



Inverse perspective mapping for real-time Lane Detection in City Streets

ALIREZA BOSAGHZADEH ^{1*}, MAJID NASIRI MANJILI ²

¹ Faculty of Computer Engineering, Shahid Rajaei Teacher Training University, Tehran, Iran

² Faculty of Computer Engineering, Shahid Rajaei Teacher Training University, Tehran, Iran

ARTICLE INFO

Article history:)

Received : 4 May 2020

Accepted: 29 Aug 2020

Published: 1 Sep 2020

Keywords:

lane detection;

lane model;

autonomous driving;

inverse perspective mapping;

Probabilistic Hough Transform

ABSTRACT

Lane detection is a crucial step in any autonomous driving system in order to decrease car accidents and increase safety. In this paper, based on inverse perspective mapping and Probabilistic Hough Transform, we propose a lane detection system which works on city street images. First, we calculate the top view of street image by adopting inverse perspective mapping. Second, the lanes are rectified using a specifically designed filter which enhances the lanes and suppresses other elements. Then, by using Probabilistic Hough transform the location of the lanes is detected in the images. For the final refinement, lane candidates are mapped to the road image using perspective mapping and the lane intensity is analyzed to reduce false acceptance. We evaluate the performance of the proposed method on Caltech-lane dataset and the obtained results prove that the proposed algorithm is able to detect straight lanes.

*Corresponding Author

Email Address: alibossagh@yahoo.co.uk

<https://doi.org/10.22068/ase.2020.540>

1. Introduction

To avoid thousands of accidents happening every day, we need to equip our cars in a way that they can automatically realize road conditions and road marks. For this reason, a lot of researchers have focused on Driver Assistance Systems (DAS) [1,2] which provides assistance for the driver and autonomous car driving. Machine learning [1] and image processing [3,4] are two important tools which ease man processes in DAS. Many outside objects like the pedestrians, shadows from trees and buildings and weather conditions, make autonomous car driving in city streets more complex than highways. Determining the correct location of the car on the road is one of the important steps in any autonomous car driving and DAS. For both purposes, Lane Departure Warning (LDW) is a very important task and lane detection is a key step to reach this goal. Therefore it is vital to have precise information about the location of road lanes and road boundaries to keep a car in the line.

Presence of too many external conditions (e.g. daytime and weather) and different lane marks, retained this area still challenging and active. This topic has been widely studied in [1-6]. Various vision-based lane detection techniques have been suggested with different approaches such as morphological operations [7,8], tracking based [9-11], Hough transform [12-17] and neural network [18-23].

2. MATERIALS AND METHODS

Many of the model based lane detection systems generally consist of the following steps [16, 24]: 1) camera calibration, 2) road lane feature extraction (edges, colors and etc), 3) model fitting. Lanes are straight lines if we look at them from the top view. However, due to the perspective projection, they lose this property. Knowing the camera parameters (using a camera calibration stage), different algorithms have been proposed which work based on Inverse Perspective Projection (IPM) [11, 24-28] to obtain the top view of the road.

In [24], the authors proposed a method based on RANSAC line fitting algorithm. At first, the

top view of the road is obtained using IPM. Then the image is filtered with a selective oriented Gaussian filters. The next step is an initial line estimation followed by a fast RANSAC algorithm to fit Bezier Splines. Similarly, the authors of [11], use the IPM image followed by line kernel filtering and RANSAC to detect lines. Furthermore, they adopt Kalman filter to track the lines and estimate their position. In [27], based on IPM, the authors proposed a lane detection method using an improved feature map. Similar to previous approaches, IPM is used to eliminate the perspective effect from the road image. They first estimate the global lane orientation using edge orientation histogram. Then, based on the initial estimation of global lane orientation they increase the SNR of the feature map. 1-D hough transform is used to search for the lines and finally, lanes are detected by fitting parabolic lane models. For IPM, one needs the camera calibration in order to estimate and remove the effect of perspective projection. The authors in [28] proposed a calibration-free algorithm that uses road boundaries for an automatic IPM. After obtaining the road top view, the image is filtered with a filter template and binarized by a fixed threshold value. Then the connected component analysis is employed for eliminating tiny areas. Lastly, the columns that have large non-zero values are considered as lane markings. The authors in [16] implement HT on a FPGA. They reduced the dimensionality of hough space and perform parallel voting. By parallel processing of the voting and synchronized initialization of Hough space, they enhance the speed on XGA video.

While most of the algorithms adopt gray level images for lane detection, the use color models has also absorbed attention [29-31,33,35]. The authors of [29] proposed lane detection based on HSI color model. In the first phase, the RGB image is converted into HSI color model and difference of intensity distribution is clustered with fuzzy c-means algorithm. In [30] the authors adopted YIQ color space. For spatial information processing, they combine fuzzy C-mean, self-clustering algorithm (SCA), and fuzzy rules and detect brightness changes. To locate the lane boundaries they use fan detection and canny edge detector. The authors of [31] proposed a lane

mark color predictor obtained by training a Gaussian Radial Basis Function on the RGB color information of lane marks. The system then receives an RGB image classifies the pixels as candidates. Horizontal gradient of the candidates are used to accept the candidates as lane marks or reject them. A final Hough line fitting is used to remove outliers. Recently, the use of color features as the inputs of a fuzzy inference system (FIS) is proposed in [32] to overcome the problem of shadow in road images. As the first feature, they use the color difference in the HSV color space and the second one is the difference in the gray level image. The shadow level is then determined by feeding the features into the FIS and the line detector parameters are then tuned accordingly. Further conditions are applied to remove invalid line segments.

With the recent success of Deep Neural Networks (DNN) in different tasks, several methods have been proposed that use them to detect lanes [19-23]. In [19] a Safe Lane-Change Aid Network (SLCAN) is proposed by training a Deep Convolutional Network to detect and classify lanes. The authors of [20] proposed two neural networks scenario. The first network determines the existence of the geometric attributes and the target via a multi-task deep convolutional network and the second network is used for structured visual detection. Both networks are used for lane boundary detection in traffic scenes. In [22], a DNN is proposed that detects the lanes from down facing cameras. Apart from being lane boundary detection, the system has a sub-centimeter accuracy.

Recently, the authors of [34] proposed a lane detection technique that is able to distinguish between dashed and solid lanes. Their method has two stages, main processing and post-processing. In the first step, they determine the region of interest and then find the line segments. In the second step, they use the angle of line segments to eliminate wrong line segments. In the last step is the lanes are detected.

The proposed method in this article adopts inverse perspective mapping (top view), two-dimensional Gaussian filter, lane localization and probabilistic Hough transformation. The algorithm works in real-time, needs few input

parameters and is robust to changes in intensities caused by different weather conditions.

It has to be mentioned that our work is different from the one proposed [24] in several aspects. First, in [24] the authors have the exact position of the camera (hence the projection matrix) using different sensors while in ours we obtain the projection matrix using only one dashed lane segment. Second, we adopt Probabilistic Hough Transform in order to find lanes instead, in [24] the authors used RANSAC [36] line and spline fitting. Third, in our method, the possible location of the lanes in the inverse perspective mapping image is obtained using a heuristic method.

The rest of this article is organized as follows: the proposed algorithm is explained in section 3. Section 4 is dedicated to the experimental results and Section 5 concludes this article..

3. PROPOSED METHOD

In this section, we explain the proposed algorithm for lane detection which has six steps till a lane is detected in the image. Due to perspective projection in imaging systems, lanes are distorted in the captured image. In the first phase, we adopt IPM to obtain the top view of the road. In the second step, a trapezoid ROI to find lanes is determined. In the third phase, the top view images are filtered such that the lanes are rectified while other objects are suppressed and then by thresholding the lane candidates are selected. The fourth step finds the lanes in the image while in the fifth phase; lines are fitted to the detected lanes. Finally, by post-processing some refinements are applied to improve the performance. The overall procedure of the proposed method can be seen in Figure 1. The blue rows demonstrate the steps while the gray rows show the sub-steps correspond to each step. In the following, each of the above-mentioned parts is explained in details.

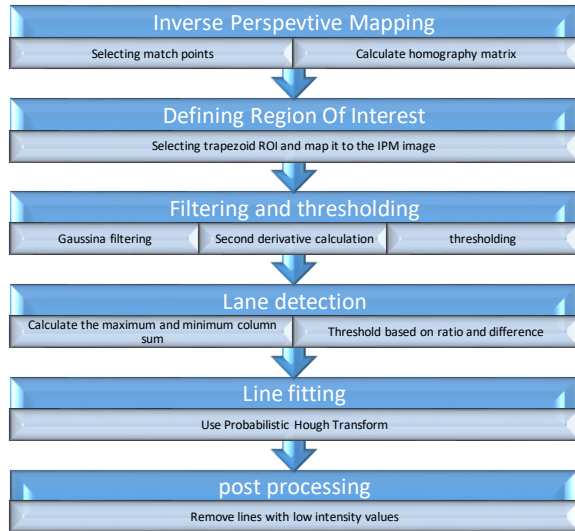


Figure 1: Overall procedure of the proposed method

3.1 INVERSE PERSPECTIVE MAPPING

Due to perspective projection in imaging systems, lane distances and lane widths change in different parts of an image. Inverse Perspective Mapping has been proposed to overcome this problem [37]. Hence, the first step is to eliminate the perspective effect of the images. To apply IPM and obtain the top view image, one needs a matrix named inverse perspective mapping matrix (homography matrix). To obtain homography matrix, the camera parameters, the distance of the camera to the road (road assumed to be flat), rotation of the camera, yaw and pitch of the camera are needed. To reduce computational cost it's assumed that yaw is too small. Knowing the entire above-mentioned parameters one can calculate the homography matrix [24].

However, if those parameters are unknown, there is another solution which can be adopted. Having four points in the image plane with their corresponding top view, one can estimate the homography matrix. For this purpose, an image which contains a dashed lane is needed. Then, four points corresponding to the corners of a dashed lane $\{p_1, p_2, p_3, p_4\}$ are extracted. The goal is to find a mapping (homography matrix) which transfer them into four relative points in the top view $\{p'_1, p'_2, p'_3, p'_4\}$ (See Figure 2). Homography matrix can be computed using these corresponding

points $(p_1, p'_1), (p_2, p'_2), (p_3, p'_3), (p_4, p'_4)$. The procedure is as follows.

These points are written as a 9×2 matrix as

$$P_i = \begin{bmatrix} -x_i & -y_i & -1 & 0 & 0 & 0 & x_i \hat{x}_i & y_i \hat{x}_i & \hat{x}_i \\ 0 & 0 & 0 & -x_i & -y_i & -1 & x_i \hat{y}_i & y_i \hat{y}_i & \hat{y}_i \end{bmatrix} \quad (1)$$

By stacking the p_i matrices calculated for the 4 corners, the P matrix is obtained as (2) where $P * H = 0$

$$P * H = \begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1 \hat{x}_1 & y_1 \hat{x}_1 & \hat{x}_1 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1 \hat{y}_1 & y_1 \hat{y}_1 & \hat{y}_1 \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2 \hat{x}_2 & y_2 \hat{x}_2 & \hat{x}_2 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2 \hat{y}_2 & y_2 \hat{y}_2 & \hat{y}_2 \\ -x_3 & -y_3 & -1 & 0 & 0 & 0 & x_3 \hat{x}_3 & y_3 \hat{x}_3 & \hat{x}_3 \\ 0 & 0 & 0 & -x_3 & -y_3 & -1 & x_3 \hat{y}_3 & y_3 \hat{y}_3 & \hat{y}_3 \\ -x_4 & -y_4 & -1 & 0 & 0 & 0 & x_4 \hat{x}_4 & y_4 \hat{x}_4 & \hat{x}_4 \\ 0 & 0 & 0 & -x_4 & -y_4 & -1 & x_4 \hat{y}_4 & y_4 \hat{y}_4 & \hat{y}_4 \end{bmatrix} = 0 \quad (2)$$

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = 0 \quad (2)$$

To avoid the trivial solution of H being all zeros, we add the constraint of $|H| = 1$. Using Singular Value Decomposition one can calculate the solution (i.e., H matrix).

Once the homography matrix H is obtained, the points can be projected using Eq. (3). For further information and mathematical proves of Eq. (1), (2) and (3), we refer the interested readers to [[38]].

In this article, we adopt the second scenario. Only in one image, we manually select four corner points correspond to the corners of a dashed lane and calculate the homography matrix using the above-mentioned procedure. By applying inverse perspective projection of an image using the calculated Homography matrix, we obtain the top view of the image where the distance and width of the lanes are constant. Figure 3 shows an example of an image and its top view. As we observe the lanes are now parallel and horizontal.

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3)$$

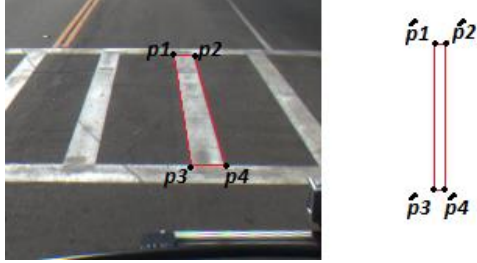


Figure 2: Selecting four corresponding points to calculate Homography matrix

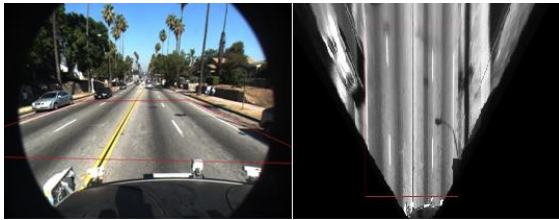


Figure 3: Image space and inversed perspective mapping view.

3.2. Region Of Interest

The next step is ROI detection to look for lanes. As the image contains a lot of unnecessary parts such as sky, trees and etc., defining a Region Of Interest (ROI) have mainly two advantages. Firstly, it reduces the computational time of the method, as we will not search the whole image but a fraction of it. Secondly, it reduces the environmental effects cause by rain and fog and can reduce the false positive error.

The authors in [39] used vanishing point to define ROI. An important remark about the ROI is that it has to have the road with the least amount of unnecessary area (the area that we know does not contain lanes). While in the top view image the lanes are in a rectangular area, it is not the case in the perspective view. Hence, adopting a rectangular ROI [32,34] in the perspective view is not the best choice.

To define a proper ROI, first, a rectangular area in the IPM image is defined (Figure 3. right). Then by applying the perspective mapping, the ROI in the perspective image is obtained. As we observe in Figure 3.left the obtained ROI has a trapezoidal shape and contains only the area in which the lane may appear.

3.3. Filtering and Thresholding

After calculating the image top view, the next step is to detect possible lanes. This process has two stages, filtering and thresholding. In the first step, the image is filtered such that the lanes are enhanced while other objects are suppressed. Then, using a variable threshold value, the candidates for lanes are selected.

In order to magnify lanes, we apply a two dimension filter similar to the one used in [24]. This filter is composed of a one dimension Gaussian smooth filter $f_v(y) = \exp(-\frac{1}{2\sigma_y^2} y^2)$ with σ_y^2 variance which helps lanes in the vertical orientation to be smoother and a one dimension second derivative Gaussian filter $f_h(x) = \frac{1}{\sigma_x^2} \exp(-\frac{1}{2\sigma_x^2} x^2)(1 - \frac{x^2}{\sigma_x^2})$ with σ_x^2 variances to magnify low-high-low intensity sequences in the horizontal orientation. While the first filter smoothes the image to reduce noise, the second one magnifies the lanes. Figure 4.left shows a sample image after filtering. As we observe, the lanes have higher values compared to non-lane parts of the image.

The next step is to select the lane candidates by thresholding. Since the lanes are enlarged, it is easy to separate them by thresholding. Similar to [24], a threshold is selected such that 97.5% of the pixels with low values are set to zero. In the thresholded image, all pixels with intensity values higher than the threshold retain their original values, while those pixels with intensity values less than the threshold had their values set to zero. Figure 4.left shows the image after applying the threshold.

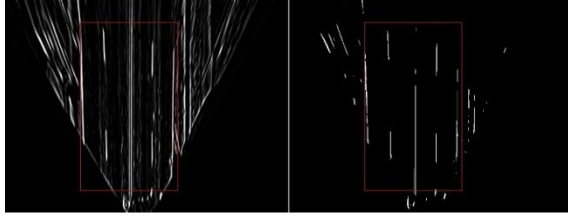


Figure 4: Filtered image and threshold image.

3.4. Lane Detection

In this step, the goal is to locate the lane candidates and to remove the candidates which were erroneously selected in the previous step. This step will further remove possible false points. The procedure is as follows.

In the thresholded image, the intensity of pixels are summed up column by column that leads to a vector with the length equal to the number of ROI columns and we call it “sum vector”. Then, a smooth Gaussian filter is applied to this vector to merge close lines. Locations of local maxima in this vector are initial candidates for lane location. However, not all local maxima correspond to the lanes and some of them might happen due to the noise or illumination variations such as lens flare or spreading sunlight in the captured image.

To refine these locations, the following operations are adopted, a) normalize all of the sum vectors in a way that the maximum value is equal to one. This step removes the differences that may happen when the images are captured in different illumination conditions. b) Eliminate values smaller than 0.2 which correspond to small objects with low-intensity values which we are sure, they are not lanes. However, in the conditions that the sun is facing a car, the values of sum vector take biases that cause many maxima that exceed 0.2. Hence, to remove these wrong local maxima, the rest of the procedure is proposed. c) Define a window proportional to local maximum value and calculate ratio and diff using Eq. (4) and Eq.(5). d) remove the maxima that have ratio smaller than 0.5 and diff value lower than 0.1. Ratio and diff constraints help to robust lane location from lens flare or spreading sunlight in the road.

$$\text{ratio} = \text{maxval} / \text{minval} \quad (4)$$

$$\text{diff} = \text{maxval} - \text{minval} \quad (5)$$

In these formulas, maxval and minval are local maximum and minimum values in the related window, respectively. Figure 5 show example of minval and maxval points for two candidates and their corresponding window. Candidate points that pass these conditions are selected for line fitting. Figure 6 shows the initial candidates and the red stars are the selected points for the next stage.

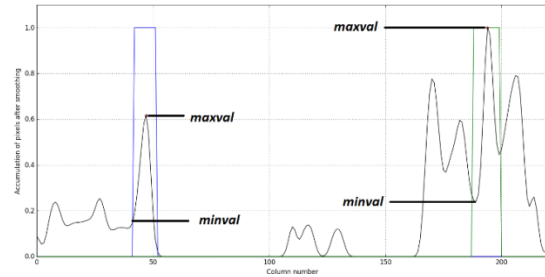


Figure 5: Minval and maxval points for two candidates.

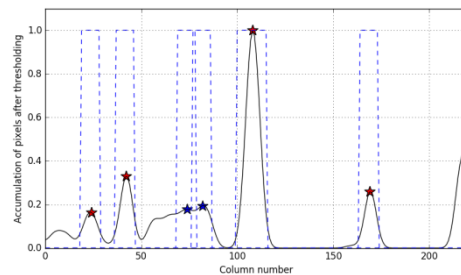


Figure 6: Red and blue stars are initial candidates for lane location. Red stars are winner candidates for lane location. Blue dot lines show the windows related to local maxima

3.5. Line Fitting

The goal of this phase is to group the candidate pixels which belong to a lane. Hough Transform (HT) is one of the main algorithms proposed for line detection. It is an algorithm based on the voting strategy in which each point, votes for the lines that pass through that point. This process is performed for all candidate points and finally, the lines that have the highest votes are selected. Every line is defined in the polar coordinate system as

$$\rho = x \cos(\theta) + y \sin(\theta) \quad (6)$$

where x and y are the location of a point on a line in the Cartesian coordinate system, θ is the

angle between the vector orthogonal to the line and the x-axis and ρ is the shortest distance between the origin and the selected line. For each candidate point (x_i, y_i) , each pair of (ρ, θ) that Eq. (6) are the parameters of a line that passes through that point. For a series of candidate points, the pairs that match with most of the candidate points are parameters of the lines that exist in the image. In order to find the lines in the image, each candidate points votes for the line parameters that pass through it. This process is time consuming especially if there are a lot of candidate points in the image.

Probabilistic Hough Transform (PHT) [40] is a variation of HT where instead of using all points for voting, it randomly selects sufficient number of samples to detect lines. According to [40], given all available input features, one can define the PHT (i.e., $H(\vec{y})$) as the log of probability density function (PDF) of the output parameters. Consider an input image with a set of feature measurements as $X_n = \{x_1, x_2, \dots, x_n\}$ where n is the number of features and a specific point in parameter space y . The PDF in the Hough space is $p(y|X_n)$ and consequently $H(y) = \ln[p(y|X_n)]$ is the PHT that by Bayes' rule is equal to

$$H(y) = \sum_{i=1}^n \ln[p(x_i|y)] + \ln[p(y)] + D$$

where D is a constant and $p(y)$ is a priori probability distribution.

In our work, we adopt PHT to fit one or more lines to the candidate points that are selected in the previous step. In our Hough model, the maximum allowable line gap is equal to 5 pixels and minimum line length is 30 pixels. Figure 7 shows the result of fitting lines on a thresholded image using PHT algorithm.



Figure 7: Red lines are fitted on the points in a thresholded image using Probabilistic Hough Transform.

3.6. Post-Processing

In the post-processing step that is the final step, final lane refinements are performed. After fitting lines on a thresholded image, the lines from top view space are projected to the image space. In the filtering step, we had used a 2d Gaussian filter that magnified low-high-low intensity sequences. This filter can cause high output values for the shadow of a tree or car, therefore, in some cases; they got selected erroneously as the lanes. For more refinement of lane localization, the calculated lines are projected to the image space and the line pixels that their intensity values are smaller than a specific threshold are eliminated. As setting a fixed value for this threshold can cause false rejection is the dark area, the threshold value is defined adaptively and is set to 30% of maximum intensity level. The lines that pass this step are considered as road lanes.

4. Experimental results

5. In this section we explain the results obtained from an empirical evaluation of the proposed lane detection system. Two subsets of the Caltech-lanes dataset namely Cordova 1 and Cordova 2 are selected for the evaluation in two modes. First, the 2-lanes mode in which the goal is to detect only two lanes boundaries on the left and right side of the car (Small ROI) and second, the all-lanes mode in which

Inverse perspective mapping for real-time Lane Detection in City Streets

all visible lanes in ROI are looked for (Larger ROI). Some of the features of the adopted database are described in Table 1. For numerical evaluations, after obtaining the number of correctly detected lanes (TP), number of missed lanes (FN) and number of erroneously detected lanes (FP), the precision, recall and F-measures have been calculated according to Eq. (7)-(9)

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP} \quad (7)$$

$$\text{Recall} = \frac{\#TP}{\#TP + \#FN} \quad (8)$$

$$F - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

Table 1: Dataset description

Clip	Dataset	#frame	#lane boundaries in 2-lane mode	#lane boundaries in all lanes in ROI mode
1	Cordova 1	250	466	1024
2	Cordoba2	406	472	845
Total		656	938	1869

Table 2 shows the results of the two-lane mode. As we observe, on the Cordova 1 set the algorithm has a very high precision and recall rate. Also in the Cordova 2 set, the recall is still high while we observe a larger value of false positive and consequently low value for precision. This is mainly due to the fact that in some parts of Cordova 2 dataset, the street boundaries are so similar to white and yellow lanes hence, erroneously are detected as lanes. As we will observe, this erroneous lane detection will also reduce the precision in the rest of the experiments. For visual demonstrations, Figure 8 shows some the result images for the two-lane detection scenario where the detected lanes are colored in green. We compare the results of our proposed method with the one proposed by Aly

[24]. As we observe in Table 3, the proposed method has a higher recall values for both datasets while a close precision in Cordova 1 and a lower value for Cordova 2 for the reason explained above.

Table 2: Results of the proposed method in 2-lane detection scenario

Dataset	#total	#detected	#TP	#FN	#FP	Precision	Recall	F-measure
Cordova 1	466	484	463	3	21	0.96	0.99	0.97
Cordova 2	472	828	464	8	364	0.56	0.98	0.71
Total	938	1312	927	11	385	0.70	0.99	0.82

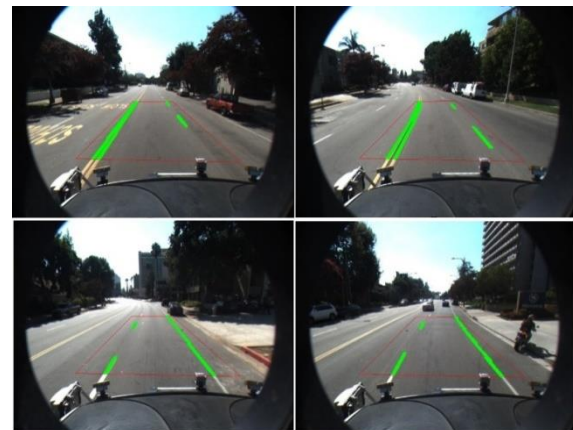


Figure 8: Proposed lane detection system for 2-lane mode on Cordova 1 (upper row) and Cordova 2 (lower row)

TABLE 3: Comparative results for 2-lanes mode

Dataset	Method	Precision	Recall	F-measure
Cordova 1	Ours	0.96	0.99	0.97
	[24]	0.97	0.97	0.97
Cordova 2	Ours	0.56	0.98	0.71
	[24]	0.62	0.96	0.75

Table 4 shows the result of the evaluation in the second mode, where the goal is to find all of the

lanes in the ROI. Similar to the two-lane mode, in Cordova 1 the precision and recall are high while in Cordova 2 the recall has a high value, meaning that most of the lanes have been detected. It should be mentioned that although the number of lanes has increased, the recall is close to the two-lane scenario, which shows that the proposed method is able to detect the lanes, no matter where the lanes are located and how big is the ROI. Figure 9 and Figure 10 shows some samples of lane detection for all visible lanes in the ROI in Cordova 1 and Cordova 2 datasets, respectively.

Moreover, we compare the performance of the proposed method with some other lane detection techniques namely, Aly method [24], and two very recent methods proposed by Hong, [32] and [34]. Since all methods have been implemented on the Caltech lane dataset, we use the performances reported by each author in their original article. As we observe from the results in Table 5, for Cordova 1 dataset the proposed method has the best performance (Precision, Recall and F-measure) compared to all comparative methods. For the Corvoda2 dataset, the proposed method has the highest recall value, meaning that it has the best performance to detect lanes if they present in the image. However, for the same reason explained before (false detection of road boundaries), the precision of the proposed method is not the best.

Table 4: Results of detecting all lanes

Dataset	#total	#detected	#TP	#FN	#FP	Precision	Recall	F-measure
Cordova 1	1024	1030	1002	22	28	0.97	0.97	0.97
Cordova 2	845	1290	830	15	451	0.64	0.98	0.77
Total	1869	2320	1832	37	479	0.79	0.98	0.84

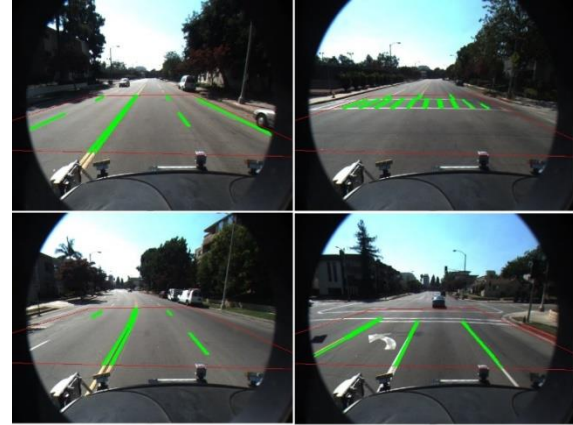


Figure 9: Our lane detection system for all visible lanes in ROI on Cordova 1.

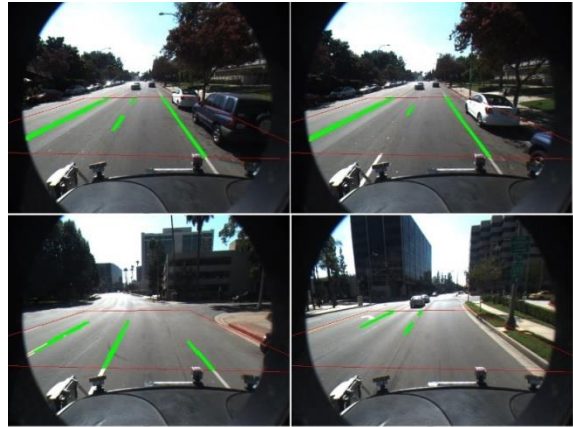


Figure 10: Our lane detection system for all visible lanes in ROI on Cordova 2.

Table 5: Comparative results for all visible lanes in ROI

Dataset	Method	Precision	Recall	F-measure
Cordova 1	Ours	0.97	0.97	0.97
	[24]	0.94	0.91	0.92
	[34]	0.96	0.93	0.94
	[32]	0.92	0.89	0.91
Cordova 2	Ours	0.64	0.98	0.77
	[24]	0.59	0.85	0.7
	[34]	0.87	0.98	0.92
	[32]	0.78	0.87	0.82

Figure 11 shows some examples that our model in spite of detecting all-lanes, has falsely detected some extra lanes. The false detection lanes are mainly due to the presence of road boundaries which has a similar shape to the road lanes.



Figure 11: Detecting false lanes in Cordova 2 dataset

Also, a system should not be forced to report lanes in an image and one way to evaluate the system is to observe the results when no lanes present in the image. Figure 12 shows the results on different images when no lane presents in the image. As we can see the system has not detected any lane in such cases.



Figure 12: Lane detection when no lane presents in the image of Cordova 1 (top row) and Cordova 2 (bottom row)

6. Conclusion

In this article, we proposed a new lane detection approach which contains several steps namely, inverse perspective projection, lane magnifying, lane selection, probabilistic Hough transform and post-processing based on intensity.

The proposed method has been evaluated on two datasets of Caltech-lane database and the results show that the proposed method has a high recall in detecting lanes. The proposed method is able to detect lanes in different scenarios such as 2-lane mode, all lanes in ROI, no lane, zebra crossing and curved lanes. The reason for the high false positive rate in the Cordova 2 dataset is the road boundaries that are similar to white and yellow lanes. Although detection of road boundaries is erroneous in the sense that they are not lanes, it does not conflict with the main idea of the algorithm because road boundaries are kind of lane marking and detecting them prevents the autonomous driving car from the departure of the road.

In our future work, we will focus on increasing the precision of our algorithm by differentiating between the lanes and the road boundaries. To further improve the performance, we propose working on filtering step in a way that avoids extracting shade of cars as lanes. Moreover, tracking the already detected lanes can further enhance the precision of the lane detection.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

- [1] Khodayari A, Yousefi M. Design an Intelligent Driver Assistance System Based On Traffic Sign Detection with Persian Context. *IJAE*. 2016; 6 (2) : 2138-2147.
- [2] Khodayari A, Yahyaei A. Design and implementation of a new algorithm for identifying moving objects around the car when leaving the park. *IJAE*. 2020; 10 (2) : 3255-3265.
- [3] Khodayari A, Ghaffari A, Fanni F. A Real Time Traffic Sign Detection and Recognition Algorithm based on Super Fuzzy Set. *IJAE*. 2017; 7 (1) :2346-2355.

- [4] Zohoorian Yazdi M, Soryani M. Driver Drowsiness Detection by Identification of Yawning and Eye Closure. *IJAE*. 2019; 9 (3) :3033-3044.
- [5] Wang, Yue, Dinggang Shen, and Eam Khwang Teoh. "Lane detection using spline model." *Pattern Recognition Letters* 21.8 (2000): 677-689.
- [6] Kim, ZuWhan. "Robust lane detection and tracking in challenging scenarios." *IEEE Transactions on Intelligent Transportation Systems* 9.1 (2008): 16-26.
- [7] Beucher, Serge, and Michel Bilodeau. "Road segmentation and obstacle detection by a fast watershed transformation." *Intelligent Vehicles' 94 Symposium, Proceedings of the*. IEEE, 1994.
- [8] Yu, Xuan, Serge Beucher, and Michel Bilodeau. "Road tracking, lane segmentation and obstacle recognition by mathematical morphology." *Intelligent Vehicles' 92 Symposium., Proceedings of the*. IEEE, 1992.
- [9] Zhao, Hongying, et al. "Annealed Particle Filter Algorithm Used for Lane Detection and Tracking." *Journal of Automation and Control Engineering* 1.1 (2013).
- [10] Liu, Guoliang, Florentin Wörgötter, and Irene Markelić. "Combining statistical hough transform and particle filter for robust lane detection and tracking." *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010.
- [11] Y. Lee, H. Kim; Real-time lane detection and departure warning system on embedded platform, 2016 IEEE 6th International Conference on Consumer Electronics, Berlin (ICCE-Berlin)
- [12] Leng, Yu-Chi, and Chieh-Li Chen. "Vision-based lane departure detection system in urban traffic scenes." *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*. IEEE, 2010.
- [13] Yoo, Hunjae, Ukil Yang, and Kwanghoon Sohn. "Gradient-enhancing conversion for illumination-robust lane detection." *IEEE Transactions on Intelligent Transportation Systems* 14.3 (2013): 1083-1094.
- [14] Liu, Guoliang, Florentin Worgotter, and Irene Markelic. "Stochastic lane shape estimation using local image descriptors." *IEEE Transactions on Intelligent Transportation Systems* 14.1 (2013): 13-21.
- [15] Daigavane, Prema M., and Preeti R. Bajaj. "Road lane detection with improved canny edges using ant colony optimization." *Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on*. IEEE, 2010.
- [16] Jungang Guan , Fengwei An , Xiangyu Zhang , Lei Chen and Hans Jürgen Mattauch , Real-Time Straight-Line Detection for XGA-Size Videos by Hough Transform with Parallelized Voting Procedures, *Sensors*, 2017,17,270
- [17] Abdelhamid Mammeri, Azzedine Boukerche, Zongzhi Tang, A real-time lane marking localization, tracking and communication system, In *Computer Communications*, Volume 73, Part A, 2016, Pages 132-143, ISSN 0140-3664,
- [18] Kim, ZuWhan. "Robust lane detection and tracking in challenging scenarios." *IEEE Transactions on Intelligent Transportation Systems* 9.1 (2008): 16-26.
- [19] S. G. Jeong, J. Kim, S. Kim and J. Min, "End-to-end learning of image based lane-change decision," 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, 2017, pp. 1602-1607.
- [20] J. Li, X. Mei, D. Prokhorov and D. Tao, "Deep Neural Network for Structural Prediction and Lane Detection in Traffic Scene," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 690-703, March 2017.
- [21] K. Behrendt and J. Witt, "Deep learning lane marker segmentation from automatically generated labels," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, pp. 777-782.

- [22] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey and V. N. Murali, "DeepLanes: End-To-End Lane Position Estimation Using Deep Neural Networks," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, 2016, pp. 38-45.
- [23] Jihun Kim, Jonghong Kim, Gil-Jin Jang, Minhoo Lee, Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection, In Neural Networks, Volume 87, 2017, Pages 109-121, ISSN 0893-6080.
- [24] Aly, Mohamed. "Real time detection of lane markers in urban streets." *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008.
- [25] Li, Z.; Cai, Z.-X.; Xie, J.; Ren, X.-P. Road Markings Extraction Based on Threshold Segmentation. In Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery, Chongqing, China, 29–31 May 2012; pp. 1924–1928.
- [26] Kheyrollahi, A.; Breckon, T.P. Automatic Real-Time Road Marking Recognition Using a Feature Driven Approach. *Mach. Vis. Appl.* 2012, 23, 123–133.
- [27] U. Ozgunalp and N. Dahnoun, "Lane detection based on improved feature map and efficient region of interest extraction," 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Orlando, FL, 2015, pp. 923-927.
- [28] Z. Ying and G. Li, "Robust lane marking detection using boundary-based inverse perspective mapping," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016, pp. 1921-1925.
- [29] Tsung-Ying Sun, Shang-Jeng Tsai and V. Chan, "HSI color model based lane-marking detection," 2006 IEEE Intelligent Transportation Systems Conference, Toronto, Ont., 2006, pp. 1168-1172.
- [30] Jyun-Guo Wang, Cheng-Jian Lin, Shyi-Ming Chen, Applying fuzzy method to vision-based lane detection and departure warning system, In Expert Systems with Applications, Volume 37, Issue 1, 2010, Pages 113-126, ISSN 0957-4174.
- [31] P. Chanawangsa and C. W. Chen, "A New Color-Based Lane Detection Via Gaussian Radial Basis Function Networks," 2012 International Conference on Connected Vehicles and Expo (ICCVE), Beijing, 2012, pp. 166-171.
- [32] Hoang, T.M.; Baek, N.R.; Cho, S.W.; Kim, K.W.; Park, K.R. Road Lane Detection Robust to Shadows Based on a Fuzzy System Using a Visible Light Camera Sensor. *Sensors* 2017, 17, 2475.
- [33] J. W. Lee and J. S. Cho, "Effective Lane Detection and Tracking Method Using Statistical Modeling of Color and Lane Edge-Orientation," 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, Seoul, 2009, pp. 1586-1591.
- [34] Hoang, T.M.; Hong, H.G.; Vokhidov, H.; Park, K.R. Road Lane Detection by Discriminating Dashed and Solid Road Lanes Using a Visible Light Camera Sensor. *Sensors* 2016, 16, 1313.
- [35] Cha-Keon Cheong, "Design of lane detection system based on color classification and edge clustering," 2011 3rd Asia Symposium on Quality Electronic Design (ASQED), Kuala Lumpur, 2011, pp. 266-271.
- [36] Forsyth, David, and Jean Ponce. *Computer vision: a modern approach*. Upper Saddle River, NJ; London: Prentice Hall, 2011.
- [37] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer, Inverse perspective mapping simplifies optical flow computation and obstacle detection, *Biological Cybernetics* 64, pp. 177-185, 1991.
- [38] Serge Belongie & David Kriegman (2007) *Explanation of Homography Estimation* from Department of Computer

Science and Engineering, University of California, San Diego.

- [39] H. Yoo, U. Yang and K. Sohn, "Gradient-Enhancing Conversion for Illumination-Robust Lane Detection," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1083-1094, Sept. 2013.
- [40] Stephens, Richard S. "Probabilistic approach to the Hough transform." *Image and vision computing* 9.1 (1991): 66-71.