Lane Departure Warning Tracking System Based on Score Mechanism

Omar G. Lotfy[1], Ahmed A. Kassem[1], Emad M. Nassief[1], Hassan A. Ali[1], Mario R. Ayoub[1], Magdy A. El-Moursy[2,3], Mohammed M. Farag[1]
[1]Electrical Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt
[2]Microelectronics Department, Electronic Research Institute, Cairo, Egypt
[3]Mentor Graphics Corporation, Cairo, Egypt
E-mails: omar_aaa2006@hotmail.com, ahmedalaa1193@gmail.com, emadmoheb47@yahoo.com, cena20105@gmail.com, eng.mario_swe@yahoo.com, magdy_el-moursy@mentor.com, mmorsy@alexu.edu.eg

***Abstract*—This paper presents a new lane tracking algorithm for the lane departure warning system without using Kalman filter. The system is capable of extracting the true lane boundaries from all detected lines including noise in the frame and estimates its future position. The new algorithm uses the score mechanism to trace the appearance of lines in previous frames using a score variable which indicates the number of frame lines that have been detected and scores state of each line (increasing or decreasing). The lateral shift is calculated from prior knowledge of lane boundary positions in previous frames. Moreover, the algorithm introduces a hysteresis loop of line score varying the number of predicted frames for line position which has a significant impact in smoothing and stabilizing the tracking process.**

Keywords— Drive Assistance Systems (DAS), Lane Detection (LD), Inverse Perspective Mapping (IPM), Lane Detection and Departure Warning (LDDW), Line Tracking, Hysteresis Loop, Lane Departure Warning (LDW), True Lane (TL).

I. INTRODUCTION

Each year, thousands of road accidents across the whole world cause serious injuries for people and reap the lives of others. Research work in the intelligent transportation systems has resulted in emerging a lot of advanced technologies such as Advanced Driver Assistance Systems (ADAS) including the Lane Departure Warning (LDW) system. This technology is designed to warn a the driver when the vehicle starts to deviate from its lane and usually relies on a camera installed on the windshield mirror that recognizes the lane boundaries and road structure. Some systems automatically take actions to return the vehicle to its lane by overriding the driver's control on the vehicle and applying the required force on the steering wheel.

LDW is a vision-based system which relies on image processing and computer vision algorithms. Many LDW systems employ lane detection algorithms to extract road lane marking from the taken images. In this work, the Lane Detection and Departure Warning (*LDDW*) system is advanced as shown in Fig. 1. The IPM block constructs a bird's eye view of the road in front of the vehicle and extracts the edge of objects using edge detection techniques. Irrelevant edges must be eliminated by converting grayscale to a binary image. Sub-regions are defined in the image. Those regions are the candidates to contain boundaries of the lane using simplified Hough transform. Line detection mechanism detects the best fit vertical lines. The algorithm is used to find points which belong to the line and then line fitting is applied to approximately fit these points into lines.

Unfortunately, surrounding vehicles, road drawing, etc. affect system performance. Line tracking based on the score

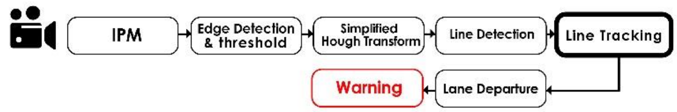


Figure 1: Block diagram of the *LDDW* system

mechanism has an important role in improving lane detection performance. It gathers information by observing the lane boundaries in previous frames. This information is used to examine the position of all lane boundaries that are detected in a single frame and extract the possible lines in the current frame. This results in eliminating noise due to road drawing or other noise. Furthermore, line tracking predicts lane boundaries in case of lane detection failure for a certain number of frames which is defined by the system.

The second part of the LDDW system is lane departure which determines the vehicle's position relative to the current lane boundaries and produces an alarm to warn driver whether the vehicle is located in a safe area or about to depart. The remaining of this paper is organized as follows. In Section II, lane detection procedures are described. The lane departure algorithm is advanced in Section III. In Section IV, experimental results of the proposed tracking algorithm are demonstrated. Conclusions are portrayed in Section V.

II. LANE DETECTION

Lane detection is performed in different steps. In subsection *II.A*, Inverse Perspective Mapping (*IPM*) is presented. Edge Detections is described in Subsection *II.B*. Hough Transform is discussed in Subsection *II.C.* Line Detection is demonstrated in Subsection *II.D*. Line Tracking is introduced in Subsection *II.E*. Finally, Lane Departure is summarized in Subsection *II.F*.

A. Inverse Perspective Mapping (IPM)

The first step in lane detection requires Inverse Perspective Mapping (IPM) to be performed. It is widely used in vision-based driving assistance algorithms [1]. When the required parameters such as the camera position, orientation, and camera intrinsic parameters are completely known and fixed. IPM is also used for flat road situations. It maps the 2D camera coordinate to the 3D world coordinates [2] where perspective effects are eliminated to construct a top view of the road in front of the vehicle. This causes each pixel to represent the same portion of the road and makes the image similar to reality. The width of the road markings is invariant with their position within the whole image. Parallel lines in the real world are restored again, which make the detection of lane easier. An image from camera frame and its bird's eye view are shown in Fig. 2. Edge detection threshold follows IPM as described in next section.

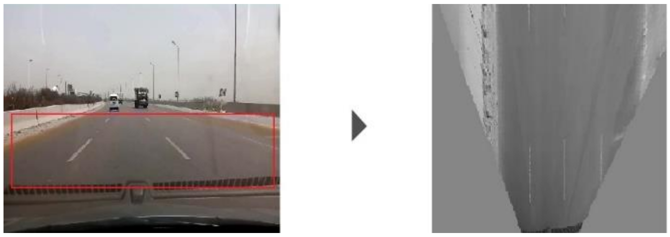


Figure 2: An example of IPM processing results.

B. Edge Detection Threshold

The IPM image is convolved with only one kernel. This kernel is 2D kernel which is obtained from two 1D kernels. The kernel is Nx1 Gaussian kernel [3] which is required to smooth and filter the image from signal components that have high frequencies. Gaussian kernel is presented in Equation (1).

$$FG(y) \; = exp(\frac{-y^2}{2\sigma^2}). \qquad (1)$$

The second kernel is 1xN Laplacian kernel which is sensitive to vertical lines and is used in detecting edges with high gradient of local intensity [6]. Equation (2) represents Laplacian kernel.

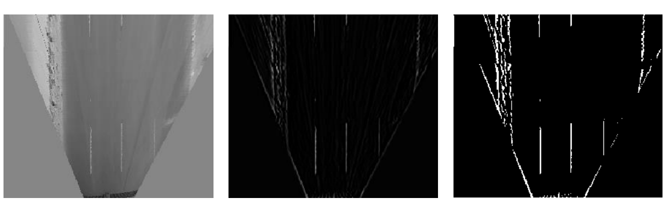
$$FL(x) = \frac{1}{\sigma^2} exp\left(\frac{-x^2}{2\sigma^2}\right)\left(1 - \frac{x^2}{\sigma^2}\right). \qquad (2)$$

These two 1D kernels are multiplied to obtain a 2D kernel that contains both features of Gaussian smoothing and Laplacian edge detection in one kernel as illustrated in Fig. 3.

(a) 1D Gaussian kernel (b) 1D laplacian kernel (c) 2D output kernel

Figure 3 The used Kernels in convolution

Instead of convolving the whole frame with the Gaussian kernel then convolving it with the Laplacian kernel in two separate steps [4], the proposed technique convolves the image with only one kernel in one step as shown in Fig. 3, which reduces the computational effort of the convolution operation. The image convolved with the 2D kernel is shown in Fig. 4.



(a) IPM. (b) After convolution (c) binary image

Figure 4: The three transformation steps to binary image

As the Laplacian kernel is involved into the y-direction of the image, this kernel is more likely to detect vertical lines than horizontal ones which facilitates detecting road lanes. Then the image is converted to a binary image. OTSU algorithm [5] is assigned for image thresholding as shown in Fig. 4(c). A Hough Transform is then needed as described in section *C*.

C. Simplified Hough Transform

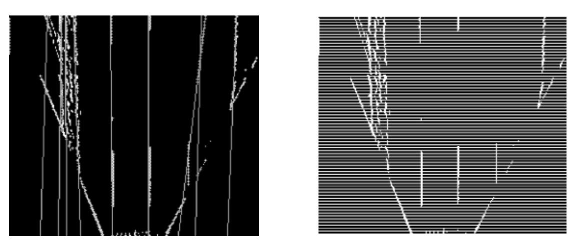
The Hough transform [6, 7] is used to detect lines with all possible orientations taking advantage of using IPM. Lanes consist of vertical straight lines allow using the simplified Hough transform [4, 8]. For each column, the simplified Hough transform adds the column elements of the binary image and obtains sub-regions which are more likely to contain lines. The boundaries of these sub-regions are specified by checking if the summation of these columns is higher than a specific threshold or not. This threshold depends on the minimum number of pixels for a line that could belong to a lane, which can be calculated by measuring the average length of lanes on a flat road using IPM calculations. The following sections contain a description of the Line Detection algorithm.

D. Line Detection

The line detection algorithm makes use of the modified Hough Transform and is inspired by the RANSAC algorithm [9]. It consists of two main parts:

1) Point Detection

This part of the algorithm concentrates on detecting points on lines—potential lanes—as shown in Fig. 5(a). For each window, points to be checked are chosen according to a specific pattern. As for the X-axis, horizontal lines are taken with equally separated distances as shown in Fig. 5(b). For each horizontal line, several points are checked until a point on the line is found—point with value equals to 255—or it exceeds the height of the window. As for the Y-axis, it depends on the probability of finding a point on the line in a specific column. This probability depends on the value of each column, which is calculated before in the Hough Transform. The column with the highest probability is used first. If the considered point is not on the line, the column with the lowest probability is then used and so on until all columns in a window are finished. This process is repeated for all horizontal lines in a window, and then repeated for all windows in order to have a cluster of points for each window. The advantage of adopting this algorithm instead of RANSAC is that RANSAC is completely random, but this algorithm exploits probability to find points on lines in shorter time which eventually improves the overall performance of the proposed algorithm.



(a) The detected lines (b) Equally separated horizontal lines

Figure 5: Detecting points steps

2) Line Fitting

For each window, the detected points are processed using the Least Mean Square (LMS) algorithm [10], and an output of the c & m parameters is calculated, where *c* is the intercepted art of y-axis and *m* is the slope indicating a first-order equation of a line that could be a lane as shown in Table 1.

Table 1: Columns of a window arranged according to its value which indicates its probability.

Column	25	27	23	24	26
Value	1760	1328	1241	984	846

E. Tracking Based on Score Mechanism

Line tracking is implemented to extract lane boundaries and differentiate it from false lines; which have been detected because of the edges of moving vehicles, road drawing, and vehicles' shadow. It is also able to predict the boundaries if the system fails to detect the line for a certain number of frames. This algorithm mainly consists of two lists: the detected list representing the position of lines in the current frame; and the tracked list representing analysis and status of the detected lines in prior frames as follows:

- Position (*m* & *c*).
- Score: Number of times line was tracked in prior frames.
- State: Indicate if the score is increasing or decreasing.
- True Lane (TL): A flag classifies line as true lane or noise.

The TL flag as shown in Fig. 6 traces the score value whenever it reaches the maximum score and sets the flag. Once a score of the line has reached the maximum score, the system sets the TL flag and cannot be cleared until it reaches the minimum score. The middle region is called the dead zone that does not instantly affect the state of the TL flag which enhances the system stability. This zone contributes in predicting lanes whose TL flag has been set before for a certain number of frames. This number depends on the difference between the maximum and minimum scores. The number of predicted frames for each line position is set as a variable that mainly depends on the score.

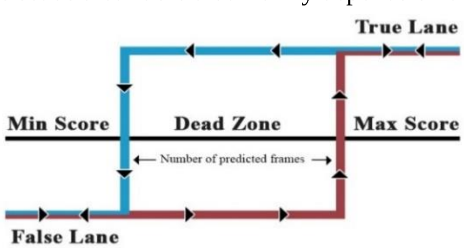


Figure 6: Hysteresis loop of detected lines

The score mechanism flow chart is shown in Fig. 7. It begins with looping around the track list searching for the best match with an input line in the detected list as shown in Fig. 8. Afterwards, the input line is assigned to the most matching tracked line, the information of this tracked line is updated as:

- Saturate score if maximum, otherwise increment.
- Update its state as increasing.
- Update its position.
- Set the TL flag if it has reached the maximum.

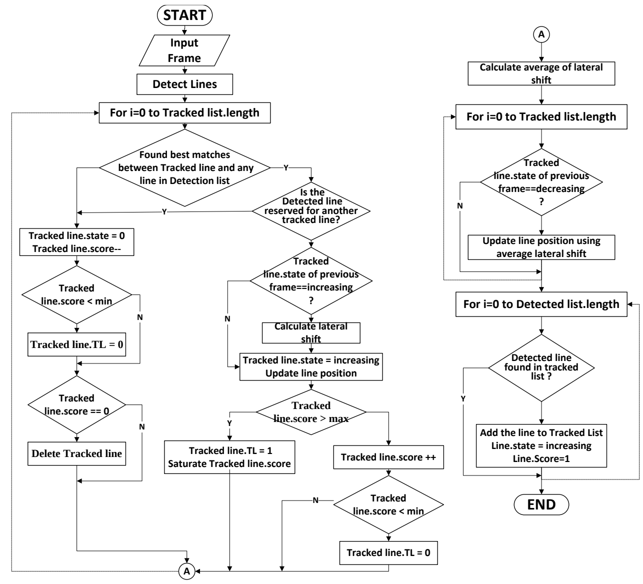


Figure 7: Flow chart for tracking the score mechanism

For all lines in the tracked list which failed to be matched with input lines, their information is updated as:

- Delete the line if its score is zero; otherwise decrement.
- Update its state as decreasing.
- Clear the TL flag if it reaches the minimum.

Updating Line positions in the tracked list whose state is decreasing is essential in order to predict its position in next frames. The lateral shift of the vehicle is obtained by calculating average shifts of lines in the tracked list whose state is increasing. The last step is inserting the input lines from the detected list which failed to find best match lines in tracked list. The tracked list is updated in each frame as every detected line that is inserted into the list could be a potential lane boundary.

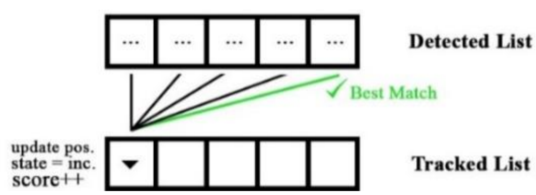


Figure 8: Search step for every tracked line in detected list

F. Lane Departure

The lane departure algorithm starts with selecting the two lines indicating the vehicle lane and rejecting any other lines. Thereafter, the vehicle's position is calculated, and an alarm output is launched based on the calculated position [11, 12].

III. EXPERMINTAL RESULTS

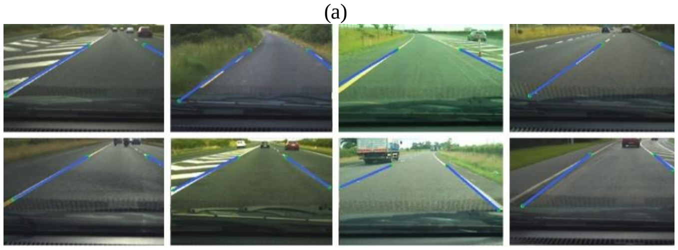
The proposed LDDW system has been implemented in C using OpenCV 2.4.11 library, the resolution of all videos is 640×480 captured with a 13-megapixel camera installed in the test vehicle at 1.2m height and an inclination angle of 8°. The algorithm is running at 65 Hz and tested on ODROID XU4 board with Cortex™-A15 2GHz and Cortex™-A7 Octa core CPUs and 2 GB of RAM.

As illustrated in Fig. 9(a), the proposed LD algorithm succeeds in detecting the lane boundaries under a variety of road conditions. Moreover, the proposed algorithm is applied to another list of videos [13] as shown in Fig. 9(b) to test the performance on different types of roads. A different scenario is shown in Fig. 9(c) where lane marks instantly disappear on the road for a certain number of frames, the line tracking start to predict the lane position based on the value of its score that is calculated from previous frames. Three similar conditions are depicted in Fig. 9(d) where the left lane boundary was covered by a vehicle which makes the system unable to detect the line. The tracking mechanism ignores the edge of the vehicle effect which had been detected in the current frames and predict the left boundary position. In Fig. 9(e), there are some frames selected from [13] to demonstrate that the algorithm is capable of differentiating between the lane marks and the road draws which had been detected by the system in the current frames. The resulting videos can be viewed at [14].

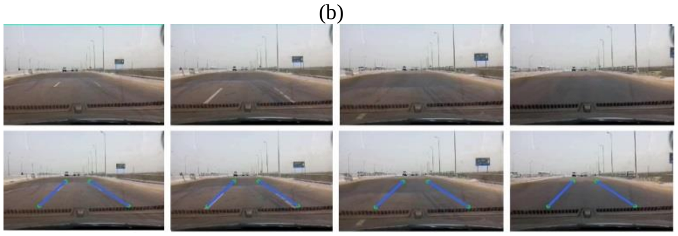
IV. Conclusions

In this paper, a new lane departure warning system, namely LDDW, based on the tracking score mechanism is introduced. Field test results indicate that the proposed tracking algorithm provides an accurate estimation of lane boundaries and a powerful capability to extract the correct lane from detected lines in video frames. The lateral shift of lines position is calculated to enhance predicting lane boundaries in future frames assuming that the system fails to detect a line in the current frame. Hysteresis loop has proved its efficiency in stabilizing the estimation and the detection processes of lane boundaries. Also, its fast response in eliminating road drawings and undesired noise under a variety of road structures and conditions is experimentally validated.

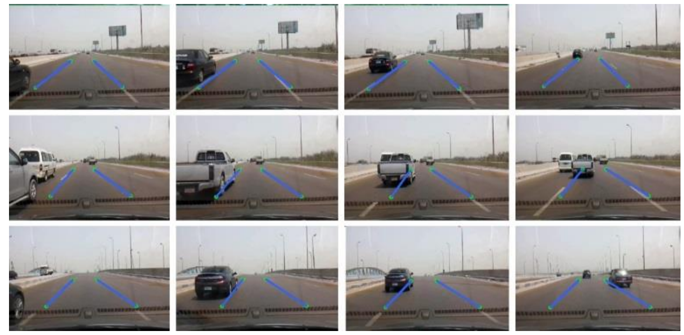
(a)



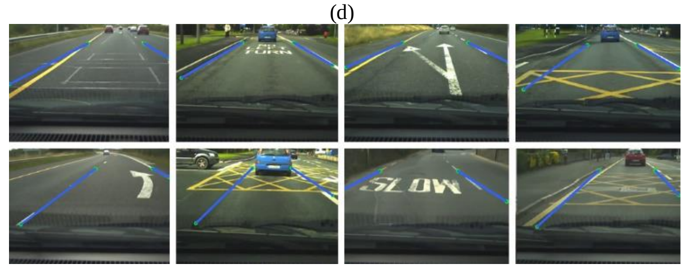
(b)



(c)



(d)



(e)

Figure 9: Lane detection examples frames

V. References

[1] Broggi, Alberto. "A massively parallel approach to real-time vision-based road markings detection." *Intelligent Vehicles' 95 Symposium., Proceedings of the*. IEEE, 1995.

[2] Oliveira, Miguel, Vitor Santos, and Angel D. Sappa. "Multimodal inverse perspective mapping." *Information Fusion* 24 (2015): 108-121.

[3] Bradski, Gary, and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008. Chapter 11

[4] Aly, Mohamed. "Real time detection of lane markers in urban streets."*Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008.

[5] Gonzalez, Rafael C., and Richard E. Woods. "Digital image processing." (2002).

[6] Davies, E. Roy. *Computer and machine vision: theory, algorithms, practicalities.* Academic Press, 2012.

[7] Duda, Richard O., and Peter E. Hart. "Use of the Hough transformation to detect lines and curves in pictures." *Communications of the ACM* 15.1 (1972): 11-15.

[8] Taubel, Gregory, and Jiann-Shiou Yang. "A lane departure warning system based on the integration of the optical flow and Hough transform methods."*Control and Automation (ICCA), 2013 10th IEEE International Conference on*. IEEE, 2013.

[9] David, A., and Ponce Jean. "Computer vision: a modern approach." (2002).

[10] Huo, Chih-Li, Yu-Hsaing Yu, and Tsung-Ying Sun. "Lane departure warning system based on dynamic vanishing point adjustment." *Consumer Electronics (GCCE), 2012 IEEE 1st Global Conference on*. IEEE, 2012.

[11] Hsiao, Pei-Yung, et al. "An embedded lane departure warning system."*Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on*. IEEE, 2011.

[12] Jung, Claudio Rosito, and Christian Roberto Kelber. "A lane departure warning system based on a linear-parabolic lane model." *Intelligent Vehicles Symposium, 2004 IEEE*. IEEE, 2004.

[13] Cualain,D.:'Lane departure detection', https://www.youtube.com/watch?v=Fa1Z4aksuu8&list=PL21A3E1820EF84AC0&feature=plpp_play_all

[14] https://www.youtube.com/playlist?list=PLmXnuvPVwbRu5f3r-r9PDlpJve8FjRhZA