IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 15, 2020, accepted April 21, 2020, date of publication May 4, 2020, date of current version May 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991930

A Lane Tracking Method Based on Progressive Probabilistic Hough Transform

MEHREZ MARZOUGUI[1,2], AREEJ ALASIRY[1], YASSIN KORTLI[2], AND JAMEL BAILI[1]

[1]College of Computer Science, King Khalid University, Abha 61413, Saudi Arabia
[2]Electronics and Micro-Electronics Laboratory, University of Monastir, Monastir 5000, Tunisia

Corresponding author: Mehrez Marzougui (mhrez@kku.edu.sa)

This work was supported by the Deanship of Scientific Research at King Khalid University (KKU) for funding this work through General Research Project under Grant G.R.P-410-39.

ABSTRACT Lane departure warning systems have gained considerable research interest in the past decade for its promising usage in automotive, where lane detection and tracking is applied. However, it is a challenging task to improve the robustness of lane detection due to environmental factors, such as perspective effect, possible low visibility of lanes, and partial occlusions. To deal with these issues, we propose a reliable vision-based real-time lane markings detection and tracking system that can adapt to various environmental conditions. The lane detection is composed of three stages: pre-processing, Adaptive Region of Interest (AROI) setting, and lane marking detection and tracking. In the pre-processing stage, smoothing and edge detection operators are applied on input frames to automatically obtain binary images, then, lane markings segmentation are carried out. After that, An Adaptive Region of Interest is extracted to reduce the computational complexity. In the subsequent detection stages, Kalman filter is employed to track road boundaries detected in the AROI using Progressive Probabilistic Hough Transform (PPHT) in the next frame. Based on road boundaries and the vehicle's position, the proposed algorithm decides if the vehicle has drifted off the lane. For the performance evaluation of lane detection and tracking, real-life datasets for both urban roads and highways in various lighting conditions are used. Applying our method to Catltech dataset, the average correct detection rate is 93.82%. In addition, the proposed method outperforms that of the state-of-the-art methods in processing time (21.54ms/frame).

INDEX TERMS Advanced driving assistance systems (ADAS), lane detection and tracking, lane departure warning system (LDWS), progressive probabilistic Hough transform (PPHT), adaptive region of interest (AROI).

I. INTRODUCTION

over the past few years, Advanced Driver Assistance Systems (ADAS) has emerged as an important research field to enhance automotive safety [1]. With the objective of traffic accidents avoidance, ADAS include a lane departure warning to alert drivers to take a corrective action when the vehicle tends to drift off a lane. However, effective lane detection is very challenging, due to the following factors: (i) types of roads, such as highways, urban roads, and unstructured roads; (ii) different lighting conditions, e.g., backlighting or low light condition; and (iii) occlusion caused by various obstacles, such as passing traffic, pedestrians, or shadows. Many approaches have been proposed to address these challenges by employing various technologies such as Global Positioning System (GPS), Light Detection and Ranging (LiDAR), as well as optical sensors, i.e., cameras [2]. The main advantage of employing GPS or LiDAR is that regardless of the weather condition the data is directly acquired. However, while the resolution of GPS images is only 10-15 m, the signal may get interrupted as when a vehicle goes through a tunnel. Although, LiDAR is popularly used due to its high resolution images, and highly reliable sensor, the cost is very high. The most typical passive sensor is an optical sensor, like a camera [2]. A standard vision-based Lane Departure Warning System (LDWS) is mainly composed of three parts: Lane detection, lane tracking, and lane departure warning [3]. Sequences of road images are captured by an onboard-camera installed in the high up of the windshield, and lane departure alerts occur to protect the driver from

The associate editor coordinating the review of this manuscript and approving it for publication was Lefei Zhang.

some unintended dangerous driving situations. Vision-based approaches, such as Standard Hough Transform (SHT) [4]–[6], Inverse Perspective Mapping (IPM) [7], Random Sample Consensus (RANSAC) [8], Hyperbola-Pair Model [9], and clustering method [10] are implemented to ensure proper road lane markings detection and tracking. Particle filter [11] and Kalman filter [5], [7] are frequently used for tracking methods. In the absence of road lane markings, the tracking module initializes a new road lane candidate and finds out settings of the traffic lane for the current frame based on previous states. These approaches perform well under satisfactory road conditions such as clear markings and few obstacles. However, in the case of different lighting conditions, the performance of these modules drops while the computational complexity rises. In this paper, we propose a passive type of vision-based real-time lane detection and tracking, where different lighting conditions, and different road types, i.e., straight and curved, are considered [5], [7], [12]. The main contribution of this research is as follows:

- In the pre-processing stage, images are transformed to grey scale image and smoothed. Then, edge detection is applied using Sobel filter. Finally, Otsu's thresholding is applied to automatically obtain binary images and to cope with the lighting challenges.
- According to the gradient information obtained by the Otsu segmentation step, we propose to identify a simple Adaptive Region of Interest (AROI) by using a horizon line in order to reduce the computational complexity.
- Since SHT presents a significantly high false positive rate and high computational complexity [4], PPHT is used to cope with these limitations.
- Kalman filter is used to track both borders of each lane markings. This additional lane-tracking step increases the probability to detect lane markings in poor conditions and improves the robustness and the efficiency of road lane detection.

For performance evaluation of the proposed method, clips from real-life Caltech dataset have been used to evaluate the lane boundary detection rate and the time complexity.

The rest of this paper is organized as follows. Section two reviews research work related to detection and tracking of lane markings systems presented in the literature. In section three, we introduce the proposed lane detection and tracking method. Section four discusses the obtained experimental results for different road types (straight or curved) under various lighting conditions. Section five concludes the paper.

II. RELATED WORK

Real time lane detection and tracking has been one of the most active fields for researchers in the last few years. Preprocessing is the first stage of lane detection that is employed to remove irrelevant noise, get accurate road lane information, and to ensure successful detection in the subsequent steps. It includes image smoothing using conventional filters such as Gaussian filter [5], [7], [13], and Median filter [11], [13], [14]. In addition, image segmentation is performed where features such as edge and light intensity are used. Moreover, gradient-enhancing method is applied to extract road lane markings in different lighting conditions [15] using conventional edge extraction operators. Compared to Canny, Robert and Prewitt edge detection operators, Sobel operator is less complicated and less sensitive to noise [4]. The drawback of a Canny edge detector is its sensibility to the edge intensity information [6], [8], which increases the false positives detection. In the case of curved road lane markings, edge detectors are not enough sensitive to different lighting conditions.

After the pre-processing step, the extraction of a Region of Interest (ROI) is performed in lane detection. It is a simple but effective method to reduce redundant image data quantity on the one hand, and lanes always appear within the predetermined region of the image on the other hand. In fact, if the lane detection is done only in the ROI and not in the whole image, the effect of environmental noise such as rain, fog, or other poor weather conditions can be reduced. Previous research works select the ROI as the bottom side of the image [17], [18], while others use the vanishing point detection technique [4], [6], [8], [14] to define the ROI. Estimating a vanishing point can be helpful in detecting lanes, because parallel lines converge on the vanishing point in a projected 2-D image. Generally, these methods perform well on straight road images, but easily disturbed by noise and potentially lead to a detection error in poor road conditions. This can be caused by insufficient quantity of paint used for marking the lane boundary, environmental effects (shadow, or occlusion), or illumination conditions.

Once the ROI is confirmed, some lane detection approaches apply Warp Perspective Mapping (WPM) [19], while others apply Inverse Perspective Mapping (IPM) [7], [20] to get the bird's eye view of the road image based on the assumption of parallel lane boundaries. Although, the effective detection portion, such as the lane, can account for approximately two-thirds of the area of the image, it may still contains noise. For instance, if the vanishing points were incorrected, processing time may increases and might leads to incorrect ROI extraction. Therefore, these methods cannot meet the real-time requirements of the automated vehicle. In order to reduce the computational complexity, here, we propose to define a simple and adaptive ROI. This is achieved by using a horizon line, dividing the scene into road and sky regions, that is based on the gradient information obtained by the Otsu process.

Methods applied on ROI include two main steps: lane tracking and departure warning. Many methods have been proposed in literature for straight and curved lane detection and tracking. Hough Transform is the most common technique used for straight lane markings detection [4]–[7], [20], [21]. However, it faces serious challenges of computational complexity and high false positive rate [4], [6], [7]. Other detection methods such as hyperbola and parabola are used and presented in [18], [20], [22]. Catmull-Rom, cubic curve, B-spline [8], [23], and RANSAC (Random Sample

Consensus) [12], [14], [24], [25] are also widely used. Lane tracking involves the prediction of future road lane marking positions while decreasing false detections. The most widely used tracking methods are Kalman Filter [5], [14], [20], [23], [26] and Particle Filter [11], [18], [27]. In [12], authors proposed RANSAC algorithm to track lane between two consecutive frames using Kalman Filter. Generally, Kalman Filter is used after Hough Transform for tracking issues [5].

Niu *et al.* [2], [28] used a modified Hough transform to extract lane profile segments and used DBSCAN (Density-Based Spatial Clustering of applications with noise) clustering algorithm. Li *et al.* [29] introduced a new method of pre-processing and ROI selection using the HSV (hue, saturation, value) color transformation to extract the white features and add preliminary edge feature detection in the pre-processing stage and then select ROI on the basis of the proposed pre-processing. Li *et al.* [30] proposed geometrical model fitting combined with feature extraction and tracking to deal with low-speed environments. However, these algorithms have not been fully implemented in the efficiency and accuracy of lane detection and do not work properly at night.

Once lane detection and tracking are completed, lane departure warning system alerts the driver for unintended lane departure. However, in the most proposed methods, lane departure can be determined knowing lateral offset [6], [18], [23], vanishing point position [17] and the optical center of the camera [9]. The lateral offset between the primary vehicle's location and its position in various images has to be tracked in real-time. The vanishing point position is the most common solution used to determine lane departure [31]. In order to find the vehicle's horizontal displacement, vanishing point position needs to be compared with those of previous frames, but this process is time consuming. Finally, lane departure is determined based on the center of a road lane and the location of the optical center of the camera.

In order to deal with the above-mentioned limitations, in this paper, an adaptive ROI is proposed that uses a horizon line to effectively reduce the computational complexity and remove noisy segments. Moreover, a modified Hough Transform technique (PPHT) is used to enhance memory efficiency and to improve the computational time [4], [10], [32]. In addition, an adaptive threshold is is employed for binary images segmentation [33]. Lastly, we apply a trust Otsu method for edge detection to cope with the lighting problems mentioned in [27].

III. PROPOSED METHOD

The proposed method is composed of three main steps: pre-processing, adaptive ROI setting and lane markings detection and tracking using PPHT. The flowchart of the proposed method is shown in Fig. 1.

A. PRE-PROCESSING

In several cases, images contain noises due to not suitable conditions: low light, too much clarity or poor weather conditions (rain, fog, etc.). Noises significantly affect the visual quality of images, and accordingly the performance of most image processing tasks. Therefore, a useful pre-processing step is used to recover lost image information and to enhance image details. In addition, smoothing is part of pre-processing techniques intended for removing noises without losing image information. On the other hand, sharpening is used to improve image visual appearance and to highlight or recover certain details of the image. However, sharpening method over a noisy image may further amplifies the noise present in it [34].

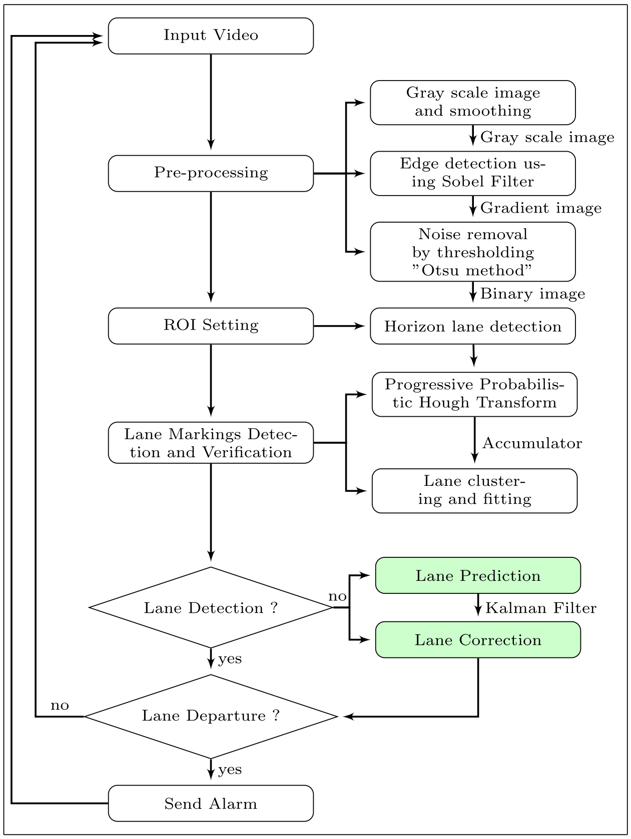

Input Video
Gray scale image and smoothing
Gray scale image
Pre-processing
Edge detection using Sobel Filter
Gradient image
Noise removal by thresholding "Otsu method"
Binary image
ROI Setting
Horizon lane detection
Progressive Probabilistic Hough Transform
Accumulator
Lane Markings Detection and Verification
Lane clustering and fitting
Lane Prediction
Kalman Filter
Lane Correction
Lane Detection ?
no
yes
no
Lane Departure ?
yes
Send Alarm

FIGURE 1. Flowchart of the proposed method.

Images captured in hazy or foggy weather conditions can be seriously degraded, thus reduces color contrast. A large number of image dehazing methods have been proposed, but often require a long processing time. Tan [35] proposed an effective image dehazing method based on two prior conditions. The first condition is that the contrast in the image without fog should be higher than that of the foggy image. The second condition is that the attenuation of field spots is a continuous function of distance that should be smooth. Fattal [36] presented a method for estimating the transmission in hazy scenes taking into consideration that the medium transmission function and the surface shading are locally and statistically uncorrelated. He *et al.* [37], [38] proposed a Dark Channel Prior (DCP) algorithm that can effectively overcome the deficiencies of the above two algorithms (proposed in [35] and [36]) to some extent. The

TABLE 1. Edge detection operators comparison.

	TPR (%)	FPR (%)	Processing Time
Sobel	91	7	19 ms
Canny	82	64	52 ms
Prewitt	88	61	48 ms

dark channel principle is sourced from a remote sensing image and an underwater image is used to summarize the rules from a natural image with no fog. Dong *et al.* [39] proposed an efficient traffic video dehazing using adaptive dark channel prior and spatial-temporal correlations method to speed up the dehazing process proposed in [35]–[37]. The method can restore the video with a resolution of 640 × 480 at about 57 frames per second (17,54ms/frame), which is too slow to be used in real-time scenarios. Hence, exploiting image dehazing methods with universality, robustness and real-time performance in lane detection and tracking will be a challenging task in the future [40].

Therefore, a useful smoothing step based on median [41] or Gaussian [42] filter is used. The objective is to enhance the contrast and to reduce the noise followed by an RGB to gray scale image in order to reduce the processing time and segmentation to binary images [24].

Once the grayscale image is created and smoothed, some classical segmentation operators such as Sobel, Canny and Prewitt filters can be used to detect edge information. In order to choose the better operator, we must first know which edge detectors perform better in term of true/false positives detection rate and processing time. The performances of the three operators were evaluated based on the following three measures: True Positive Rate (*TPR*), False Positive Rate (*FPR*) and the time complexity [43]. *TPR* refers to the probability of correctly detected lane markings. *FPR* refers to the probability of falsely selecting a given object or contours such as vehicles, trees or electrical poles, as lane markings. *TPR* and *FPR* are computed by using (1).

$$TPR = \frac{TP}{(TP + FN)}\%$$
$$FPR = \frac{FP}{(FP + TN)}\% \tag{1}$$

where *TP* is the number of true positive detections, *FN* is the number of false negative detections, *FP* is the number of false positive detections and *TN* is the number of true negative detections. The three operators were tested using 450 frames, and 640 × 360 input image size. After segmentation, we have used PPHT to detect lane markings, without other refinement stages and results are shown in Tab. 1

The Sobel operator gives 91% TPR, 7% FPR and 19ms processing time, The Canny operator gives 82% TPR, 64% FPR and 52ms processing time, and the Prewitt operator gives 88% TPR, 61% FPR and 48ms processing time. In terms of the TPR, the three operators are quite similar. However, in terms of the FPR, Sobel edge detector outperforms the other operators. In fact, Canny and Prewitt filters are very sensitive to irrelevant objects as well as lane markings, which rapidly increases the number of false positives. The Canny and Prewitt operators are also computationally expensive compared to Sobel edge detection algorithm [44]. Conversely, the Sobel operator is less sensitive to noise and less complex than Canny and Prewitt operators, and it is able to detect the main markings [4]. Hence Sobel operator is suitable to extract road lane information in real-time and to improve system performance under different lighting conditions.

To deal with the lighting problems and to efficiently segment binary images, Otsu method is used to reduce computation time and to remove noise [27]. The Otsu process searches for the threshold that minimizes the intra-class variance to obtain a binary image [45]. Therefore, pixels are split into two classes C_1 and C_0: maximum level (typically 255) and minimum level (typically 0) as background and foreground respectively. The average value in a binary image is obtained using formula (2).

$$\mu(t) = \omega_0(t)\mu_0(t) + \omega_1(t)\mu_1(t) \tag{2}$$

where μ_i are the average gray value of the background and the foreground, ω_i are the existing probabilities of the classes C_0 and C_1 obtained by the relative number of pixels by the threshold T. The value of T is set using formula (3).

$$T = \sum_{i=0}^{L-1} \mu_i p(\mu_i) \tag{3}$$

where L are the gray levels, $p(\mu_i) = n_i/N$ is the probability of occurrence of gray scale μ_i. n_i are the pixels with grayscale μ_i, and N is the total number of pixels. Image segmentation is performed when the threshold T is obtained.

B. HORIZON LINE DETECTION AND AROI SETTING

Given that the on-board camera was installed on the top of the windshield, input images contain useless background information, such as the sky, trees and electrical poles, on both sides of the road. After edge detection, this substantial redundant information still remains in the image. In order to lower image redundancy and reduce algorithm complexity, we can set an Adaptive Area of Interest (AROI) on the image. We only set the input image to the AROI area and this should increase the accuracy of the system and shorten the consumption time of the subsequent image processing steps. In this paper, we establish an AROI using a horizon line that effectively removes noisy line segments and reduces the computational complexity. The road region pixels usually show more intensity than sky region pixels. Hence, we use the horizon line to divide the scene into road region and sky region. This step has two main advantages: it reduces the effect of lighting conditions, and the influence of the sky region (artificial light at night-time, blue sky in daytime), as well as reducing the computation time. In this work we use the Vertical Mean Distribution (VDM) method proposed in [46] to identify the horizon line. The VMD is determined

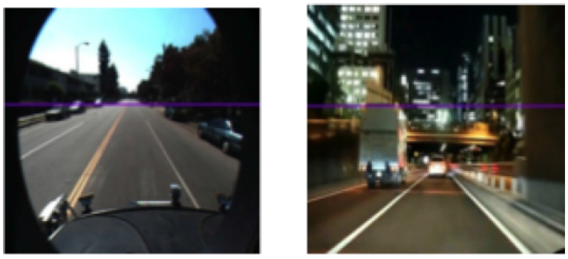

(a) (b)

FIGURE 2. Detected horizon line using VMD technique.

by using (4).

$$VMD(i) = \frac{1}{W}\sum_{j=1}^{W} I_G(i,j) \in (1,H) \tag{4}$$

where $VMD(i)$ is the average gray of each row. W and H stands for the width and the height of the original image respectively. $I_G(i,j)$ are the pixels intensity obtained by Otsu threshold process. The illustration of the AROI setting for road region is presented as follows:

$$I_{AROI(x,y)} = \begin{cases} I_G(x,y) & \text{if } y \leq y_{hl} \\ 0 & \text{otherwise} \end{cases}$$

where $I_{AROI(x,y)}$ and y_{hl} represent edge pixel in the selected AROI, and a vertical coordinate of the horizon line respectively.

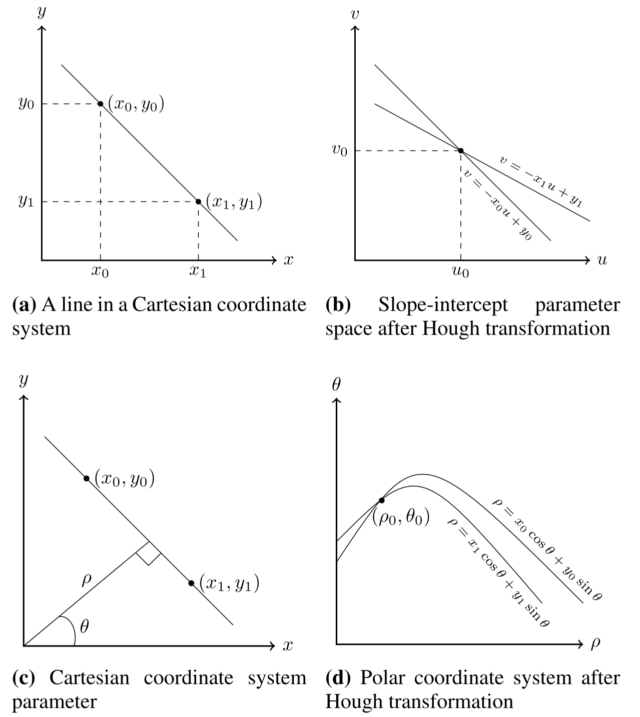
Fig. 2 shows the obtained horizon line (Violet line) under various lighting conditions. Fig. 2a and Fig. 2b show an example of AROI setting results based on the VMD technique applied on images from the Caltech dataset in urban road at day and night time respectively. This method is efficient and more convenient compared to vanishing point detection due to the reduced computation time.

C. LANE MARKINGS DETECTION AND TRACKING

PPHT is used to extract the straightest lines in the AROI. It allows to group edge features into an appropriate set of segments. This process is followed by a classical K-means clustering algorithm to fix the road lane. Therefore, we need to choose two lines (right and left side) only from the lines provided by PPHT with specific angles and lengths.

1) PROGRESSIVE PROBABILISTIC HOUGH TRANSFORM

Standard Hough transform is designed to detect straight lines [47], but with some modification it can be used to detect other arbitrary shapes. The basic principles of Hough Transform are shown in Fig. 3. Each point of the straight line crossing the point (x_0, y_0) and the point (x_1, y_1) (Fig. 3a) corresponds to a straight line $v = -x_0\,u + y_0$ and a straight line $v = -x_1\,u + y_1$ on the (u, v) parameter space after Hough Transformation (Fig. 3b). These two lines intersect at the point (u_0, v_0), where u_0 and v_0 are the parameters of the line determined by the points (x_0, y_0) and (x_1, y_1) respectively. On the other hand, the straight lines $v = -x_0\,u + y_0$ and $v = -x_1\,u + y_1$ where the parameter space (u, v) intersects at the same point and the collinear points in parameter space (x, y) are corresponding [48]. According to this characteristic, given some specific points in Fig. 3a, the line equations connecting these points in Fig. 3b can be calculated by using Hough transform.



(a) A line in a Cartesian coordinate system

(b) Slope-intercept parameter space after Hough transformation

(c) Cartesian coordinate system parameter

(d) Polar coordinate system after Hough transformation

FIGURE 3. Mapping from Cartesian space (x, y) (a) to the slope-intercept parameter space (u, v) (b) and from Cartesian space (x, y) (c) to the polar parameter space (ρ, θ) (d).

Duda *et al.* [49] propose that straight lines can be parameterized by (ρ, θ). The mapping relations between image point (x, y) and (ρ, θ) parameter space (Fig. 3c and Fig. 3d respectively) satisfy the following:

$$\rho = x\cos\theta + y\sin\theta$$

where (x, y) are coordinates of nonzero pixels in binary image, ρ is the distance between the x-axis and the fitted line, and θ is the angle between x-axis and the normal line.

As shown in Fig. 3, the Hough transform transforms the points of the image from the Cartesian parameter space (Fig. 3c) into the polar coordinate parameter space (Fig. 3d). The collinear point (x_0, y_0) and point (x_1, y_1) in Fig. 3c intersect at the same point (ρ_0, θ_0) (Fig. 3d). Here, ρ and θ are the polar parameters of the desired straight line [42].

Hough Transform employs a voting procedure where all edge pixels vote to identify a certain class of shapes in the image. The algorithm for detecting straight lines can be divided into the following steps:

- Edge detection, e.g. using the Sobel edge detector.
- Mapping of edge points to the Hough space and store in an accumulator.

- Interpretation of the accumulator to yield lines of infinite length. The interpretation is done by thresholding and possibly other constraints.
- Conversion of infinite lines to finite lines.

SHT provides inaccurate results in curved lane detection [4], [6], [13]. In fact, it takes into account all edge pixels, which increase the computational complexity and generates undesired markings candidates. On the other hand, PPHT is employed to efficiently detect both straight and curved lanes [5], [32], [50]. As developed in [50], PPHT algorithm uses only a small amount of randomly selected edge pixels compared to SHT. In particular, the PPHT improves the process of SHT by minimizing the number of voting pixels. It is initialized by a subset of randomly selected pixels, then, SHT is performed on the subset. PPHT accumulates only a small fraction of pixels as candidates of the lane markings, which improves the computation time. In this work, PPHT usually returns 5 to 15 lines while SHT returns 30 to 50 lines. The PPHT algorithm proceeds as follows:

1) Randomly, select a new point from the input image for voting in the accumulator array, with contributions to all available pair of (ρ, θ). Then update the accumulator and remove the selected pixel from the input image.
2) Check if the highest peak (the pair of(ρ, θ) with the most voting points) in the updated accumulator is greater than a predefined threshold T(N). If not return to Step1.
3) Choose the longest segment (which can be denoted by starting point Pt_0 and ending point Pt_1 of all lines.
4) Remove all points of the longest line from the input image.
5) Remove all points of the selected line in Step3 ($Pt_0 - Pt_1$) from the accumulator, which means those points are not part of any other voting process.
6) If the selected segment is longer than a predefined minimum length, then take the segment ($Pt_0 - Pt_1$) as one of the output results.
7) Return to Step1.

The flowchart of the PPHT is shown in Fig. 4. A Lane is extracted by identifying the cells in the PPHT accumulator that appears in the AROI. Therefore, the orientation angle threshold (θ) of edge pixels is used to determine the left and right lane markings as illustrated in (5) and (6).

$$P_L = (x, y)I_{G(AROI(x,y))} = 255, \quad x \leq w/2, \quad \theta_L(x, y) \in [15^\circ, 70^\circ] \tag{5}$$

$$P_R = (x, y)I_{G(AROI(x,y))} = 255, \quad x \leq w/2, \quad \theta_R(x, y) \in [-15^\circ, -70^\circ] \tag{6}$$

where P_L and P_R are the set of all possible edge pixels (x, y) of the left and right lane markings respectively, w is the width of the AROI and θ is the orientation angle of the pixel (x, y). The orientation angle of the left and right lane markings are $\theta_L \in [15^\circ, 70^\circ]$ and $\theta_R \in [-15^\circ, -70^\circ]$ respectively, as shown in (5) and (6). PPHT processes only lines with qualified length and removes undesired lines as shown in Fig. 5a. In fact, lines with an orientation angle that exceed the pre-defined threshold are considered as noise using (7).

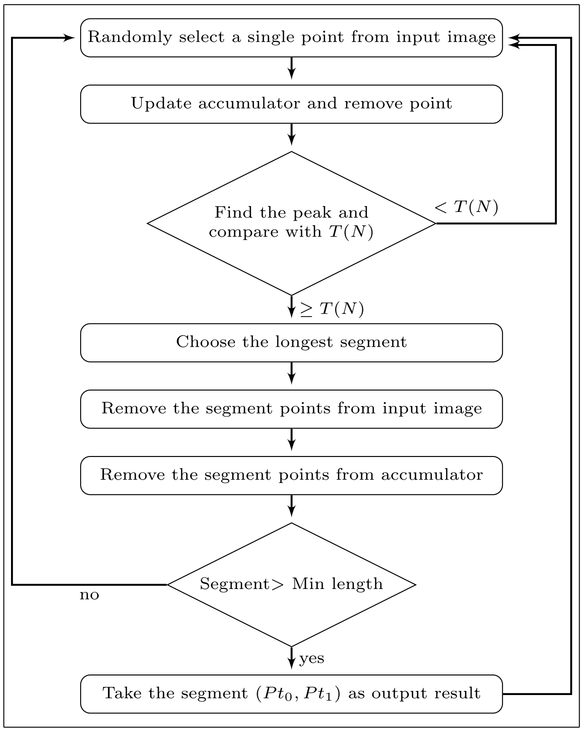

Randomly select a single point from input image
Update accumulator and remove point
Find the peak and compare with $T(N)$
$< T(N)$
$\geq T(N)$
Choose the longest segment
Remove the segment points from input image
Remove the segment points from accumulator
Segment> Min length
no
yes
Take the segment (Pt_0, Pt_1) as output result

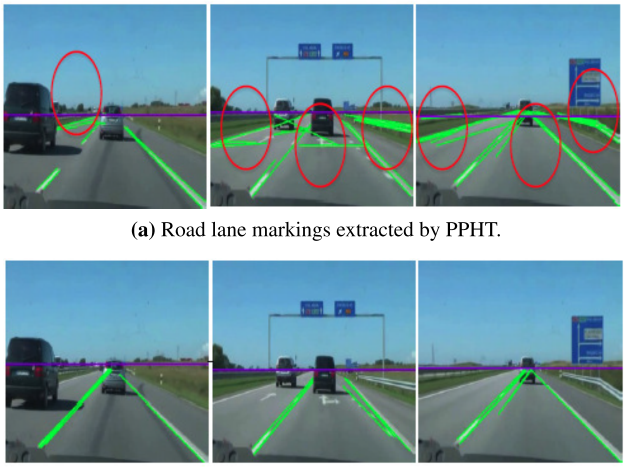
FIGURE 4. Flowchart of the progressive probabilistic hough transform.

$$T_0 \leq \theta_i = \arctan\left(\left|\frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}\right|\right) \leq T_1 \tag{7}$$

where T_0 and T_1 are the pre-defined angle limits, (x_i, y_i) and $(x_{(i+1)}, y_{(i+1)})$ are the coordinates of end-points (Pt_0, Pt_1). Experimentally, we used images of 640 × 360 pixels, the pre-defined threshold for the left lane markings is set to 0.25 for T_0 and 0.80 for T_1 (i.e., the angle between the lines and the car orientation is approximately set between 15 and 60 degrees). The pre-defined threshold for the right lane markings is set to 1.10 for T_0 and 1.19 for T_1 (i.e., the angle between the lines and the car orientation is approximately set between 115 and 145 degrees). Fig. 5a and Fig. 5b show the experimental results of edge detection and noise cancellation using PPHT and orientation angles.

2) K-MEANS CLUSTERING

PPHT does not take into account pre-defined angle thresholds along with the number of voting pixels in the accumulator. This can lead to unwanted lines for various reasons such as insufficient amount of marking lines, blurred images and passing vehicles. These noisy lines with unqualified angles can be removed using K-means clustering technique [51] as shown in Fig. 7a. K-means clustering algorithm is used to classify edge pixels into two groups: left and right lane. Zhao *et al.* [51] proposed a conventional cluster algorithm to localize the lane lines using a K-means clustering algorithm.



(a) Road lane markings extracted by PPHT.

(b) Elimination of line segments affected by shadows, noise and traffic signs on the road scene.

FIGURE 5. Edge detection and noise cancellation using PPHT and orientation angles.

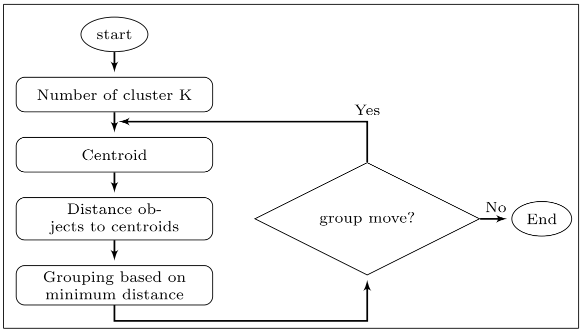

start
Number of cluster K
Yes
Centroid
Distance objects to centroids
group move?
No
End
Grouping based on minimum distance

FIGURE 6. Flowchart of the used K-means cluster algorithm.

In edge pixels, the set of lane markings S is defined by (8).

$$S = \{S_k \,|(P(x_{ks}, y_{ks}), P(x_{ke}, y_{ke}))\}, \quad k = 1, 2, 3, \ldots \tag{8}$$

where K is the total number of segments in S, $P(x_{ks}, y_{ks})$ and $P(x_{ke}, y_{ke})$ are the first and the last pixels of the K^{th} edge respectively. K-means cluster algorithm flowchart is as illustrated in Fig. 6.

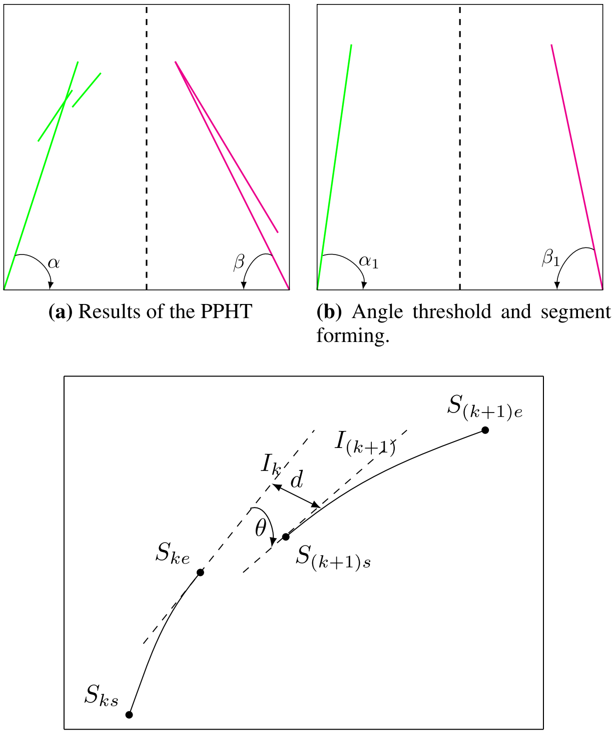
Let us choose the number of clusters as $K = 2$. Two edge pixels are selected as initial cluster centers of each lane markings. The remaining edge pixels are assigned to their closest cluster center according to the orientation equation (9).

$$\begin{aligned} min\ L(d, \theta) = d\ &(P(x_{ke}, y_{ke}), \angle P(x_{ke}, y_{ke})) \\ &+ \theta \left(P(x_{ke}, y_{ke}), P(x_{(k+1)s}, y_{(k+1)s})\right) \end{aligned} \tag{9}$$

With

$$0 \leq d\ (P(x_{ke}, y_{ke}), \angle P(x_{ke}, y_{ke})) \leq \lambda$$
$$0 \leq \theta \left(P(x_{ke}, y_{ke}), P(x_{(k+1)s}, y_{(k+1)s})\right) \leq \lambda$$

where d is the margin between the point $S_{(k+1)s}$ and the tangent I_k, θ is the difference of angles between two arbitrary lines I_k and $I_{(k+1)}$, and λ is a constant, as shown in Fig. 7c.



(a) Results of the PPHT

(b) Angle threshold and segment forming.

(c) Final road lane markings

FIGURE 7. Road lane detection steps.

In order to reduce the number of unnecessary line segments in the adaptive ROI, an arbitrary pair of line segments is selected from the linking process. Adequate lane markings are tracked using the classical K-means clustering technique. For this purpose, two-lane markings are selected from the set of lines obtained by PPHT with adequate lengths and angles as shown in Fig. 7b. Experimental results show that this method provides the pair of lines with the highest lane candidates probability for the next step.

D. LANE BOUNDARY ESTIMATION USING KALMAN FILTER

Lane tracking increases the probability to detect lane markings in poor conditions. Intensive research studies using Kalman Filter (KF) [5], [14], [26] or Particle Filter (PF) [51] have been done. In order to increase the fidelity and the efficiency of lane detection system, KF is used to track both limits of each lane markings extracted by PPHT. Based on previous state and current measurements, KF predicts the post-state with updating the covariance matrix. According to [14], [52], the relation between the state of a model at time k and its state at time $(k - 1)$ can be expressed by (10).

$$x_k = F_k x_{(k-1)} + B_k u_k + W_k \tag{10}$$

where F_k is the state transition matrix to be applied to the previous state vector $x_{(k-1)}$; B_k is the control matrix to update the external control vector u_k; W_k is the process noise with a covariance of Q_k given by (11).

$$W_k \sim N(0, Q_k) \tag{11}$$

In our proposed method, the process noise W_k resulting from lane detection can be considered as Gaussian distribution, which makes the lane tracking based on Gaussian stochastic process. We use two KF for the two side lane markings given by PPHT, with $Pt_1(X_{Pt1}, Y_{Pt1})$ and $Pt_0(X_{Pt0}, Y_{Pt0})$ are respectively the start point and the end point of a given line segment, respectively. The state vector x_k for our system for one single lane boundary K at time t is defined by (12).

$$x_k(t) = \left[X_{pt0}Y_{pt0}X_{pt1}Y_{pt1}X'_{pt0}Y'_{pt0}X'_{pt1}Y'_{pt1}\right]^T \tag{12}$$

where $X'$and Y' are the derivative forms of X and Y representing the lateral speed of the lane boundary movement.

Experimentally, the best value that satisfies tracking is 0.2, the state transition model $F_k \in R^{8\times 8}$ used in our system is given by (13).

$$F_k = \begin{pmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{13}$$

With

$$B_k = 0,\ u_k = 0,\ \Delta t = 0.2$$

The control matrix B_k and the control vector u_k in (10) will not be taken into account, because there is no input from external control in our proposed system.

E. LANE DEPARTURE WARNING SYSTEM (LDWS)

Lane departure warning is set based on the angle values between the middle horizontal axis and the road lane boundaries. Let C be the lane width, c the departure condition and α is the position of the middle lane relative to the axis X_c of the vehicle. The vehicle position with respect to the left and right lane markings is obtained using (14) [53].

$$\theta_l = \arctan\left(\frac{(C/2+c)}{\cos\alpha}\right),$$
$$\theta_r = \arctan\left(\frac{(C/2-c)}{\cos\alpha}\right) \tag{14}$$

where θ_l and θ_r are the angles of the left and right lane markings with respect to the horizontal axis.

Let us consider θ to be the sum of θ_l and θ_r (used to determine the vehicle's orientation) and T_d is the adequate threshold value. If θ is lower than $-T_d$, the vehicle drifts to the left, while if θ is higher than T_d, the vehicle drifts to the right, and a lane departure warning is activated.

IV. EXPERIMENTAL RESULTS

A. DATASET AND SETTING

Different datasets in various environments and lighting conditions were analyzed to evaluate the proposed method as shown in Tab.2. The vehicle light and street features as well as weather and time can make artificial light changes. On the other hand, the road nature influences the driving assistance system. In fact, normal roads are characterized by clear and straight lane markings, while urban roads are usually curved with high noise level markings. The system was subjected to various experimental settings representing realistic constraints. For example, traffic light, obstacles and pedestrians, as well as various roads types such as highways, local city roads, and tunnels. The publicly-available datasets; Caltech [7] and SLD 2012 [12], which contains complex road scenarios like intersections, bright spots and night were used in order to compare our method with others. Datasets images were captured with a resolution of 640 × 500, 640 × 480, 640 × 360, and 1280 × 800 pixels. The experiments were carried out on an Intel Core i7-2630QM CPU, 8GB RAM and Windows 7 64-bit operating system, using the software Opencv and C/C++ language.

TABLE 2. Environmental conditions.

Road type	Time	Different conditions
Datasets 1 (Highway, Local City Roads, and Tunnel)	Day, Sunrise, Sunset	Clear/ Rainy /Cloudy
	Night	Street lamps/Car lamps
	N/D	White and yellow lamps
Caltech dataset (Aly, 2008 [7])	Day Urban	Shadows and Curved Streets
SLD dataset (Borkar et al., 2009) [12]	Day Highway	Shadows and Curved Streets

B. EVALUATION CRITERIA

Three criteria are used in the evaluation of lane detection performance: Missed Lane Detection (*MLD*), Incorrect Lane Detection (*ILD*), and Detection Rate (*DR*). *MLD* refers to the percentage of falsely detected lane markings. *ILD* refers to the percentage of incorrect detected contours or objects such as trees or vehicles and *DR* is the detection rate. *MLD*, *ILD*, and *DR* are computed by using (15).

$$MLD = \frac{MD}{N} \times 100\%;\ ILD = \frac{ID}{N} \times 100\%;$$
$$DR = \frac{C}{N} \times 100\% \tag{15}$$

where *MD* is the missed detection, *ID* is the incorrect detection, *C* is the successfully detected images in the dataset, and *N* is the full dataset images.

1) LANE DETECTION RESULTS AND EVALUATION

1. For comparative evaluation, different experimentational settings (day and night time) where used for the three methods. Method (A) uses edge detection based on Sobel operator

TABLE 3. Comparative performance evaluation in (%) for lane markings detection at day and night time.

Performance (%)	Day			Night		
Methods	(A)	(B)	(C)	(A)	(B)	(D)
MLD	6.88	8.70	7.48	5.66	7.08	5.87
ILD	1.41	6.47	11.33	3.84	8.29	12.43
DR	91.70	84.81	81.17	90.48	84.61	81.87
Time (ms/frame)	22	27	36	20	24	33

and Otsu algorithm followed by PPHT and K-means clustering algorithms to detect road lane markings. Whereas Method (B) applies edge detection based on Sobel operator without Otsu algorithm, followed by PPHT and K-means clustering algorithms. Finally, Method (C) applies edge detection based on Canny operator followed by PPHT and K-means clustering algorithms. Lane markings detection results for 494 frames from the datasets in day and night time are summarized in Tab.3.

Experimental results show that Sobel-based edge detection with Otsu algorithm (method A) overcomes Sobel-based edge detection (method B) and Canny-based edge detection (method C). The Canny and Sobel operators suffer from high computational time complexity and higher ILD. The detection rate increases to 91.70% (day) and 90.48% (night) for the method (A), which decreases the *ILD* to 1.41% (day) and 3.84% (night). On the other hand, the detection rate decreases to 81.17 % applying method C, which increases the *ILD* to 11.33% (day) and 12.43% (night). In fact, the objective of the Otsu algorithm is to cope with the lighting problems, shadow and noise. The incorrect detected lane markings percentage for method (B) using Sobel operator is 6.47% (day) and 8.29% (night) whereas the incorrect detected lane markings percentage for method (C) using Canny operator is 11.33% (day) and 12.43% (night).

The proposed algorithm is mainly composed of three steps (pre-processing, AROI detection and setting, lane detection and lane tracking), two quantitative indexes are used to evaluate the performance of each method (A, B and C): the detection rate (%) and the time complexity (ms). Fig.8 shows the execution times of the main modules of the proposed method. These times were measured on an Intel Core i7-2630QM CPU with 8GB RAM using only a single core. The proposed method requires a total execution time of 21.54 ms/frame. Pre-processing and lane detection steps take the longest time (7.42 ms and 10.3 ms in average respectively), while the time consumption of AROI setting and lane tracking steps is relatively shorter that is 1.46 ms and 2.36 ms in average, respectively. On the other hand, the proposed algorithm maintains the shortest time consumption at day and night for method A. In fact, the time consumed by the algorithm increases significantly when the Incorrect Lane Detection (ILD) increases as shown in Tab.3.

If time complexity and *ILD* reduction are our major concern, the Sobel-based edge detection with Otsu algorithm is the best choice for lane markings detection. The proposed technique was able to detect and accurately locate road lane markings despite the presence of noise on the road. Tab.4 summarizes the results of 7610 treated images from Caltech and SLD datasets under various lighting conditions. Road lane markings detection results with our proposed algorithm for many road image sequences in different lighting conditions are shown in Fig.9.

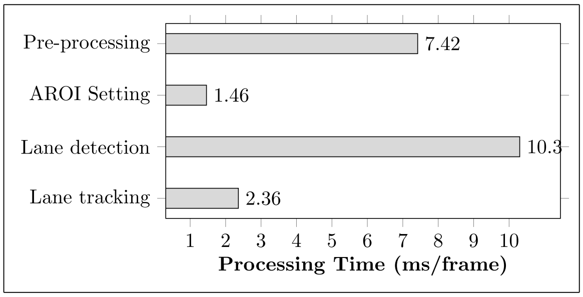


FIGURE 8. Processing time in (ms) of the proposed algorithm.

C. COMPARISON WITH OTHER METHODS

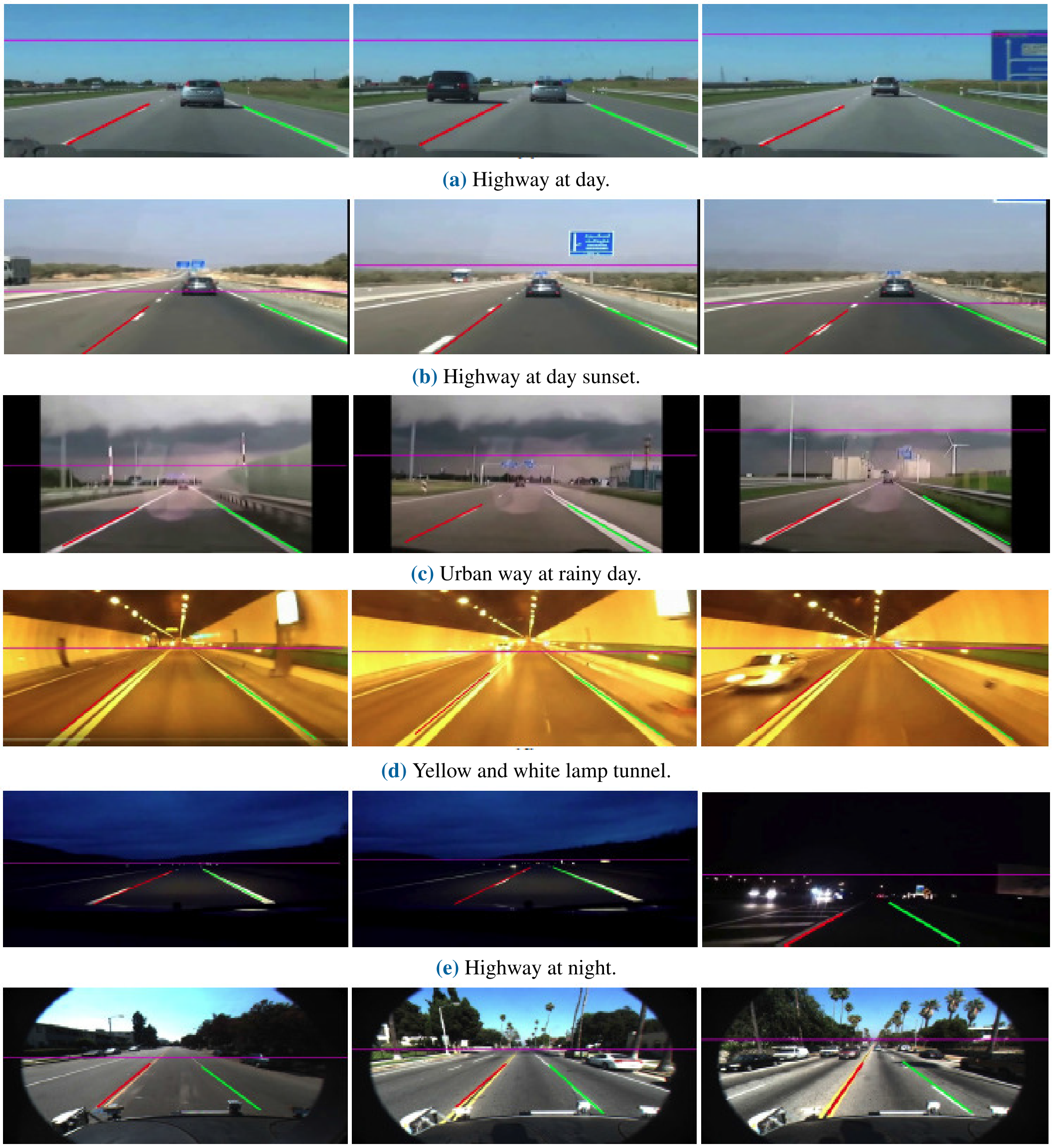
The comparative analysis was carried out in two-lane modes, in which the lane detection only applied to the two boundaries of a lane on Caltech dataset. All images have the size of 640 × 480 pixels. We compared our results with the methods described by Aly [7] in 2008; Yoo *et al.* [15] in 2013; Li *et al.* [30] in 2018; Hoang *et al.* [54] in 2016; Son *et al.* [6] in 2015; Hou *et al.* [55] in 2016; Shin *et al.* [19] in 2015; Guo *et al.* [56] in 2015; Liu *et al.* [57] in 2013 and Ruyi *et al.* [58] in 2011 under Caltech lanes dataset. Comparison results are summarized in Tab. 5. The main reason behind selecting these approaches to compare our to is that Aly [7] generated the public Caltech lanes dataset for evaluation and most researchers evaluated performance of their lane detection method using this dataset. On the other hand, the selected state-of-the-art lane detection methods were implemented in a highly optimized environment OpenCV and C++.

Son *et al.* [6] proposed method to extract lane markings using Canny operator and Hough Transform under different lighting conditions. Yoo *et al.* [15] use gradient-enhancing conversion method to deal with lighting changes and extract lane markings using an adaptive Canny operator, Hough transform, and curve model fitting to extract lane markings. Aly [7] use inverse perspective mapping (IPM) and RANSAC model fitting to extract lane markings. Li *et al.* [30] use symmetrical local threshold (SLT) and Bresenham line voting space (BLVS) to locate and detect the lane markings, while the Kalman filter is used to track the key points of the linear and curved parts of the lane. Hoang *et al.* [54] use the LSD method to make the proposed system robust to occlusion. Shin *et al.* [19] proposed the use of multiple particle filters for detecting left and right lane borders separately.

As shown in Tab. 5., the average lane detection rate of the proposed method outperforms the most of the state-of-the-art

TABLE 4. Lane markings detection results under different lighting conditions.

Performance (%)	Highway, Local City Roads, and Tunnel Datasets					Caltech dataset (Aly, 2008 [7])	SLD dataset (Borkar, 2009 [12])	Average
Illumination Conditions	Highway Day	Highway Day-sunset	Urban Day-Rainy	Tunnel	Highway Night	Urban Day	Highway Day	
Total frames (7610)	1351	867	847	1489	1440	480	1136	
MLD (%)	2.66	2.07	1.29	0	1.86	2.74	2.05	1.81
ILD (%)	7.17	4.38	8.5	1.41	2.93	6.27	5.13	5.11
DR (%)	90.15	93.54	90.20	98.85	96.4	91.6	92.7	93.34
Time/ (ms/frame)	22.7	20.03	23.65	19.52	19.6	20.8	21.4	21.54



(a) Highway at day.

(b) Highway at day sunset.

(c) Urban way at rainy day.

(d) Yellow and white lamp tunnel.

(e) Highway at night.

(f) Urban way at day.

FIGURE 9. Results of lane markings detection applied on images from Caltech dataset in different lighting conditions.

TABLE 5. The performance comparison of different lane detection methods under Caltech lanes dataset (Aly 2008 [7]).

Methods		Detection rate (%)	Time (ms/frame)
Aly et al. (2008) [7]		96.30	191
Yoo et al. (2013) [15]		96.37	50
Li et al. (2018) [31]		90.3	23.2
HOANG et al. (2016) [55]		90	33.09
Son et al. (2015) [6]		93.6	31.2
Hou et al. (2016) [56]		92.50	180
Shin et al. (2015) [19]		92.78	60.7
Guo et al. (2015) [57]		95.75	40
Liu and Li (2013) [58]		93.15	400
Ruyi et al. (2011) [59]		95.90	62
Proposed method	Highway at Day	90.15	22.7
	Highway at Day-sunset	93.54	20.57
	Urban way at Day-Rainy	90.20	23.65
	Tunnel	98.85	19
	Highway at Night	96.4	19.6
	Average	93.82	21.54

detection methods and is about 2% lower than some other methods. It is mostly due to state-of-the-art lane detection methods used sophisticated processing environment such as 2 to 4 cores higher processor clock speed and some methods like Shin *et al.* [19]; Ruyi *et al.* [58] applied tracking stage after detection stage for enhancing lane detection rate. The proposed method achieved comparable results in average lane detection rate at 93.82% with real-life datasets.

However, our method outperforms most of the state-of-the-art lane detection methods in computation time. The average computation time of the proposed method under Caltech lanes dataset is 21.54ms per frame. Some of state-of-the-art lane detection methods used top-view based approach for enhancing lane detection rate, which required intensive computing power and increase the processing time per frame. It is noted that real-life datasets contained smaller resolution frame images of 320 × 180 and resulted shorter average processing time per frame. Hence, our lane detection method with shorter processing time is an ideal for real-time implementation for lane detection applications. In the near future, we will try to test the proposed method using a progressive framework based on convolutional neural network (CNN) model [59]. Results can be improved using nonlinear descriptor systems with process disturbances and measurement output noises [60].

V. CONCLUSION

This paper addresses the problem of straight and curved lane detection and tracking under different lighting conditions through the proposed solution. In order to exhaustively exploit information on input images and to decrease computational complexity, horizon line and adaptive ROI are defined and used to ensure reliable real-time system. Progressive Probabilistic Hough Transform is defined as the main algorithm to extract boundary information is defined and used to detect road lane markings edges and to meet real-time system conditions. Otsu's threshold is also used to obtain gradients information, to deal with lighting problems and to enhance pre-processing results. We have also introduced K-means cluster process to locate the current lane markings. Moreover, Kalman filter is employed to track lane boundaries in the adaptive region of interest using PPHT. Based on lane boundaries, we can decide if the vehicle drifts off the lane using the vehicle's position. The proposed method ensures an accurate lane tracking; gather reliable information about the vehicle orientation to correctly indicates a lane departure. The system works properly under various lighting conditions, presents the best detection rate (93.82%) and reduced computation time (21.54ms/frame). However, the Sobel filter and PPHTs require much processing time in our proposed system. Our future work is therefore focused on the hardware/software implementation of the proposed road lane markings detection and tracking system using the Xilinx Zynq SoC in order to reduce the processing time and power.

REFERENCES

[1] J. Navarro, J. Deniel, E. Yousfi, C. Jallais, M. Bueno, and A. Fort, "Influence of lane departure warnings onset and reliability on car drivers' behaviors," *Appl. Ergonom.*, vol. 59, pp. 123–131, Mar. 2017.

[2] J. Piao and H. Shin, "Robust hypothesis generation method using binary blob analysis for multi-lane detection," *IET Image Process.*, vol. 11, no. 12, pp. 1210–1218, Dec. 2017.

[3] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, "Overview of environment perception for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2584–2601, Oct. 2017.

[4] Y. Kortli, M. Marzougui, B. Bouallegue, J. S. C. Bose, P. Rodrigues, and M. Atri, "A novel illumination-invariant lane detection system," in *Proc. 2nd Int. Conf. Anti-Cyber Crimes (ICACC)*. Abha, Saudi Arabia: King Khalid University, Mar. 2017, pp. 166–171.

[5] A. Mammeri, A. Boukerche, and G. Lu, "Lane detection and tracking system based on the MSER algorithm, Hough transform and Kalman filter," in *Proc. 17th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst. (MSWiM)*, New York, NY, USA, 2014, pp. 259–266.

[6] J. Son, H. Yoo, S. Kim, and K. Sohn, "Real-time illumination invariant lane detection for lane departure warning system," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1816–1824, Mar. 2015.

[7] M. Aly, "Real time detection of lane markers in urban streets," in *Proc. IEEE Intell. Vehicles Symp.* Eindhoven, The Netherlands: Eindhoven University of Technology, Jun. 2008, pp. 7–12.

[8] H. Tan, Y. Zhou, Y. Zhu, D. Yao, and K. Li, "A novel curve lane detection based on improved river flow and RANSA," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, Oct. 2014, pp. 133–138.

[9] Q. Chen and H. Wang, "A real-time lane detection algorithm based on a hyperbola-pair model," in *Proc. IEEE Intell. Vehicles Symp.*, Tokyo, Japan, Jun. 2006, pp. 510–515.

[10] R. N. Hota, S. Syed, S. Bandyopadhyay, and P. Krishna, "A simple and efficient lane detection using clustering and weighted regression," in *Proc. 5th Int. Conf. Manage. Data COMAD*. Mysore, India: Computer Society of India, Dec. 2009, pp. 1–8.

[11] Y. Wang, N. Dahnoun, and A. Achim, "A novel system for robust lane detection and tracking," *Signal Process.*, vol. 92, no. 2, pp. 319–334, Feb. 2012.

[12] A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with ransac and Kalman filter," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Cairo, Egypt, Nov. 2009, pp. 3261–3264.

[13] M. B. de Paula and C. R. Jung, "Automatic detection and classification of road lane markings using onboard vehicular cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3160–3169, Dec. 2015.

[14] Y. Li, A. Iqbal, and N. R. Gans, "Multiple lane boundary detection using a combination of low-level image features," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1682–1687.

[15] H. Yoo, U. Yang, and K. Sohn, "Gradient-enhancing conversion for illumination-robust lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1083–1094, Sep. 2013.

[16] W. Ding, Y. Li, and H. Liu, "Efficient vanishing point detection method in unstructured road environments based on dark channel prior," *IET Comput. Vis.*, vol. 10, no. 8, pp. 852–860, Dec. 2016.

[17] P.-Y. Hsiao, C.-W. Yeh, S.-S. Huang, and L.-C. Fu, "A portable vision-based real-time lane departure warning system: Day and night," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 2089–2094, May 2009.

[18] Y. Wang, L. Bai, and M. Fairhurst, "Robust road modeling and tracking using condensation," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 4, pp. 570–579, Dec. 2008.

[19] B.-S. Shin, J. Tao, and R. Klette, "A superparticle filter for lane detection," *Pattern Recognit.*, vol. 48, no. 11, pp. 3333–3345, Nov. 2015.

[20] S. Jung, J. Youn, and S. Sull, "Efficient lane detection based on spatiotemporal images," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 289–295, Jan. 2016.

[21] D. Hanwell and M. Mirmehdi, "Detection of lane departure on high-speed roads," in *Proc. 1st Int. Conf. Pattern Recognit. Appl. Methods (ICPRAM)*, vol. 2, 2012, pp. 529–536.

[22] Z. Nan, P. Wei, L. Xu, and N. Zheng, "Efficient lane boundary detection with spatial-temporal knowledge filtering," *Sensors*, vol. 16, no. 8, p. 1276, 2016.

[23] K. Zhao, M. Meuter, C. Nunn, D. Müller, S. Muller-Schneiders, and J. Pauli, "A novel multi-lane detection and tracking system," in *Proc. IEEE Intell. Vehicles Symp.*, Alcala de Henares, Spain, Jun. 2012, pp. 1084–1089.

[24] P. V. Ingale and K. S. Bhaga, "Comparative study of lane detection techniques," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 4, no. 5, pp. 381–390, 2016.

[25] L. Zefeng, X. Ying, S. Xin, L. Licai, W. Xingzheng, and S. Jianhao, "A lane detection method based on a ridge detector and regional G-RANSAC," *Sensors*, vol. 19, no. 18, p. 4028, 2019.

[26] S. Sivaraman and M. M. Trivedi, "Integrated lane and vehicle detection, localization, and tracking: A synergistic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 906–917, Jun. 2013.

[27] Q. Li, L. Chen, M. Li, S.-L. Shaw, and A. Nuchter, "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 540–555, Feb. 2014.

[28] J. Niu, J. Lu, M. Xu, P. Lv, and X. Zhao, "Robust lane detection using two-stage feature extraction with curve fitting," *Pattern Recognit.*, vol. 59, pp. 225–233, Nov. 2016.

[29] M. Li, Y. Li, and M. Jiang, "Lane detection based on connection of various feature extraction methods," *Adv. Multimedia*, vol. 2018, pp. 1687–5699, Aug. 2018.

[30] Q. Li, J. Zhou, B. Li, Y. Guo, and J. Xiao, "Robust lane-detection method for low-speed environments," *Sensors*, vol. 18, no. 12, p. 4274, 2018.

[31] M. Haloi and D. B. Jayagopi, "A robust lane detection and departure warning system," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Seoul, South Korea, Jun./Jul. 2015, pp. 126–131.

[32] Y.-W. Seo and R. R. Rajkumar, "Utilizing instantaneous driving direction for enhancing lane-marking detection," in *Proc. IEEE Intell. Vehicles Symp.*, Dearborn, MI, USA, Jun. 2014, pp. 170–175.

[33] C. Mu and X. Ma, "Lane detection based on object segmentation and piecewise fitting," *TELKOMNIKA Indonesian J. Electr. Eng.*, vol. 12, no. 5, pp. 3491–3500, 2014.

[34] C. Pérez-Benito, S. Morillas, C. Jordán, and J. A. Conejero, "Smoothing vs. sharpening of colour images: Together or separated," *Appl. Math. Nonlinear Sci.*, vol. 2, no. 1, pp. 299–316, Jun. 2017.

[35] R. T. Tan, "Visibility in bad weather from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[36] R. Fattal, "Single image dehazing," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–9, 2008.

[37] K. M. He, J. Sun, and X. O. Tang, "Single image haze removal using dark channel prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2009, pp. 1956–1963.

[38] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[39] T. Dong, G. Zhao, J. Wu, Y. Ye, and Y. Shen, "Efficient traffic video dehazing using adaptive dark channel prior and spatial–temporal correlations," *Sensors*, vol. 19, no. 7, pp. 1593–1612, 2019.

[40] W. Wang and X. Yuan, "Recent advances in image dehazing," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 3, pp. 410–436, Jul. 2017.

[41] M.-C. Chuang, J.-N. Hwang, and K. Williams, "A feature learning and object recognition framework for underwater fish images," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1862–1872, Feb. 2016.

[42] H. Xu and H. Li, "Study on a robust approach of lane departure warning algrithm," in *Proc. IEEE Int. Conf. Signal Process. Syst. (ICSPS)*, Jul. 2010, pp. 201–204.

[43] Z. Zhixiao, L. Penghui, H. Mengxia, Z. Wenhui, and L. Yibing, "Drivers' lane-changing maneuvers detection in highway," in *Proc. Int. Conf. Man-Mach.-Environ. Syst. Eng. (MMESE)*, vol. 406. Singapore: Springer, 2016, pp. 21–29.

[44] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*, 2nd ed. Beijing, China: Publishing House of Electronics Industry, 2002.

[45] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[46] K. H. Lim, K. P. Seng, L.-M. Ang, and S. W. Chin, "Lane detection and Kalman-based linear-parabolic lane tracking," in *Proc. Int. Conf. Intell. Hum.-Mach. Syst. Cybern.*, vol. 2, 2009, pp. 351–354.

[47] P. V. C. Hough, "Method and means for recognizing complex patterns," U.S. Patent 3 069 654, Dec. 18, 1962.

[48] J. W. Lee and U. K. Yi, "A lane-departure identification based on LBPE, Hough transform, and linear regression," *Comput. Vis. Image Understand.*, vol. 99, no. 3, pp. 359–383, Sep. 2005.

[49] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.

[50] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic Hough transform," *Comput. Vis. Image Understand.*, vol. 78, no. 1, pp. 119–137, Apr. 2000.

[51] H. Zhao, O. Kim, J.-S. Won, and D.-J. Kang, "Lane detection and tracking based on annealed particle filter," *Int. J. Control, Autom. Syst.*, vol. 12, no. 6, pp. 1303–1312, Dec. 2014.

[52] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, Mar. 2006.

[53] X. An, M. Wu, and H. He, "A novel approach to provide lane departure warning using only one forward-looking camera," in *Proc. Int. Symp. Collaborative Technol. Syst. (CTS)*, 2006, pp. 356–362.

[54] T. Hoang, H. Hong, H. Vokhidov, and K. Park, "Road lane detection by discriminating dashed and solid road lanes using a visible light camera sensor," *Sensors*, vol. 16, no. 8, p. 1313, 2016.

[55] C. Hou, J. Hou, and C. Yu, "An efficient lane markings detection and tracking method based on vanishing point constraints," in *Proc. 35th Chin. Control Conf. (CCC)*, Jul. 2016, pp. 6999–7004.

[56] J. Guo, Z. Wei, and D. Miao, "Lane detection method based on improved RANSAC algorithm," in *Proc. IEEE 12th Int. Symp. Auton. Decentralized Syst.*, Mar. 2015, pp. 285–288.

[57] W. Liu and S. Li, "An effective lane detection algorithm for structured road in urban," in *Intelligent Science and Intelligent Data Engineering* (Lecture Notes in Computer Science). Heidelberg, Germany: Springer, 2013, pp. 759–767.

[58] J. Ruyi, K. Reinhard, V. Tobi, and W. Shigang, "Lane detection and tracking using a new lane model and distance transform," *Int. J., Mach. Vis. Appl.*, vol. 22, no. 4, pp. 721–737, Jul. 2011.

[59] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Bian, and Y. Yang, "Progressive learning for person re-identification with one example," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2872–2881, Jun. 2019.

[60] Y. Wu, B. Jiang, and N. Lu, "A descriptor system approach for estimation of incipient faults with application to high-speed railway traction devices," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 10, pp. 2108–2118, Oct. 2019.

MEHREZ MARZOUGUI was born in Kasserine, Tunisia, in 1972. He received the B.Sc. degree in electronics from the University of Tunis, in 1996, and the M.Sc. and Ph.D. degrees in electronics from the University of Monastir, in 1998 and 2005, respectively. From 2001 to 2005, he was a Research Assistant with the Electronics and Micro-Electronics Laboratory. From 2006 to 2012, he has been an Assistant Professor with Electronics Department, University of Monastir. Since 2013, he has been an Assistant Professor with the Engineering Department, College of Computer Science, King Khalid University. He is the author of more than 30 articles. His research interests include hardware/software cosimulation, image processing, and multiprocessor system on chip (MPSoC).

AREEJ ALASIRY received the B.Sc. degree in information systems from King Khalid University, Abha, Saudi Arabia, and the M.Sc. degree (Hons.) in advanced information systems and the Ph.D. degree in computer science and information systems from the Birkbeck College, University of London, U.K., in 2010 and 2015, respectively. She is currently an Assistant Professor with the College of Computer Science, King Khalid University. She currently the College Vice Dean of Graduate Studies and Scientific Research. Her main research interests include machine learning and data science.

YASSIN KORTLI was born in Kasserine, Tunisia, in 1990. He received the B.Sc. degree in micro-electronics and the master's degree in micro-electronics and nano-electronics from the University of Monastir, in 2013 and 2015, respectively. He is currently pursuing the joint Ph.D. degree with the AI-ED Laboratory, University of Western Brittany, France, and the Laboratory of Electronics and Micro-Electronics, University of Monastir. He has published over ten refereed journal articles and conference papers. His research interests include circuit and system design, pattern recognition, and image and video processing.

JAMEL BAILI was born in Sousse, Tunisia, in 1976. He received the B.Sc., M.Sc., and Ph.D. degrees in electronics from the University of Monastir, in 2001, 2003, and 2009, respectively. From 2003 to 2013, he was a Research Assistant with the Micro-Electronics and Instrumentation Laboratory. From 2010 to 2013, he was an Assistant Professor with the Electronics Department, University of Sousse. Since 2013, he has been an Assistant Professor with Engineering Department, College of Computer Science, King Khalid University. His research interests include embedded systems, instrumentation, and digital signal processing.