J Real-Time Image Proc (2019) 16:1781–1794
DOI 10.1007/s11554-017-0687-2

ORIGINAL RESEARCH PAPER

# Real-time illumination and shadow invariant lane detection on mobile platform

Ayhan Küçükmanisa[1] · Gökhan Tarım[1] · Oğuzhan Urhan[1]

Received: 20 September 2016 / Accepted: 2 April 2017 / Published online: 9 April 2017

**Abstract** In this work, a novel lane detection method using a single input image is presented. The proposed method adopts a color and shadow invariant preprocessing stage including a feature region detection method called as maximally stable extremal regions. Next, candidate lane regions are examined according to their structural properties such as width–height ratio and orientation. This stage is followed by a template matching-based approach to decide final candidates for lane markings. At the final stage of the proposed method, outliers are eliminated using the random sample consensus approach. The proposed method is computationally lightweight, and thus, it is possible to execute it in real-time on consumer-grade mobile devices. Experimental results show that the proposed method is able to provide shadow, illumination and road defects invariant performance compared to the existing methods.

**Keywords** Advanced driver assistance systems · Lane detection · MSER · Shadow invariant · Illumination invariant

✉ Oğuzhan Urhan
urhano@kocaeli.edu.tr

Ayhan Küçükmanisa
ayhan.kucukmanisa@kocaeli.edu.tr

Gökhan Tarım
tarim.gokhan@gmail.com

[1] Electronics and Telecommunication Engineering Department, University of Kocaeli, Kocaeli, Turkey

## 1 Introduction

Nowadays the increasing number of vehicles on roads makes traffic safety more important. The need for driver assistance systems that contain reliable methods to inform drivers about their mistake for ensuring driver safety is increasing day by day. These kind of systems are called as Advanced Driver Assistance Systems (ADAS) in general, and they attract the attention of consumer electronics and automotive industry.

An important portion of the accidents originates from the uncontrolled lane change of vehicles due to driver fatigue. There are many ADAS to warn the driver and avoid possible collisions. One of the most commonly used ADAS for this purpose is called as lane departure warning system (LDWS) which aims to warn driver when the vehicle moves out of its lane.

A typical LDWS has lane detection and lane departures decision stages. Primarily, it is required to detect lane markings for deciding the lane departures. Lane detection is a challenging problem because of the illumination and road conditions. However, LDWS must reliable enough to detect the lane markings even in the case of serious artifacts originating from lighting conditions, environment effects and the road surface problems. There are many methods in the literature to detect lane departures as will be described in the following section. However, only a limited number of work focus on the real-time implementation of the LDWS. This paper directly aims efficient detection of lanes for lane departure detection on a mobile platform.

This paper is organized as follows. In Sect. 2, a survey of recent lane detection methods is presented. The proposed lane detection method is explained in Sect. 3. The experimental results and detailed performance evaluation of the proposed method in this work are discussed in

Sect. 4. In this section, mobile platform implementation of the proposed method is presented in Sect. 4 as well. Finally, Sect. 5 provides the conclusion.

## 2 Related work

Recently, many vision-based methods have been proposed in the literature for lane detection. Some of these methods presented in [1–3] employ stereo imaging sensor, whereas most of them as in [4–6] utilize only single imaging sensor. In general, lane detection methods consist of three main stages: preprocessing, feature extraction, and post-processing.

Preprocessing aims to improve lane detection accuracy in challenging scenarios like rainy and foggy weather conditions, poor or over-illumination, and shadow cases. In feature extraction stage, it is targeted to find lane marking candidates. After obtaining lane marking candidates, outliers are eliminated to obtain final lane marking positions at the post-processing stage.

Gaikwad et al. [4] proposed a PLSF (piecewise linear stretching function)-based method for preprocessing stage to enhance input image. Hough transform [5] is applied to PLSF improved image, and finally Euclidean-based decision approach is employed to decide the position lanes. The performance of this method directly depends on the PLSF parameters where it may not be possible to find a reliable set of parameters for different scenarios.

It is difficult to detect dashed lane markings compared to continuous lanes. As a solution to this problem, a temporal blurring approach is presented by Borkar et al. [6] where successive images at specific intervals are used to construct an average image with continuous lane markings from the images that may contain dashed lane markings. Additionally, this method employs IPM (inverse perspective mapping) as described by Bertozzi et al. [7] to take advantage of parallelism of the lane markings. After these preprocessing stages, Hough transform is applied to detect straight lines. Finally, one-dimensional (1-D) correlation is employed over the samples of region detected by Hough transform to obtain lane marking features. RANSAC as presented by Fischler et al. [8] is applied on the detected lane marking features to decide lane markings. Kalman filtering [9] is utilized to track detected lanes in this method.

The proposed method by Espinoza et al. [10] initially segments the area corresponding to the road employing mean shift clustering and Gabor textural features. Lane geometry and position relative to the vehicle in three-dimensional (3-D) coordinates is obtained using the segmented road region, steerable filter response, IPM, Hough transform, and MSAC (M-estimator Sample and Consensus) as described by Torr et al. [11]. The estimation of the lane geometry is improved by making use of extended Kalman filter. In this method, lane detection process is only applied to the road area obtained by road segmentation stage. This approach simply improves the lane detection accuracy and reduces computational load.

For illumination-robust lane detection, a linear discriminant analysis (LDA)-based gradient-enhancing conversion introduced by Yoo et al. [12]. This method aims to generate optimal RGB color channel weights that maximize gradients at lane boundaries on the road to distinguish lanes from roads. After the gradient-enhancing conversion, adaptive Canny edge detection, Hough transform, and curve model fitting are used one after another. This method is robust to illumination changes thanks to the conversion method presented.

In nighttime, it becomes challenging to detect lanes due to the relatively low lighting. The average gray value in images captured by camera in nighttime is quite lower compared to that of daytime. A novel algorithm for nighttime detection of the lane markings is introduced by You et al. [13]. This method employs neighborhood average filtering, 8-directional Sobel operator, and thresholding as proposed by Otsu [14] in the preprocessing stage. Next, distribution of pixel intensity values on boundaries of lanes candidates is analyzed. If the same behavior is observed for different directions, this position is kept as lane candidates while others are considered as noise. At the last step, the lane markings are obtained using conventional Hough transform.

Mammeri et al. [15] proposed MSER [16] and Hough transform-based lane detection method. For the details of MSER, the reader is referred to the [16]. In the method presented by Mammeri et al. [15], possible lane areas are detected by making use of MSER. Undesirable regions are eliminated by applying a three-stage refinement algorithm. Progressive Probabilistic Hough Transform (PPHT) as proposed by Stephens [17] is used to detect lane markings. Finally, the detected lane markings are tracked by making use of Kalman filter. The main drawback of this method originates from the scanning-based refinement algorithm. Although this method is able to achieve real-time processing performance, it results in many false lane detections.

A lane detection method which works real time on a DSP (digital signal processor) platform is presented by Küçükyıldız et al. [18]. At the first step, this method uses Sobel filter to detect vertical edges. Next, Hough transform is applied to the filtered image for detecting lines. At this stage, only the lines with a certain angle range are taken into consideration. Next, detected lines are examined pair-by-pair by taking all possible combination into account. A line is eliminated if it is not parallel to other lines and its distance to other lines is outside of the predefined range. After elimination, line pair with maximum Hough score is decided as lane markings. Finally, detected lane markings are tracked by Kalman filter.

Unlike image analysis-based methods, a convolutional neural network-based method is proposed in [19]. In this work, firstly input image is smoothed by making use of a 5 × 5 Gaussian kernel in order to minimize the effect of noise. Then, it is aimed to enhance the lane markings by taking the image convolution process with the kernel called the hat-shape kernel similar to the structure of the lane markings. A convolutional neural network model is created in this work, and the lane markings are detected by the RANSAC method over the feature map formed at the end of this network. Although the processing speed and evaluation criteria are not given, it is stated that the average lane detection performance is above 96%. A deep learning framework for mobile devices is recently presented in Tveit et al. [20]. Even though this framework provides 10 fps processing performance, this cannot be directly used for lane detection purpose since there many additional steps required to detect lanes. In general, deep learning-based methods are not suitable for low-power mobile devices for real-time processing.

A comparison of recent LDWS methods covered in this section is given in Table 1. In this table, the methods used in preprocessing, feature extraction, and post-processing stages together with the tracking, evaluation metrics used to determine performance of proposed methods with employed databases and processing speed on targeted platforms are given.

## 3 Proposed lane detection method

The block diagram of the proposed method is shown in Fig. 1. The proposed single input image-based lane detection method has three main stages. The first step is to obtain MSER from the enhanced B color channel of the input image. At the second step, some of the MSER are eliminated using a hybrid method which includes structural analysis and Gauss-based template matching. Finally, RANSAC approach is employed on the correlation coefficients obtained at previous step to detect lane markings by eliminating outliers.

### 3.1 Preprocessing

The preprocessing stage contains enhanced B channel image construction and MSER detection steps. The proposed lane detection method is performed on the selected ROI (region of interest) to achieve real-time lane detection performance.

Illumination and shadow effects may cause severe problems for lane detection systems by producing inconsistent results. MSER can be utilized to detect extremal regions such as pavement, vehicles, and lanes in road images. However, shadows and illumination changes might affect MSER detection results. In these cases, the lanes may not be detected or detected regions size can display significant deviation from the real region size. This behavior of MSER significantly reduces performance of the lane detection. In order to obtain better lane detection performance at harsh shadow and illumination variation situations, the proposed method investigates performance of MSER on different color space channels (e.g., RGB: red, blue, and green, HSV: hue, saturation, and value) and grayscale image. Figure 2 shows detected MSERs on different color space channels. In Fig. 2a, three lane regions are in the input image. These regions are used to evaluate MSER detection performance on different color channels. Note that different color codes simply represent different regions detected by MSER approach in this figure. As seen from Fig. 2f, nearly all three regions are detected as MSER when B channel is employed which makes this color channel suitable for lane marking candidate detection. Other color channels are not capable of providing reliable results by missing some lane markings.

The lane markings are generally painted using white or yellow color on the road surface. White lane markings can be easily detected by MSER-based approach using the B color channel of RGB color space even in the case of varying shadow and illumination. However, detection performance in the case of yellow lane markings is lower since the pure yellow color is generated by the maximum contribution of red and green channels. Thus, it is proposed to enhance B channel image to alleviate this drawback of MSER on B channel for the detection of lane markings. The enhanced B channel image is constructed as

$$B(x,y) = B(x,y) + ((|R(x,y) - B(x,y)|) \times (1 - \text{level})) \tag{1}$$

$$\text{level} = \text{OTSU}\left(\frac{|R(x,y) - B(x,y)|}{\max(|R(x,y) - B(x,y)|)}\right) \tag{2}$$

where $R(x,y)$ and $B(x,y)$ denote the red and blue channel of input images, whereas level shows the threshold level obtained by Otsu algorithm. The MSER detection results are shown in Fig. 3. It is seen from these results that the proposed enhanced B channel image approach improves the MSER detection results in the case of yellow lanes markings.

### 3.2 Feature extraction

After the MSERs are obtained in the preprocessing stage, feature extraction is carried out. The feature extraction approach utilized in this work is based on structural analysis of candidate regions and 1-D Gauss template matching.

**Table 1** Comparison of recent LDWS methods

| Method | Preprocessing | Feature extraction | Post-processing | Tracking | Performance evaluation metric | Platform/speed (fps) | Database |
|---|---|---|---|---|---|---|---|
| Borkar et al. [6] | Temporal blurring, adaptive thresholding, IPM | Hough transform, Gauss template matching | RANSAC | Kalman filter | Average distance between the ground truth and lane location | PC/1.25 | Their own dataset |
| Tapia-Espinoza et al. [10] | Mean shift clustering, Gabor and GMRF texture features | Steerable filters, IPM, Hough transform | Clothoid model equations, MSAC | Extended Kalman filter | RMSE, standard deviation of the error | PC/5.6 | Their own dataset (ground truth manually generated) |
| Yoo et al. [12] | LDA based gradient-enhancing conversion, adaptive Canny edge detection | Hough transform, edge-linking method | Clothoid curve model lane fitting | – | Visual evaluation (detection of lanes on real lane markings and bending directions are true) | PC/20 | Their own dataset |
| You et al. [13] | Neighborhood average filtering, 8-directional Sobel operator, OTSU thresholding, area of interest | Distribution of pixels (gradient, intensity), Hough transform | – | – | Not mentioned | PC/17 | Their own dataset |
| Küçükyıldız et al. [18] | Vertical 1-D Sobel filter, thresholding | Hough transform, IPM | Metric analysis (lane properties) | Kalman | Not mentioned | DSP/30 | Carnegie Mellon University dataset |
| Gaikwad et al. [4] | ROI, PLSF | Hough transform | Euclidean distance | – | Not mentioned | PC/28 | Their own dataset, LabelMe dataset |
| Mammeri et al. [15] | MSER | MSER refinement, scanning algorithm, PPHT | – | Kalman filter | Not mentioned | PC/24 | Their own dataset |

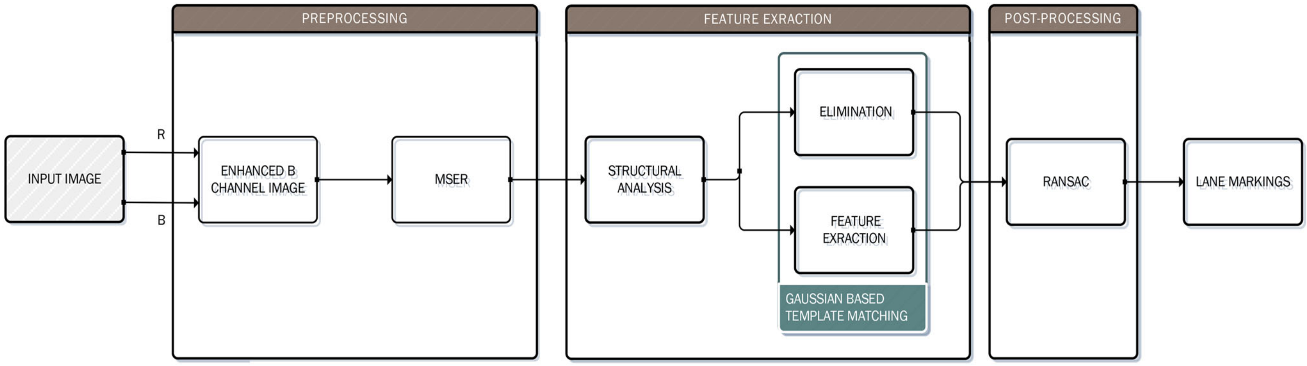
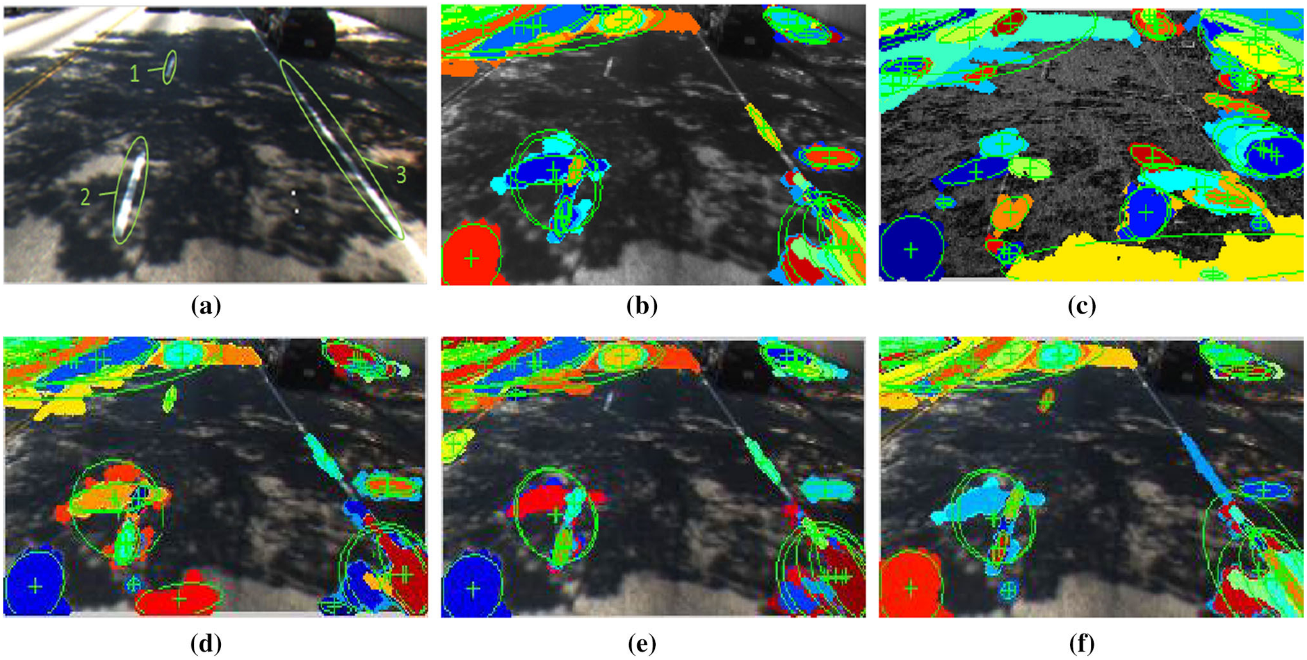


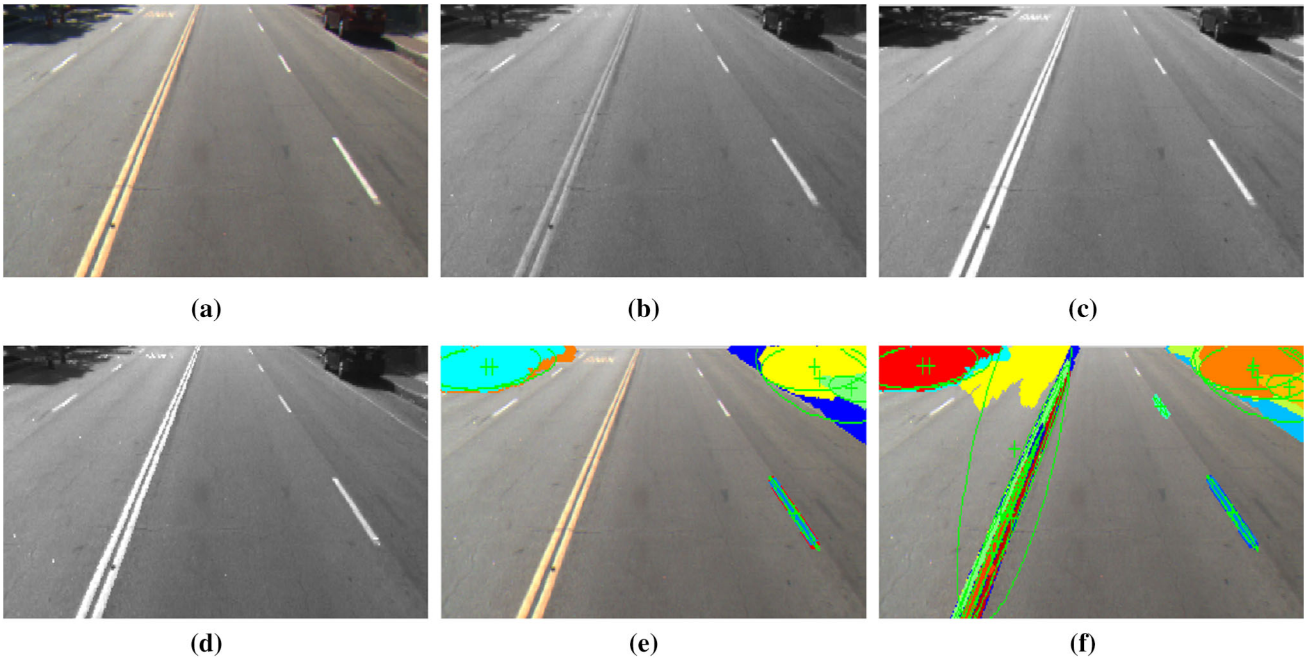
**Fig. 1** Flowchart of proposed method



(a) (b) (c)

(d) (e) (f)

**Fig. 2** MSER detection results in *different color* channels **a** original image, **b** MSER results on grayscale image, **c** MSER results on hue channel image, **d** MSER results on red channel image, **e** MSER results on green channel image, **f** MSER results on blue channel image



(a) (b) (c)

(d) (e) (f)

**Fig. 3** MSER detection results of enhanced B channel **a** original image, **b** blue channel of original input image, **c** red channel of original input image, **d** proposed enhanced B image, **e** MSER results on blue channel image, **f** MSER results on the proposed enhanced image

### *3.2.1 Structural analysis*

The MSERs have some particular characteristics such as size and orientation. Using these discriminative features, some of the regions that do not show lane properties can be eliminated. The lane marking orientations in captured image vary in a certain range depending on the position of the image sensor and lens characteristics. The MSERs which have an orientation at the outside of the specified range are eliminated. Lane markings display slim shape which means that their height/width ratio is higher than 2 in general. In the proposed method, ellipses are fitted to

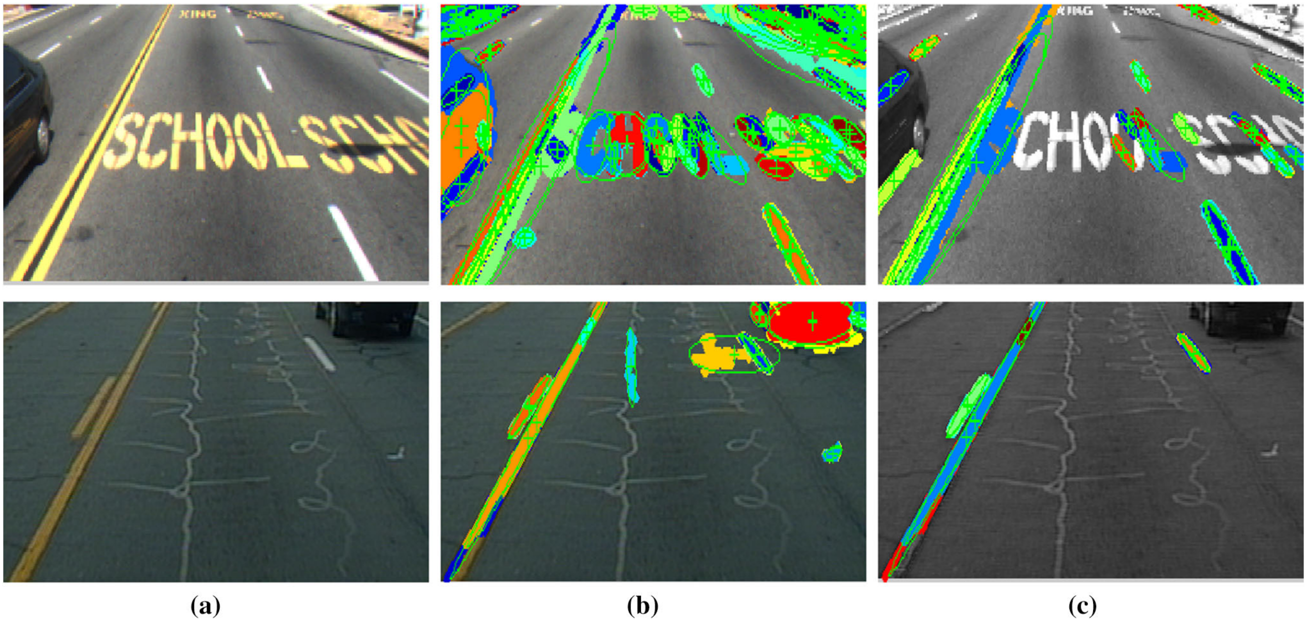


**Fig. 4** Structural elimination of MSERs **a** original image, **b** MSER detection result, **c** MSER detection results after orientation and size ratio elimination

MSERs in order to obtain their size ratio. If the ratio of a principal axis of the ellipse of a given MSER is lower than a fixed threshold, this MSER is simply eliminated. Figure 4 shows examples of this orientation and size ratio-based structural elimination of MSERs.

### *3.2.2 Gaussian-based template matching*

The lane markings have to be noticeable to the drivers and pedestrians. Thus, the lanes are painted with distinct colors that the drivers can easily differentiate the lanes from typical gray roads. It is obvious that there is dark-light–dark (DLD) intensity transition along the lane markings. The DLD transitions along the lanes that acquired from sampled rows of an example image are shown in Fig. 5 in the yellow squares.

Figure 6 shows the similarity between DLD intensity transition at an example lane marking and a 1-D Gaussian function. It is clear from this cropped figure portion that the DLD intensity transitions in the yellow squares in Fig. 5 are similar to the 1-D Gaussian function shown in Fig. 6. The MSERs obtained in previous stages can be further examined using this additional shape information. An MSER which is detected on lanes should display this DLD transitions. The normalized cross-correlation technique-based template matching is performed using a predefined template over the MSERs to eliminate non-lane regions similar to the method presented by Borkar et al. [6]. The template for the correlation computation is constructed by taking average of lane marking widths from different databases.

The cross-correlation computation is performed for sampled rows for a given MSER to reduce computational complexity. The sampling interval has a minimum value. However, the sampling interval can be increased proportional to height of the region. If the majority of template matching results for a given row in the MSER is greater than a fixed threshold, then this row is kept for next step. Otherwise, this row is eliminated since it is considered as a non-road region. The number of remaining rows in MSER region at this point is computed in the second elimination stage. If the majority of the rows for a given MSER is signaled as non-road area, then this MSER is directly discarded. This process is repeated for each detected MSER. Finally, the position giving the maximum correlation coefficient of each non-eliminated MSER is selected as lane feature points. Figure 7 shows feature points obtained using the above-mentioned majority voting approach.

### *3.2.3 Post-processing*

In the previous stage, candidate lane marking features are obtained. In the proposed lane detection method, straight lines are detected by RANSAC algorithm to eliminate outlier feature points. For this purpose, a line model is employed for RANSAC. The feature points that are located within a certain distance to the line are considered as inliers. The lane detection results after RANSAC stage is shown in Fig. 8.

## 4 Experimental results and mobile implementation

There is not a widely adopted approach for the performance comparison of lane detection methods. Some methods do not mention their evaluation metrics they used,

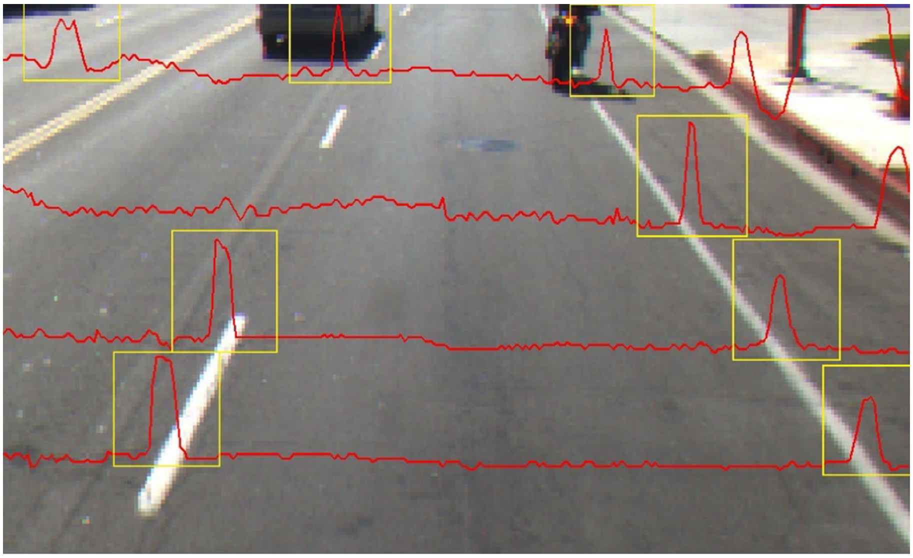


**Fig. 5** DLD transitions along the lanes

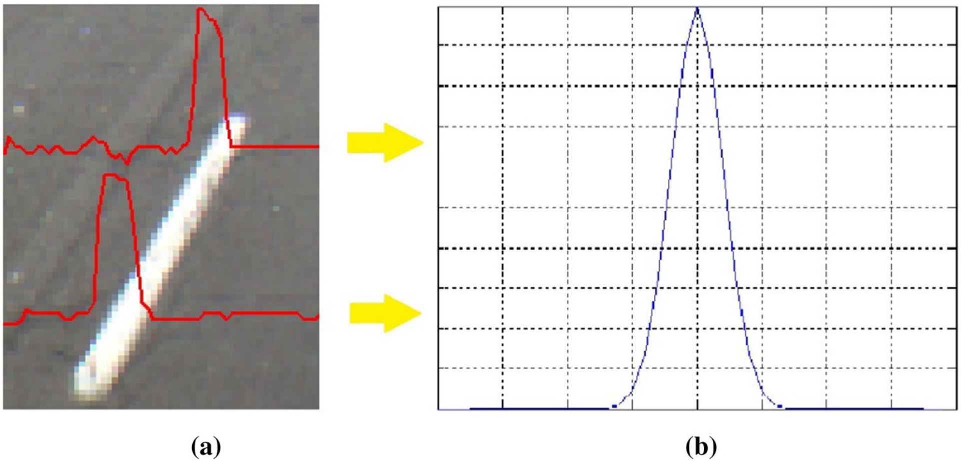


**Fig. 6** Similarity between the DLD transition and 1-D Gaussian shape **a** A lane segment, **b** example 1-D Gaussian function

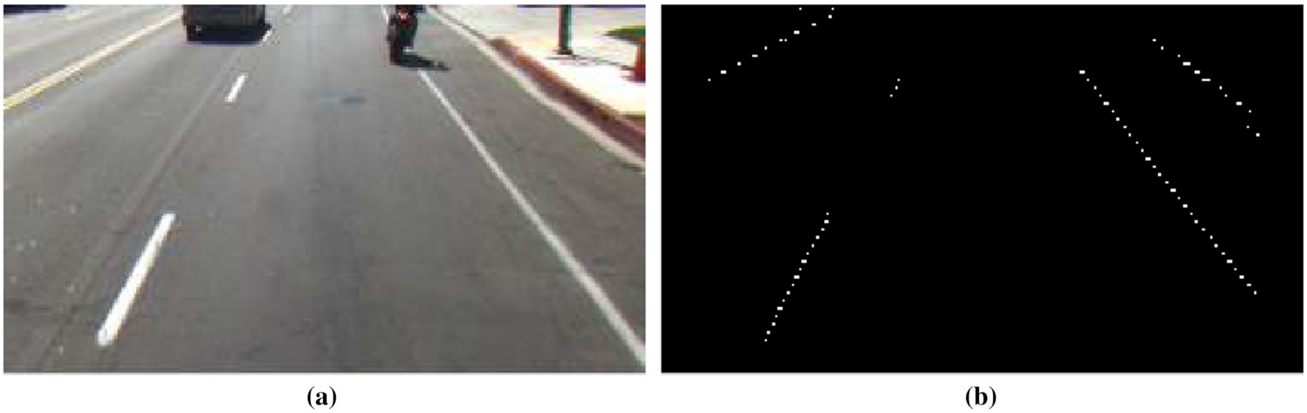


**Fig. 7** Proposed voting algorithm **a** original image, **b** the points showing lane features

whereas some of them employ only visual evaluation. In this work, an objective comparison approach is adopted by computing MSE (mean square error) between the ground truth and lane position estimation approach.

### 4.1 Ground truth generation

Ground truth of lane position is required to compute MSE performance of lane detection methods. Borkar et al. [6]

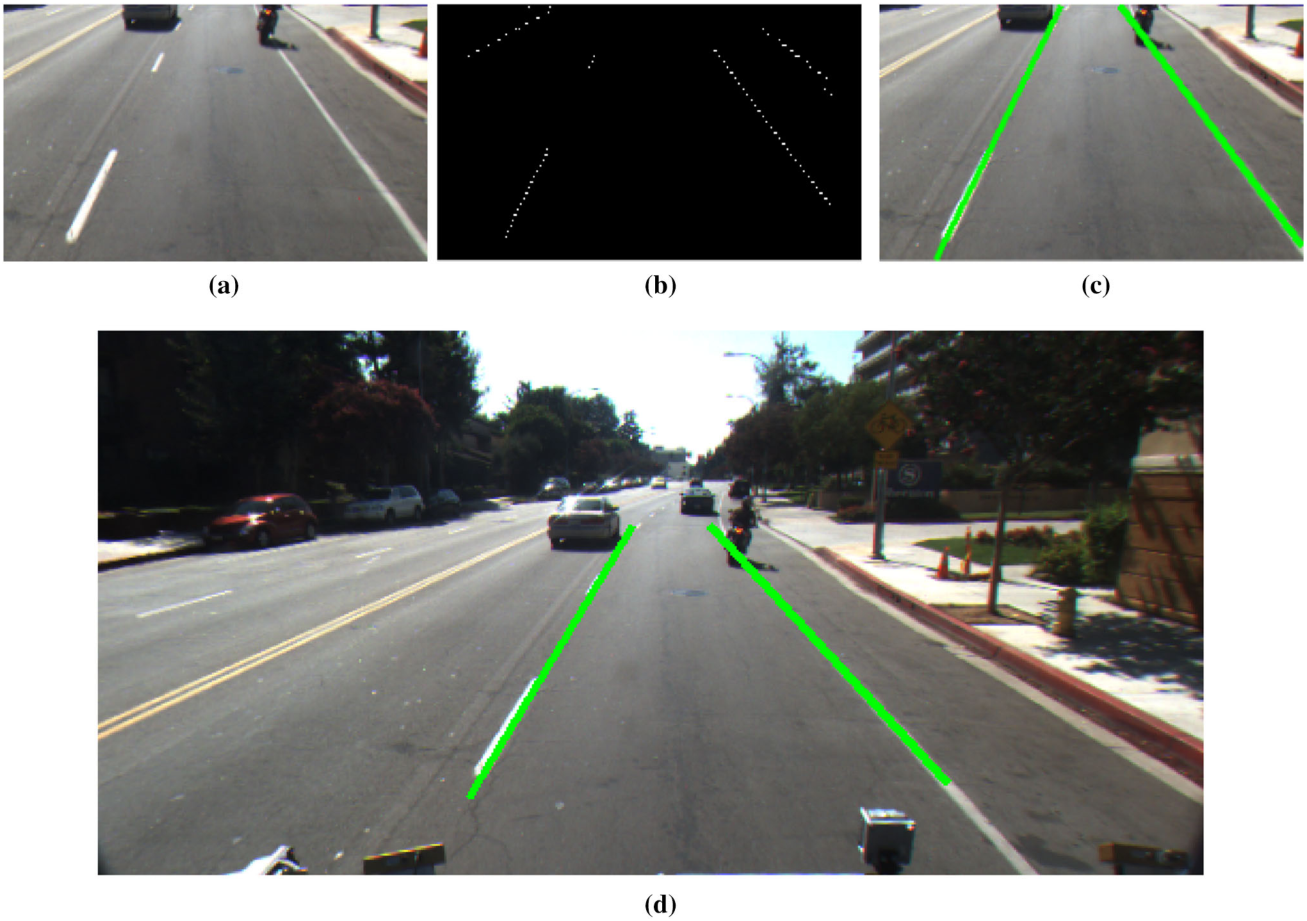


**Fig. 8** Lane detection result of proposed method **a** original image, **b** detected feature points, **c** detected results in ROI, **d** detection results shown on the original image

proposed a technique that uses time-sliced images and cubic spline interpolation. This approach works well on clearly marked roads and long-term sequences. But the interpolation may lead errors on non-marked road frames.

In this work, ground truth data are manually generated for each frame of used databases. The ground truth data generation begins by marking points on lane markings in predefined road area in the captured image. After manual marking finished, linear interpolation is used to fill points on lane marking locations between the rows in defined road area in the image. This process is repeated for each frame. Then obtained ground truth data are stored in a XML (Extensible Markup Language) file. The ground truth data of Caltech lane dataset [21] are generated using the proposed method. Borkar et al. [6] provides ground truth data of their datasets.

### 4.2 Lane detection results

Performance assessment of lane detection methods is performed by comparison of lane detection result and ground truth data. The most common performance evaluation metrics are true-positive rate (TPR), false-positive rate (FPR), and accuracy (ACC). These metrics calculated as

$$\mathrm{TPR} = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})} \tag{3}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{(\mathrm{FP}+\mathrm{TN})} \tag{4}$$

$$\mathrm{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{(\mathrm{TP}+\mathrm{FN}+\mathrm{FP}+\mathrm{TN})} \tag{5}$$

In lane detection performance comparison, true-positive (TP) means that the algorithm detects lane marking correctly when ground truth lane exists. It is considered as a false-positive (FP) case when the algorithm detects lane marking while no ground truth exists for this decision. True-negative (TN) means that the algorithm does not decide a lane mark while there is not a ground truth lane in reality. False-negative (FN) is the case where the algorithm does not detect any lane marking but ground truth data exist for this decision [22].

In this work, correct lane detection decision is given when the distance between detected lane marking and ground truth data points within a defined interval. This interval is set to half of the average lane marking width.

Figure 9 shows visual comparative results of the method presented by Borkar et al. [6], Espinoza et al. [10], Küçükyıldız et al. [18], Gaikwad et al. [4], Mammeri et al. [15] and proposed method for different input images from the Caltech and Borkar dataset. As seen from this figure, the proposed method is able to provide reliable lane detection result for harsh road and illumination situations.

Tables 2 and 3 show objective evaluation in terms of accuracy of these methods on Caltech and Borkar's dataset for left and right lane marking separately. Additionally, average of left and right lane marking detection performance is given in Tables 2 and 3 for a general comparison. The Caltech lane dataset has 1225 frames, whereas Borkar's dataset includes 2800 frames. Thus, the methods are compared on the 4025 images totally which includes challenging input images. Cordova 1 is a sequence with the least amount of adverse effects so that the lane markings look obviously. Cordova 2 sequence has extremely bright regions on the road area because of the illumination. Thus, this affects DLD intensity transition which is the most critical feature of lane markings. The shadows that are heavily observed in the Washington 1 sequence cause the lane markings to deform and decrease pixel values over the lane markings. Washington 2 sequence has text-like road surface signs. In general, the road surface signs have same color with the lane markings. Some of road surface signs like texts in the Washington 2 sequence could affect performance of lane marking detection because characters of text have slim shape with strong edges same as the lane markings. S1C1 sequence has road corruptions resemble to lane markings, and S2C1 sequence has shadows, vehicle near the lane markings and road surface signs. These adverse effects make it difficult to detect lane markings.

All the methods in Table 2 show good performance for Cordova 1 test sequence. When the lane detection performance of the methods in Table 2 for Cordova 2 sequence is assessed, the method presented by Gaikwad et al. [4] is the most affected method by illumination changes because of the nonadaptive PLSF parameters. The proposed method in this work is robust to the illumination changes thanks to MSER method implemented on enhanced B color channel. This improvement is achieved since MSER applies thresholding to the intensity image by a set of certain threshold levels in order to obtain a feature region. When the Washington 1 and S2C1 sequences are examined, it is seen that the proposed method in this study is the least affected method by shadows. The method in proposed by Mammeri et al. [15] is not affected so much as well. However, other method significantly affected from the shadow since they use a preprocessing filter to detect edges. This filter produces false edges at the shadow regions.

The proposed method by Mammeri et al. [15] uses MSER on gray scale images directly so that this method is able to detect lanes on shadow regions. However, these regions are smaller compared to the detected region by the proposed method. Since the proposed method employs enhanced B color channel, it is capable of detecting larger regions on the lane markings even in the shadow situations. Thus, it has better performance compared to the method presented by Mammeri et al. [15].

As seen from Tables 2 and 3, the proposed method provides best overall performance for these databases achieving by up to 76% accuracy. It is important to note that the method presented by Mammeri et al. [15] has considerably lower performance compared to the proposed method even though this method also employs MSER. This comparison reveals the proposed method is able to significantly improve performance of existing MSER-based lane detection method thanks to improvements introduced in this paper.

All parameters used in the proposed method with their fixed values are given in Table 4. Note that the performance results given in Tables 2 and 3 are given according to the parameter values given in Table 4. The performance of the proposed lane marking detection method is analyzed for different values (10 value) of all parameters used in the method. Figure 10 shows overall accuracy of proposed method for different values of the important parameters.

As seen from Fig. 10, the performance of the proposed method is not directly depended on the specific values of the parameters. The proposed method is able to provide high accuracy for wide range of input parameters. The only situation that the performance decreases is shown in Fig. 10j–l where MSER minimum orientation parameter is set to a value higher than 0.5 rad (i.e., 28.6°). This is an expected situation since the higher values of this parameter might eliminate real lane markings when it is increased. Thus, it is clear that performance of the proposed method is not depended on certain parameter values, fixed camera setup or a particular database.

The proposed method is implemented in MATLAB on a PC with 2.3 GHz Quad core CPU. Average total execution times for an input frame of methods proposed by Borkar et al. [6], Espinoza et al. [10], Küçükyıldız et al. [18], Gaikwad et al. [4], Mammeri et al. [15] and proposed method on this PC are given in Table 5. These methods are implemented in MATLAB for the performance evaluation of the proposed method. It is important to note that all the implantations have similar code optimization level. As seen from these results, the proposed method has reasonable level of computational load compared to the other methods in the literature. Even though the methods in [4]

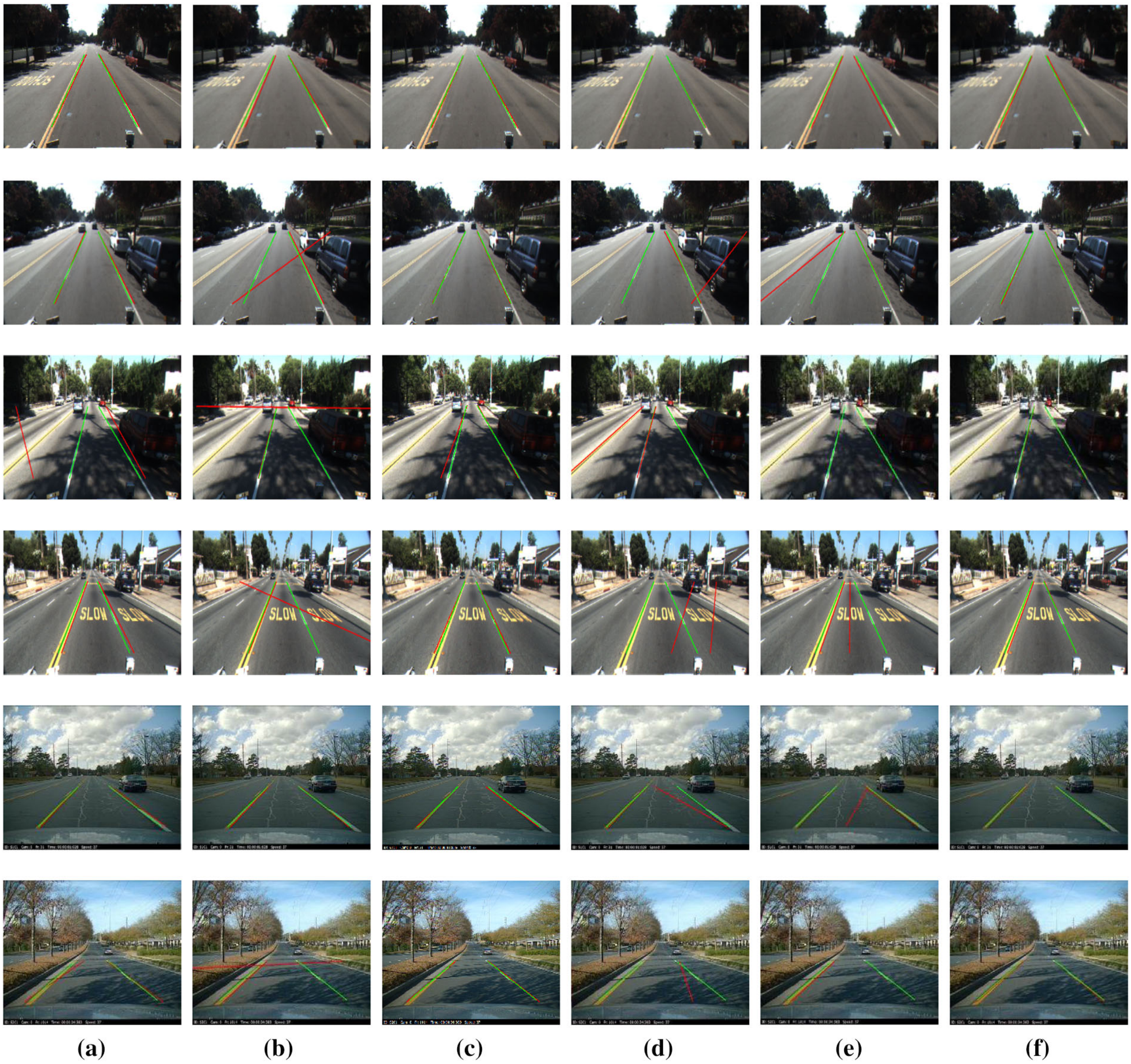


**Fig. 9** Lane detection results **a** Borkar et al., **b** Espinoza et al., **c** Küçükyıldız et al., **d** Gaikwad et al., **e** Mammeri et al., **f** proposed method

**Table 2** Evaluation results on Caltech lane dataset [21]

| Scene | Borkar et al. [6] | | Espinoza et al. [10] | | Küçükyıldız et al. [18] | | Gaikwad et al. [4] | | Mammeri et al. [15] | | Proposed method | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Left | Right | Left | Right | Left | Right | Left | Right | Left | Right | Left | Right |
| Cordova 1 | 0.77 | 0.68 | 0.63 | 0.66 | 0.83 | 0.91 | 0.68 | 0.36 | 0.88 | 0.70 | **0.92** | **0.92** |
| Cordova 2 | 0.62 | **0.51** | 0.58 | 0.42 | 0.53 | 0.25 | 0.06 | 0.14 | 0.78 | 0.33 | **0.87** | 0.33 |
| Washington 1 | 0.49 | 0.67 | 0.62 | 0.36 | 0.76 | **0.79** | 0.67 | 0.11 | 0.27 | 0.26 | **0.81** | 0.73 |
| Washington 2 | 0.79 | 0.79 | 0.47 | 0.59 | 0.85 | **0.88** | 0.83 | 0.27 | 0.77 | 0.59 | **0.87** | 0.80 |
| Average | 0.65 | 0.64 | 0.58 | 0.49 | 0.71 | **0.65** | 0.50 | 0.20 | 0.66 | 0.44 | **0.86** | **0.65** |
| Overall average | 0.64 | | 0.53 | | 0.68 | | 0.35 | | 0.55 | | **0.76** | |

The bold values show the best performace

**Table 3** Evaluation results on Borkar lane dataset [6]

| Scene | Borkar et al. [6] | | Espinoza et al. [10] | | Küçükyıldız et al. [18] | | Gaikwad et al. [4] | | Mammeri et al. [15] | | Proposed method | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Left | Right | Left | Right | Left | Right | Left | Right | Left | Right | Left | Right |
| S1C1 | **0.91** | 0.58 | 0.78 | 0.44 | 0.74 | 0.78 | 0.87 | 0.55 | 0.66 | 0.70 | **0.91** | **0.84** |
| S2C1 | **0.62** | 0.51 | 0.25 | 0.16 | 0.40 | 0.51 | 0.29 | 0.29 | 0.30 | 0.29 | 0.59 | **0.64** |
| Average | **0.76** | 0.55 | 0.51 | 0.30 | 0.57 | 0.65 | 0.58 | 0.42 | 0.48 | 0.49 | 0.74 | **0.74** |
| Overall average | 0.65 | | 0.41 | | 0.61 | | 0.50 | | 0.49 | | **0.74** | |

The bold values show the best performace

**Table 4** Parameters used in the proposed method

| Parameter | Value | Description |
|---|---|---|
| Correlation coefficient threshold | 0.8 | Used in the correlation computation |
| Lane marking width | 20 | Pixel size of lane marking width |
| Sigma | 2 | Sigma value of Gauss function |
| MSER maximum area variation | 0.1 | Maximum area variation value between extremal regions at varying intensity threshold levels |
| MSER minimum orientation | 0.4 | Minimum orientation of a MSER as a lane marking |
| MSER maximum orientation | 1.2 | Maximum orientation of a MSER as a lane marking |
| RANSAC distance | 0.1 | The distance between data point and the straight line used to decide whether a point is an inlier or not |
| MSER region area range maximum | 4080 | Maximum size of the region in pixels |
| MSER region area range minimum | 10 | Minimum size of the region in pixels |
| MSER threshold delta | 1.5 | Step size between intensity threshold levels used as a criterion in MSER detection |
| Size ratio | 3 | MSER height/width ratio |
| Voting threshold | 0.2 | Number of correlation result greater than correlation coefficient threshold for a MSER proportionally height of a MSER |
| Sampling interval | 3 | Minimum value of correlation computation interval |
| Sample ratio | 0.01 | Sample ratio determines sampling interval proportionally height of a MSER |

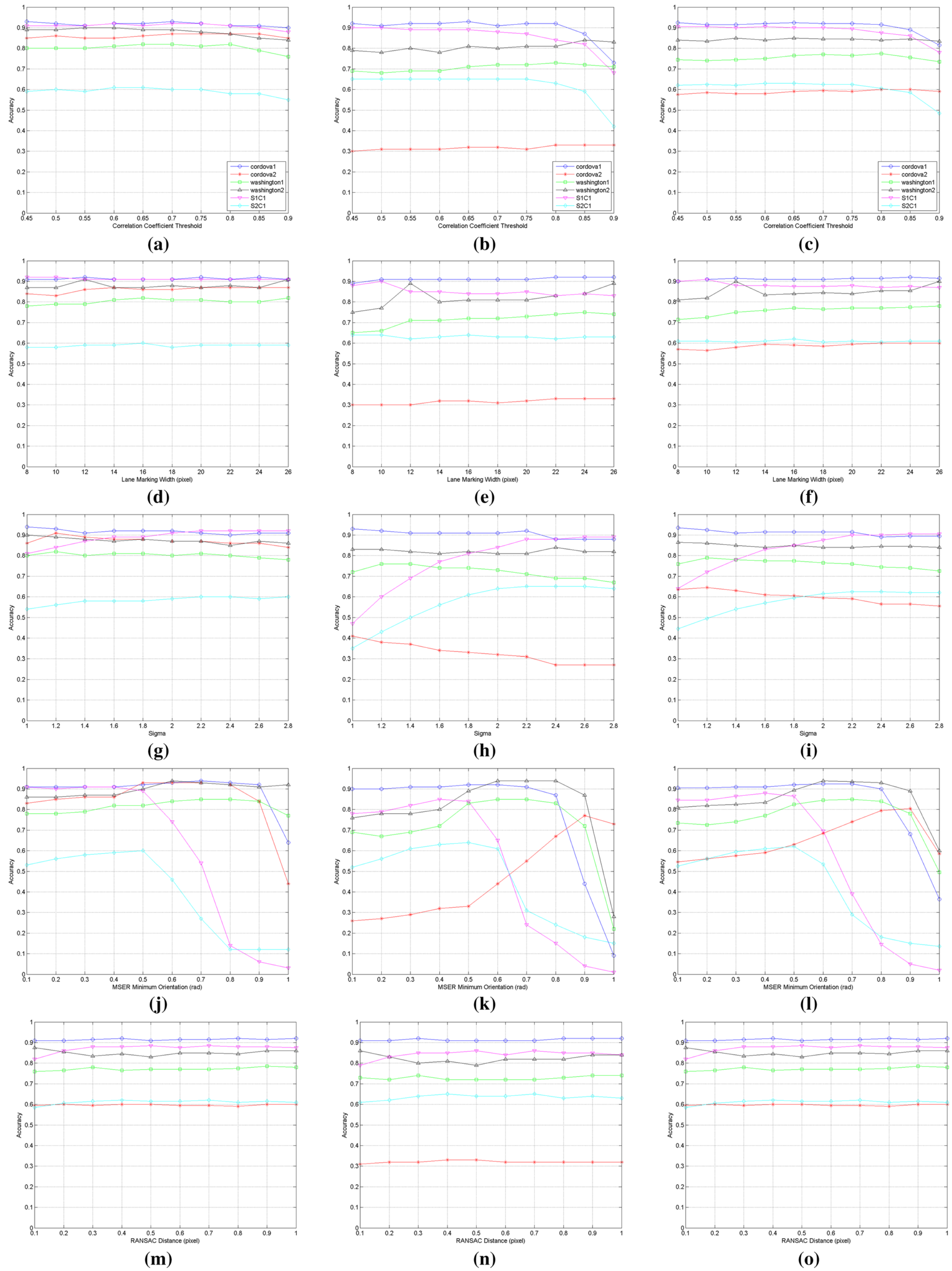
and [18] are faster than the proposed method, their lane detection performance is significantly lower compared to the proposed method as seen in Tables 2 and 3. Thus, it is clear that the proposed method has a good performance/speed trade-off.

### 4.3 Mobile platform implementation of the proposed method

The proposed method is also implemented on smart phones with Android OS and iOS using C++ language. For the iOS platform, the application is developed using Swift. We complied Open CV libraries to employ them in objective C. Next, we employ these objective C functions in Swift to process captured image frames. For the Android platform, the application is developed by Java-DVM. However, the image processing portion is written in C++ using Open CV libraries and compiled by making use of NDK to use it in Java-DVM.

Technical specifications of these devices are given in Table 6. Additionally, single-instruction multiple data (SIMD) structure provided by the mobile CPUs is employed to enable real-time processing. Table 7 shows average execution times of the proposed method in milliseconds for VGA (640 × 480 pixel) input images. Note that execution times are obtained using C++ systems functions.

Usage of SIMD significantly improves computation performance. As seen from this table, the total times required for a single video frame are 30.94 ms for Android OS and 22.65 ms for iOS. These total times correspond 32.32 and 44.15 fps processing speed, respectively. In Fig. 11, screenshots of mobile device screen are shown while the proposed method is running on the mobile device.



**(a)** **(b)** **(c)**

**(d)** **(e)** **(f)**

**(g)** **(h)** **(i)**

**(j)** **(k)** **(l)**

**(m)** **(n)** **(o)**

**Fig. 10** Evaluation results of proposed method with varying parameters **a** left, **b** right, **c** average detection results with correlation coefficient threshold; **d** left, **e** right, **f** average detection results with lane marking width; **g** left, **h** right, **i** average left lane marking detection results with sigma; **j** left, **k** right, **l** average detection results with MSER minimum orientation; **m** left, **n** right, **o** average detection results with RANSAC distance

**Table 5** Average execution times of different methods on a PC

| | Borkar et al. [6] | Espinoza et al. [10] | Küçükyıldız et al. [18] | Gaikwad et al. [4] | Mammeri et al. [15] | Proposed Method |
|---|---|---|---|---|---|---|
| Time | 8520 ms | 2520 ms | 169 ms | 180 ms | 50 ms | 550 ms |

**Table 6** Technical specifications of smart phones

| | Sony Xperia C4 | iPhone 6S |
|---|---|---|
| CPU | Octa-core 1.7 GHz Cortex-A53 | Dual-core 1.84 GHz Twister |
| RAM | 2 GB | 2 GB |
| Screen resolution | 1080 × 1920 pixels | 750 × 1334 pixels |
| OS | Android OS v5.0 (Lollipop) | iOS 9 |

**Table 7** Average execution times on a smart phone

| Stage | Time consumed on Android OS (ms) | Time consumed on iOS (ms) |
|---|---|---|
| Preprocessing | 14.60 | 6.93 |
| Feature extraction | 16.10 | 15.16 |
| Post-processing | 0.24 | 0.56 |
| Total | 30.94 | 22.65 |

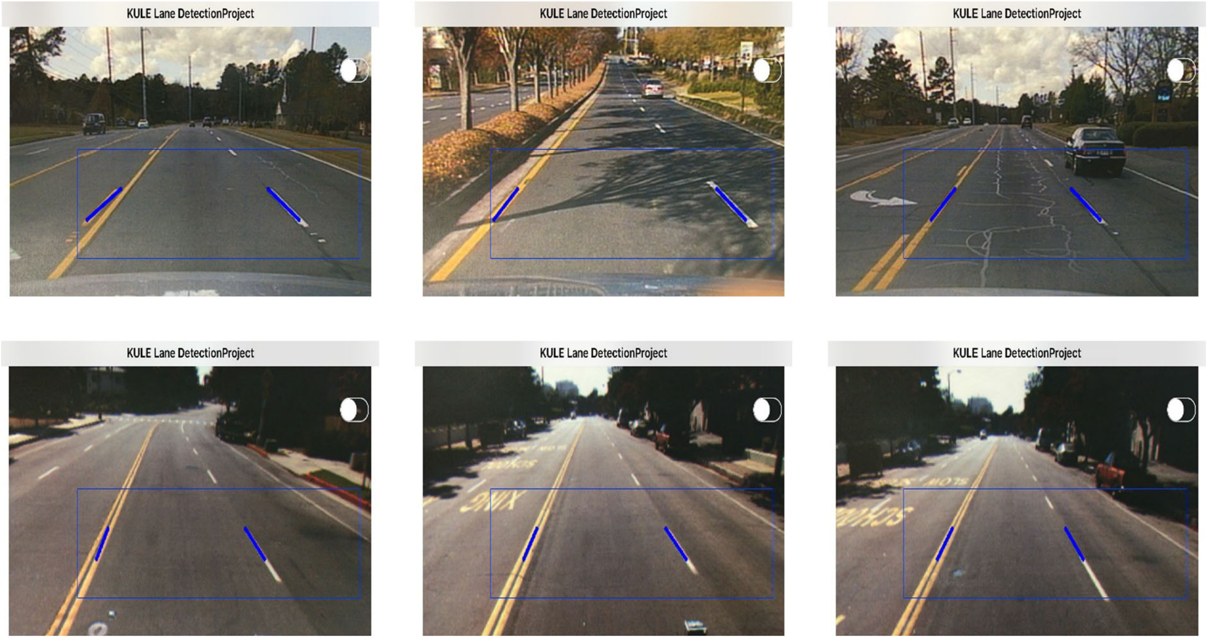

KULE Lane DetectionProject
KULE Lane DetectionProject
KULE Lane DetectionProject
KULE Lane DetectionProject
KULE Lane DetectionProject
KULE Lane DetectionProject

**Fig. 11** Screenshots of mobile device implementation of proposed method

## 5 Conclusions

In this paper, a novel lane detection method using a single input image is presented. The main target of the proposed method is providing robust lane detection results even in the case of severe shadow and illumination effects. Enhanced B color channel image is feed to MSER approach to detect candidate regions. Next, characteristics of MSER are evaluated together with a 1-D correlation-based approach to decide final candidate lane points. At the last stage, outliers are eliminated to decide lane positions. The experimental results show that the proposed method is able to determine lane markings in various conditions at significantly higher accuracy compared to existing methods in faster manner. Real-time mobile device implementations are also provided to show consumer-grade device performance of the proposed method. We will investigate nighttime performance of the proposed method as a future work.

**Acknowledgements** Authors would like to thank the editor and reviewers for their constructive recommendations which we believe have improved the understanding and representation of the paper. Authors also would like to thank M. Torres-Torriti for sharing their implementation and A. Borkar and H. Ocak for helping the implementation of their methods.

## References

1. Nedevschi, S., Schmidt, R., Graf, T., Danescu, R., Frentiu, D., Marita, T., Oniga, F., Pocol, C.: 3D lane detection system based on stereovision. In: 7th International IEEE Conference on Intelligent Transportation Systems, pp. 161–166 (2004)
2. Gao, F., Jiang, D., Xu, G., Cui, Y., Wang, X.: A 3D curve lane detection and tracking system based on stereovision. CICTP **2012**, 1247–1258 (2012)
3. Chunzhao, G., Mita, S., McAllester, D.: Lane detection and tracking in challenging environments based on a weighted graph and integrated cues. In: Intelligent Robots and Systems, pp. 5543–5550 (2010)
4. Gaikwad, V., Lokhande, S.: Lane departure identification for advanced driver assistance. IEEE Trans. Intell. Transp. Syst. **16**(2), 910–918 (2015)
5. Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Commun. ACM **15**, 11–15 (1972)
6. Borkar, A., Hayes, M., Smith, M.T.: A novel lane detection system with efficient ground truth generation. IEEE Trans. Intell. Transp. Syst. **13**(1), 365–374 (2012)
7. Bertozzi, M., Broggi, A.: GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection. IEEE Trans. Image Process. **7**, 62–81 (1998)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
9. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Basic Eng. **82**, 35–45 (1960)
10. Tapia-Espinoza, R., Torres-Torriti, M.: Robust lane sensing and departure warning under shadows and occlusions. Sensors **3**(1), 3270–3298 (2013)
11. Torr, P.H.S., Zisserman, A.: MLESAC: a new robust estimator with application to estimating image geometry. Comput. Vis. Image Underst. **78**, 138–156 (1996)
12. Yoo, H., Yang, U., Sohn, K.: Gradient-enhancing conversion for illumination-robust lane detection. IEEE Trans. Intell. Transp. Syst. **14**(3), 1083–1094 (2013)
13. You, F., Zhang, R., Zhong, L., Wang, H., Xu, J.: Lane detection algorithm for night-time digital image based on distribution feature of boundary pixels. J. Opt. Soc. Korea **17**(2), 188–199 (2013)
14. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**, 62–66 (1979)
15. Mammeri, A., Boukerche, A., Tang, Z.: A real-time lane marking localization, tracking and communication system. Comput. Commun. **73**, 132–143 (2016)
16. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of British Machine Vision Conference, pp. 384–396 (2002)
17. Stephens, R.S.: Probabilistic approach to the Hough transform. Image Vis. Comput. **9**(1), 66–71 (1991)
18. Küçükyıldız, G., Ocak, H.: Development and optimization of a DSP-based real-time lane detection algorithm on a mobile platform. Turkish J. Electr. Eng. Comput. Sci. **22**(6), 1484–1500 (2012)
19. Kim, J., Kim, J., Jang, G., Lee, M.: Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection. Neural Netw. **87**, 109–121 (2017)
20. Tveit, A., Morland, T., Rost, T.: DeepLearningKit—an GPU optimized deep learning framework for Apple's iOS, OS X and tvOS developed in Metal and Swift. arXiv:1605.04614 (2016)
21. Aly, M.: Real time detection of lane Markings in urban streets. In: 2008 IEEE Intelligent Vehicle Symposium, pp. 7–12 (2008)
22. Kumar, A.M., Simon, P.: Review of lane detection and tracking algorithms in advanced driver assistance system. Int. J. Comput. Sci. Inf. Technol. **7**(4), 65–78 (2015)

**Ayhan Küçükmanisa** received his B.Sc., M.Sc. degrees in Electronics and Telecommunication engineering from the University of Kocaeli, Kocaeli, Turkey, in 2010, 2012, respectively. Currently, he is pursing toward to Ph.D. degree. Since 2013, he has been with the Department of Electronics and Telecommunications Engineering, University of Kocaeli, Turkey as a research assistant. His research interests include image/video processing and embedded systems.

**Gökhan Tarım** received his B.Sc. degree in Electronics and Telecommunication engineering from the University of Kocaeli, Kocaeli, Turkey, in 2015. Since 2015, he has been with Kocaeli University Laboratory of Embedded and Vision Systems (KULE) as a researcher. His research interests include image processing and embedded systems.

**Oğuzhan Urhan (S′02–M′06)** received his B.Sc., M.Sc., and Ph.D. degrees in Electronics and Telecommunication engineering from the University of Kocaeli, Kocaeli, Turkey, in 2001, 2003, and 2006, respectively. Since 2001, he has been with the Department of Electronics and Telecommunications Engineering, University of Kocaeli, Turkey, where he is currently full professor. He was a visiting professor at Chung-Ang University, South Korea, from 2006 to 2007. He is the director of Kocaeli University Laboratory of Embedded and Vision Systems (KULE). His research interests include digital signal, image/video processing and embedded systems.