gain \mathbf{K}_k . The predicted state $\hat{\mathbf{A}}_k^-$ is used to draw a line for detecting the possible edges in the next frame. Error checking is performed consecutively to ensure the stability of the tracking system. Once the error is larger than a certain threshold (β), a full lane detection system has to be carried out on the entire picture.

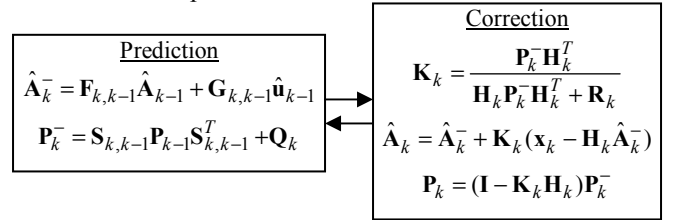


Figure 6: Kalman filtering method.

IV. SIMULATION RESULTS

All results were generated using Matlab 2007a with the Core 2 Duo processor at 1.8GHz with 1GB RAM. A video sample #1 was captured under sunny day condition at around 12 p.m. in an urban area using camera. Another video sample #2 was obtained from [14]. These video samples were used to verify the performance of the proposed lane detection and tracking system. Initial value for $\beta = 30$ and $\delta = 25$ were set in these experiments.

Some lane detection results were demonstrated in Fig. 7. Fig. 7(a) showed that the driving path was successfully retrieved in frontal view. The detected lane boundaries were therefore used to estimate the left-right lane model in the next step. Some other lane detection results were demonstrated in Fig. 7(b) & (c). As a result, left-right lane boundaries were expected in the images.

Once the road boundaries were located, lane modeling was employed to follow the road boundaries. In Fig. 8, the lane boundaries were detected and estimated properly in video sample #1 continuously with the proposed tracking system. Lane tracking system was then applied to video sample #2 and it was successfully tracked in Fig. 9. As compared to the results in Fig. 10(e)-(g) using method [9], our proposed tracking system has better performance with better state parameters updating scheme. All parameters for

left and right boundaries are updated at once with the Kalman filtering method.

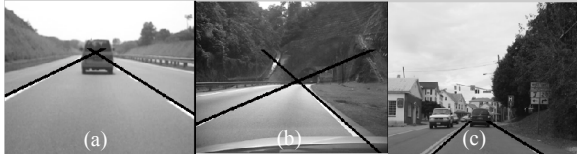


Figure 7. Lane detection results:(a)on highway, (b)before tunnel, (c)in city.

V. CONCLUSION

A vision-based lane detection and Kalman-based linear-parabolic tracking system has been presented in this paper. An image is first performed with horizon localization and lane region analysis. It is followed by lane markings extraction to obtain the left-right lane boundaries. The detected left-right boundaries are used to predict a linear-parabolic model. This model is continuously estimated and tracked using Kalman filtering method. The possible edges are to be found nearby to the estimated lane boundaries. No camera parameters are involved in the proposed system. In addition, the performance of lane detection is further enhanced with the assistance of lane model estimation in faster processing speed.

REFERENCES

- [1] J. M. Armingol, A. de la Escalera, C. Hilario, J. M. Collado, J. P. Carrasco, M. J. Flores, J. M. Pastor, and F. J. Rodríguez, "IVVI: Intelligent vehicle based on visual information," *Robotics and Autonomous Systems*, vol. 55, pp. 904-916, 2007.
- [2] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *Intelligent Transportation Systems*, vol. 7, pp. 20-37, 2006.
- [3] Y. Zhou, R. Xu, X. F. Hu and Q. T. Ye, "A robust lane detection and tracking method based on computer vision," *Measurement Science and Technology*, vol. 17, p. 736, 2006.
- [4] Z. W. Kim., "Robust Lane Detection and Tracking in Challenging Scenarios," *Intelligent Transportation Systems*, vol. 9, pp. 16-26, 2008.
- [5] S. H. Kim, J. H. Park, S. I. Cho, S. Y. Park, K. S. Lee, and K. H. Choi, "Robust Lane Detection for Video-Based Navigation Systems," in *Tools with Artificial Intelligence, ICTAI 2007*, pp. 535-538.
- [6] B. Yu and A. K. Jain, "Lane boundary detection using a multiresolution Hough transform," in *Image Processing*, 1997, pp. 748-751 vol.2.
- [7] L. Weina, Z. Yucai, M. Yu Quan, and L. Tao, "An Integrated Approach to Recognition of Lane Marking and Road Boundary," in *Knowledge Discovery and Data Mining*, 2008, pp. 649-653.
- [8] A. A. M. Assidiq, O. O. Khalifa, M. R. Islam, and S. Khan, "Real time lane detection for autonomous vehicles," in *Computer and Communication Engineering*, 2008, pp. 82-88.
- [9] C. R. Jung and C. R. Kelber, "A robust linear-parabolic model for lane following," in *Computer Graphics and Image Processing*, 2004, pp. 72-79.
- [10] C. R. Jung and C. R. Kelber, "Lane following and lane departure using a linear-parabolic model," *Image and Vision Computing*, vol. 23, pp. 1192-1202, 2005.
- [11] Y. Wang, E. K. Teoh, and D. G. Shen, "Lane detection and tracking using B-Snake," *Image and Vision Computing*, vol. 22, pp. 269-280, 2004.
- [12] N. Benmansour, R. Labayrade, D. Aubert, S. Glaser, and D. Gruyer, "A model driven 3D lane detection system using stereovision," in *Control, Automation, Robotics and Vision*, 2008, pp. 1277-1282.
- [13] E. Cuevas, D. Zaldivar, and R. Rojas, "Kalman filter for vision tracking," *Freie University Berlin, Berlin, Technical Report B 05-12*, 2005.
- [14] The Vision and Automation Systems Center, "may30_90 - 160 frames of a sunny 1-lane road ": Robotics Institute in the School of Computer Science, Carnegie Mellon University, 1997-2003.

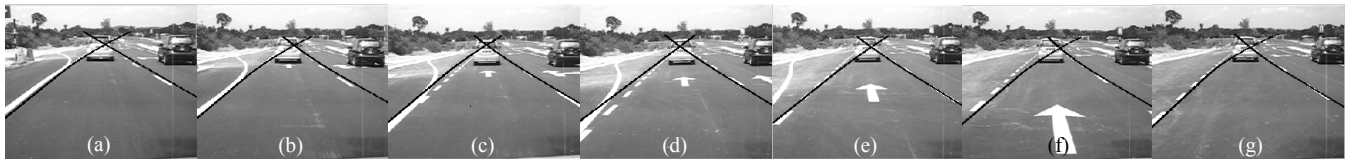


Figure 8(a)-(g). Lane tracking results for video sample #1.

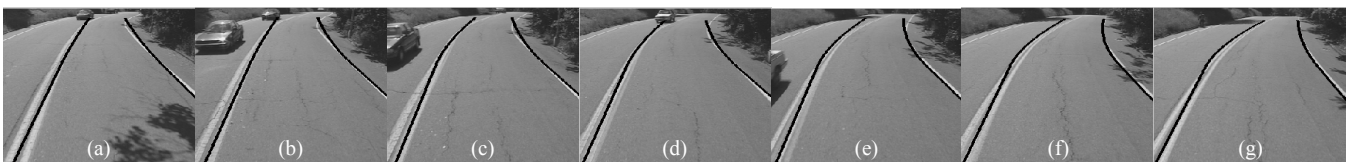


Figure 9(a)-(g). Lane tracking results for video sample #2.

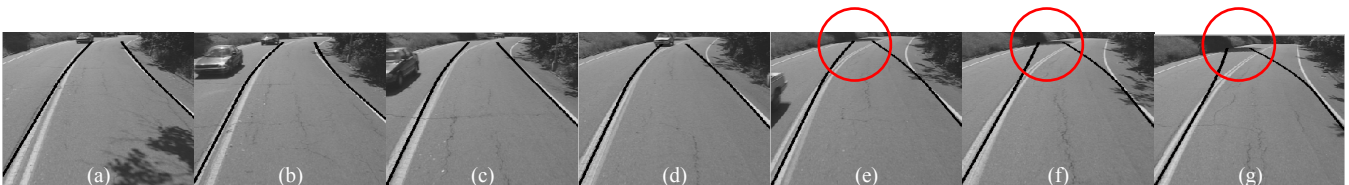


Figure 10(a)-(g). Lane tracking results using method [9] for video sample #2. Red circles denoted the lost tracked model estimation.