Real-Time Lane Detection-Based Line Segment Detection

Ahmed Mahmoud[1], Loay Ehab[1], Mohamed Reda[1], Mostafa Abdelaleem[1],
Hossam Abd El Munim[2], Maged Ghoneima[3], M. Saeed Darweesh[4,5], and Hassan Mostafa[1,5]
[1]Electronics and Electrical Communications Engineering Department, Faculty of Engineering, Cairo University, Egypt
[2]Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University
[3]Mechatronics Engineering Department, Ain-Shams University, Egypt
[4]Electronics and Communications Engineering Department, Institute of Aviation Engineering and Technology, Egypt
[5]Nanotechnology Department, Zewail City of Science and Technology, Egypt

***Abstract*—This paper introduces a robust algorithm for real-time lane detection using the lane markers in urban streets or highway roads. It is based on applying Region of Interest (ROI) on the input image of the road from a calibrated camera in the front of the car, generating the top view of the image using Inverse Perspective Mapping (IPM), applying the core algorithm Line Segment Detection (LSD) which is followed by post-processing steps. Applying curve fitting to the line segments to get the right and left lines or curves. Finally, to get the output stream inverse IPM is applied. The proposed algorithm can detect the road lanes discriminating dashed and solid road lanes, straight and curved road lanes overcoming the shadow effect challenge with real-time performance 70 frames per second.**

Keywords—Lane detection, Lane keeping assist, Computer vision, Autonomous vehicles, Lane departure warning systems.

I. INTRODUCTION

Nearly 1.25 million people die in road crashes each year, on average 3,424 deaths a day. Up to 90% of accidents are due to human factor [1]. According to the national transportation safety board (NTSB) nearly 10 % of the car accidents in the US caused by wrong lane departure which equates to over 530,000 accidents each year [2]. Therefore, many advanced driver assistance system (ADAS) have been developed to help drivers to drive safely. Lane detection is one of the key features of the advanced driver assistance system. Lane detection process has several major challenges, such as attaining robustness to inconsistencies in lighting and background clutter. In this paper, several image processing, computer vision algorithms will be implemented to address these challenges.

Lane detection approaches can be divided into two categories: sensor-based approaches and vision-based approaches. Sensor-based approaches use devices such as light imaging detection and ranging (LIDAR), radar, laser sensors, and even global positioning systems (GPS) [3-4]. While vision-based approaches using camera system can be divided into two categories: model-based methods which create a mathematical model of the road structure using the geometric coordinates of the camera and the road as input parameters and feature-based methods which can distinguish lane marker points of lane markings from the non-lane areas by characteristic features of the road, such as color, gradient, or edge. In the presented work in [5], center line of the left and right boundaries of a road lane was only detected. On the other hand, the proposed algorithm can detect the accurate left and right boundaries of a road lane. The adaptive region of interest was used consuming high process time while the fixed region of interest consuming no time is used [6-7].

In this paper, a robust and effective lane detection algorithm is proposed using feature-based methods. First, region of interest (ROI) is applied to the input image from the camera to neglect the non-road region. Second, inverse perspective mapping (IPM) is applied to the output frame from ROI to get the birds eye view. Third, the core algorithm, line segment detection (LSD) is applied to get all line segments in the image which is followed by filtration to remove the false line segments. For connecting the line segments to get the detected lane, curve fitting method is applied. Finally, the inverse IPM is performed to generate the output stream.

The paper is organized as follows: Section II gives a detailed description of the proposed algorithm. Section III shows the results and discussion. Finally, conclusion is given in Section IV.

II. PROPOSED LANE DETECTION ALGORITHM

The proposed algorithm is divided into main processing steps ROI, IPM, LSD and post-processing steps (Filtration, Curve fitting, Output stream). Each part will be discussed in the following subsections.

A. Region of Interest (ROI)

The first step in the proposed algorithm is taking the part of the image that contains the information necessary to detect the lanes accurately.
The image of the road is divided into road region and non-road region. The role of this stage is to take only the part of the image where the road lanes exist the road region part of the image. In the proposed algorithm, we use a fixed ROI which has two benefits:

1. Focus the attention only on a subregion of the input image, which helps in reducing the runtime considerably.
2. It is the most suitable for the real-time application.

B. Inverse Perspective Mapping (IPM)

The second step in proposed algorithm is to generate a top view of the road image [8]. This will get rid of the perspective effect in the image, and so lanes that appear to converge at the horizon line are now vertical and parallel. This uses the main assumption in this paper that the lanes are parallel (or close to parallel) to the camera.

To get the IPM of the input image, we assume a flat road, and use the camera focal length, optical center, pitch angle, yaw angle, and height above ground parameters to perform this transformation. Starting by defining a world frame F_w as $\{F_w\} = \{x_w, y_w, z_w\}$ which centered at the camera optical center, a camera frame $\{F_c\} = \{x_c, y_c, z_c\}$, and an image frame $\{F_i\} = \{u, v\}$ as shown in Fig. 1.

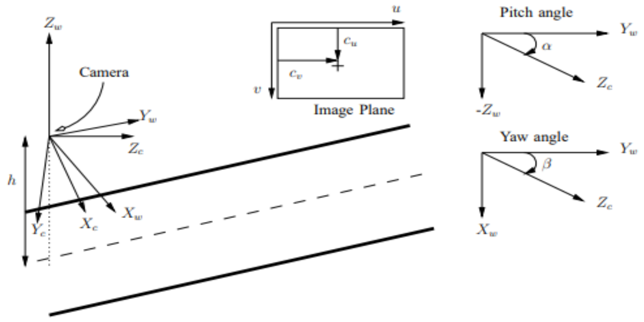

Fig. 1. IPM coordinates. Left: the coordinate axes (world, camera, and image frames). Right: definition of pitch α and yaw β angles.

We assume that the camera frame X_c axis stays in the world frame $X_w Y_w$ plane i.e. we allow for a pitch angle α and yaw angle β for the optical axis but no roll. The height of the camera frame above the ground plane is h. starting from any point in the image plane${}^{i}p = \{u, v, 1, 1\}$, it can be shown that its projection on the road plane can be found by applying the homogeneous transformation as formula (1).

$$ {}^{g}_{i}T = h \begin{bmatrix} \frac{-1}{f_u} c_2 & \frac{1}{f_v} s_1 s_2 & \frac{1}{f_u} c_u c_2 - \frac{1}{f_v} c_v s_1 s_2 - c_1 s_2 & 0 \\ \frac{1}{f_u} s_2 & \frac{1}{f_v} s_1 c_1 & -\frac{1}{f_u} c_u s_2 - \frac{1}{f_v} c_u s_1 c_2 - c_1 c_2 & 0 \\ 0 & \frac{1}{f_u} c_1 & -\frac{1}{f_v} c_v c_1 + s_1 & 0 \\ 0 & \frac{1}{h f_u} c_1 & \frac{1}{h f_v} c_v c_1 - \frac{1}{h} s_1 & 0 \end{bmatrix} \quad (1) $$

When ${}^{g}p = {}^{g}_{i}T\,{}^{i}p$ it will be the point on the ground plane corresponding to ${}^{i}p$ on the image plane, where $\{f_u, f_v\}$ are the horizontal and vertical focal lengths, respectively, $\{c_u, c_v\}$ are the coordinates of the optical center, and $c_1 = cos\,\alpha$, $c_2 = cos\,\beta$, $s_1 = sin\,\alpha$, and $s_2 = sin\,\beta$. These transformations can be efficiently calculated in matrix form for hundreds of points. The inverse of the transform can be easily found using formula 2.

$$ {}^{i}_{g}T = \begin{bmatrix} f_u c_2 + c_u c_1 s_2 & c_u c_1 c_2 - s_2 f_u & -c_u s_1 & 0 \\ s_2(c_v c_1 - f_v s_1) & c_2(c_u c_1 - f_v s_1) & -f_v c_1 - c_v s_1 & 0 \\ c_1 s_2 & c_1 c_2 & -s_1 & 0 \\ c_1 s_{22} & c_1 c_2 & -s_1 & 0 \end{bmatrix} \quad (2) $$

Where starting again from a point on the ground ${}^{g}p = \{x_g, y_g, -h, 1\}$, subpixel coordinates can be found on the image frame by ${}^{i}p = {}^{i}_{g}T\,{}^{g}p$ and then rescale the homogeneous part. Using these two transformations, we can project a window of interest from the input image onto the ground plane. Fig. 2 shows a sample IPM image. The left side shows the original image with the region of interest in yellow, and the right image how the transformed IPM image.

Fig. 2. IPM sample. Left: input image with region of interest in yellow. Right: the IPM view

C. Line Segment Detection (LSD)

LSD is an algorithm used for detection of line segments in a photo and it stands on Helmholtz principle which states that no detection is perceived in noise which can be applied in the computer vision field as any object desired to be detected shall be combined with a geometrical event and its regularity is validated in the Contrario model in case of line segment this event is rectangle as line segment can be viewed as rectangle containing pixels with various orientations [9]. Some definitions should be known first like level line field which is a vector calculated to all the pixels of the image and its magnitude is the gradient of the pixel and its direction is from the dark side to the lighter side on the edge of transition from one pixel to another and its orientation is defined by an angle called the level line angle (LLA). Contrario model in this algorithm is defined as a photo with same dimensions of the original one where level line fields of pixels are defined as independent random variables and their level line angles have uniform distribution from 0 to 2π [10]. Steps of the algorithm are discussed as follows:

1. Image Scaling

Details in image perceived by the human eye differ when the image is seen at different scales and this happens exactly in the algorithm as line segments detected differ when they are validated at different scales which can cause what is known as staircase effect at which some line segments are falsely detected or not detected at all due to image scale as shown below in Fig. 3 so image scaling is the first stage of the algorithm and the scale used for image is 80% as this is the smallest which overcomes the staircase effect while also preserving most details of the photo. Image scaling is done using Gaussian subsampling after filtering the high frequencies in the image to avoid aliasing [9].

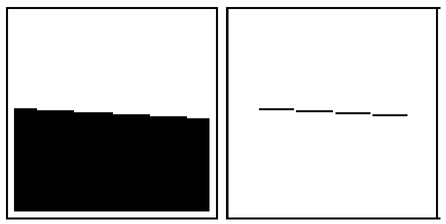

Fig. 3. Image before scale where line segment is detected as four small line segments

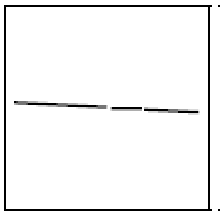

Fig. 4. Line segment detected correctly after using 80% scaling

2. Gradient Computation

The second step is gradient computation which is done by 2x2 mask which calculates the difference in intensities between a pixel and the neighboring pixels to it and we use 2x2 mask to reduce the dependencies among the level line fields magnitudes of the neighboring pixels to keep faithful to our assumption in the Contrario model that they are independent random variables. And pixel level line calculated from (5).

$$gx = \frac{i(x+1,y) + i(x+1,y+1) - i(x,y) - i(x,y+1)}{2} \quad (3)$$

$$gy = \frac{i(x,y+1) + i(x+1,y+1) - i(x,y) - i(x+1,y)}{2} \quad (4)$$

$$Level\ line\ angle = arctan\left(\frac{gx(x,y)}{-gy(x,y)}\right) \quad (5)$$

$$Gradientmagnitude = G(x,y) = (gx(x,y) + gy(x,y))^{1/2} \quad (6)$$

Then pixels are pseudo-ordered due to their gradient magnitudes calculated from (3), (4), (6) into quantized 1024 bins to optimize sorting in linear time and small gradient magnitudes are rejected because they introduce a large quantization error [9].

3. Region Growing and Rectangle Approximation

Region Growing is performed on ordered pixels where a seed pixel is chosen as the highest gradient magnitude unused pixel and if the absolute difference between its LLA and the neighboring ones LLA is within tolerance $\bar{\imath}$ it joins the region and the region angle is updated as the average angle of the region's pixels and the tolerance is a design parameter chosen by the designer and it shouldn't be too extreme and it was chosen in our implementation by the value $\pi/8$. After that, the region is approximated to a rectangle with its length and width are chosen such that they are the smallest dimensions which cover the region and the rectangle orientation is calculated by interpreting the rectangle as a solid object and the gradient magnitudes of its pixels as weight of its points and the orientation of the rectangle is calculated as the direction of the first axis of inertia of the object [9].

4. Validation by Calculating NFA

Each rectangle is validated by calculating the probability that a rectangle can be present by the same regular structure or more in the Contrario model. the regularity of the structure of rectangle is defined by the number of aligned points in it and aligned points are defined as the points whose level line angle similar to the rectangle orientation within certain tolerance so we can use binomial probability distribution to measure that probability which leads us to an exhausting number of tests as $(NM)^{5/2}$ as shown in the Fig. 5 where N and M are dimensions of photo after scaling.

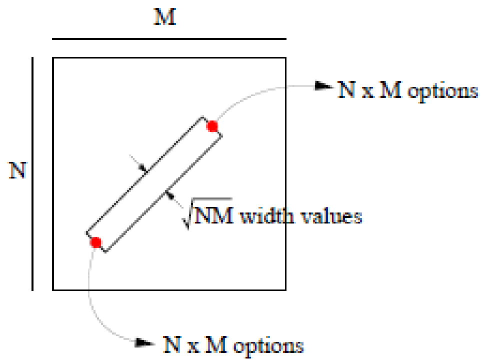

Fig. 5. Number of tests determination

Probability of having rectangle with same regular structure or more regular in the Contrario model defined as NFA (number of false alarms) = ((NM)^(2.5))*$\sum_{k=j}^{n} \binom{n}{j} * p^j * p^{1-j}$ where p is the probability where LLA of pixel with in tolerance $\bar{\imath}$ and it is calculated as $\frac{\bar{\imath}}{\pi}$ and the resulted NFA is compared by threshold set to 1 a as mentioned before. Finally, the algorithm outputs the starting and ending points of line segments in the photo and it works with complexity proportional to the dimensions of the photo.

D. Filtration

Line segments resulting from the algorithm are filtered due to their angle and distance also the angle of each line can be used to detect whether it belongs to the left side of the lane or the right side of the lane.

E. Curve Fitting using Second Order Polynomial Approximation

We use a second order polynomial approximation because it is most suitable for straight lanes and curved ones and we apply the second order polynomial approximation on the output point of the LSD stage to visualize the lane for the user. Let $\{(x, y_l)\}$ and $\{(x, y_r)\}$ be the set of data points for the left and right lane returned by LSD. These data points give a coarse estimate of the road boundaries which at times seem unrealistic.

As road boundaries are simple curves which can be safely approximated by a second order polynomial, we fit these data points to a polynomial of the form $y = a\,x^2 + bx + c$. The data points are fit to the curve by minimizing the sum of the squares of the offsets of the points from the curve. The best fitting curve *f(x)* has the least square error according to equation 7.

$$\sum_{i=1}^{n}[\,y_i - f(x_i)]^2 = \sum_{i=1}^{n}[\,y_i - (a\,x^2 + bx + c)]^2 = minimum \quad (7)$$

Where a, b and c are unknown coefficients and x_i are the known row positions and y_i the corresponding column positions. To obtain the least square error, the unknown coefficients a, b and c must yield zero first derivatives [11].

$$\left\{\begin{aligned} \frac{\partial \Pi}{\partial a} &= 2\sum_{i=1}^{n} x_i^2 [\, y_i - (a\, x_i^2 + b x_i + c)] \; = 0 \\ \frac{\partial \Pi}{\partial b} &= 2\sum_{i=1}^{n} x_i [\, y_i - (a\, x_i^2 + b x_i + c)] \; = 0 \\ \frac{\partial \Pi}{\partial c} &= 2\sum_{i=1}^{n} [\, y_i - (a\, x_i^2 + b x_i + c)] \; = 0 \end{aligned}\right\} \tag{8}$$

By expanding the equation 8, we get equation 9:

$$\left\{\begin{aligned} \sum_{i=1}^{n} x_i^2 y_i &= a\sum_{i=1}^{n} x_i^4 + b\sum_{i=1}^{n} x_i^3 + c\sum_{i=1}^{n} x_i^2 \\ \sum_{i=1}^{n} x_i y_i &= a\sum_{i=1}^{n} x_i^3 + b\sum_{i=1}^{n} x_i^2 + c\sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} y_i &= a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i + c\sum_{i=1}^{n} 1 \end{aligned}\right\} \tag{9}$$

The unknown coefficients can be obtained by solving this linear equation once the coefficients of the polynomial are estimated, we can have a smooth and realistic estimate for the road lanes.

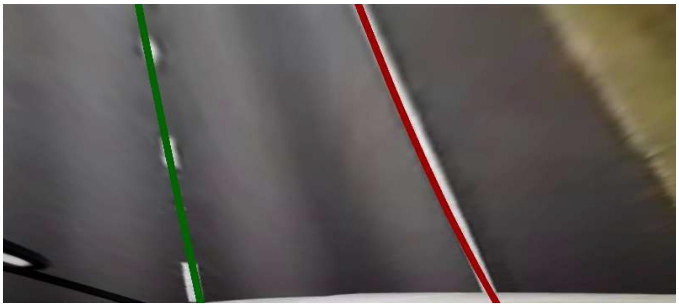

Fig. 6. Smooth road lanes extracted after fitting a second order polynomial

F. Multiple Frames

Detection based on multiple frames was introduced to overcome the problem of no detection in some frames and to increase smoothness of the lines drawn after detection, the input to the curve fitting is not just the output of the current frame from LSD block, instead detected lines from multiple previous frames (up to five frames) is stored and added to the detection of current frame to run curve fitting on.

This method allowed more robust, smooth and continuous detection without the cost of additional computational power, it also can be introduced as a lane tracking approach without the usage of additional algorithm (i.e. Kalman Filter) and the cost of excess runtime caused by it.

Table 1. Runtime on both PC and TX1

	CPU: i7, 2.6GHZ, 16GB RAM	**Nvidia Jetson TX1**
Frame execution time (ms)	13.7491	60
Average frame per second	72.72	16.66

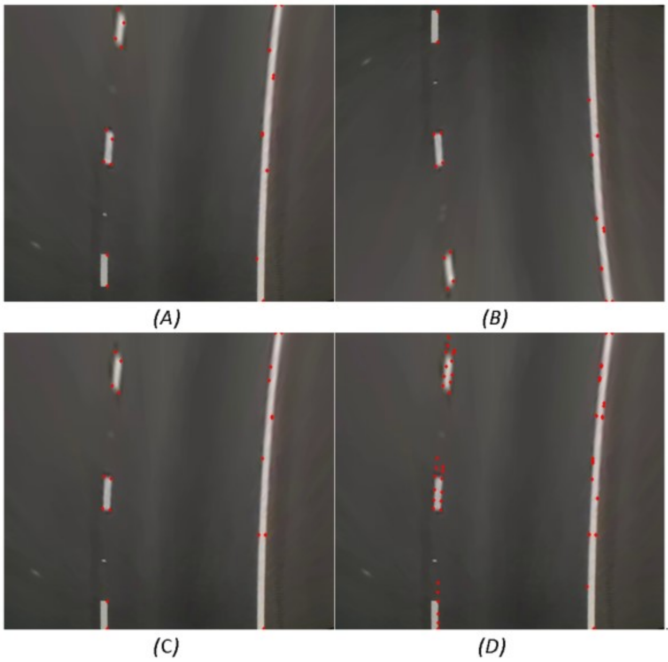

Fig. 7. Detection based on multiple frame

Fig. 7(A) shows output of LSD block for one frame as red dots are the start and end of every detected line. Fig. 7(B) shows detected lines of next frame. Fig. 7(C) is the detected lines of third frame. Then all detected lines from the three frames are accumulated to be the input of curve fitting block as shown in Fig. 7(D).

III. RESULT AND DISCUSSION

Fig. 8(A) shows input image from camera. Fig. 8(B) shows ROI selection to reduce the processing time by processing a smaller image. Fig. 8(C) is the result after IPM. Then the detected lines by LSD algorithm is drawn by the points of start end and of each line in Fig. 8(D). In Fig. 8(E) lines are filtered to allow only lines with angles that can be part of lane. Fig. 8(F) is after drawing lines resulted from curve fitting based on multiple frames. Fig. 8(G) original view of camera frame with the drawn lines after inverse IPM. Fig. 8(H) shows lane keeping assist feature guiding arrows to keep driver in the center of the lane.

The proposed technique is evaluated on two environments which are a computer platform with an Intel i7: 2.6GHz core and 16GB of memory and Nvidia Jetson TX1 and the runtime is shown in Table 1.

The execution time of each block in the algorithm was evaluated on Nvidia TX1 as listed in Table 2.

Table 2. Time of every block on TX1

Block	**Execution time**
ROI and IPM	8.326 ms
LSD	33.5048 ms
Filtration	1.7541 ms
Curve fitting	0.452 ms
Inverse IPM	17.6404 ms

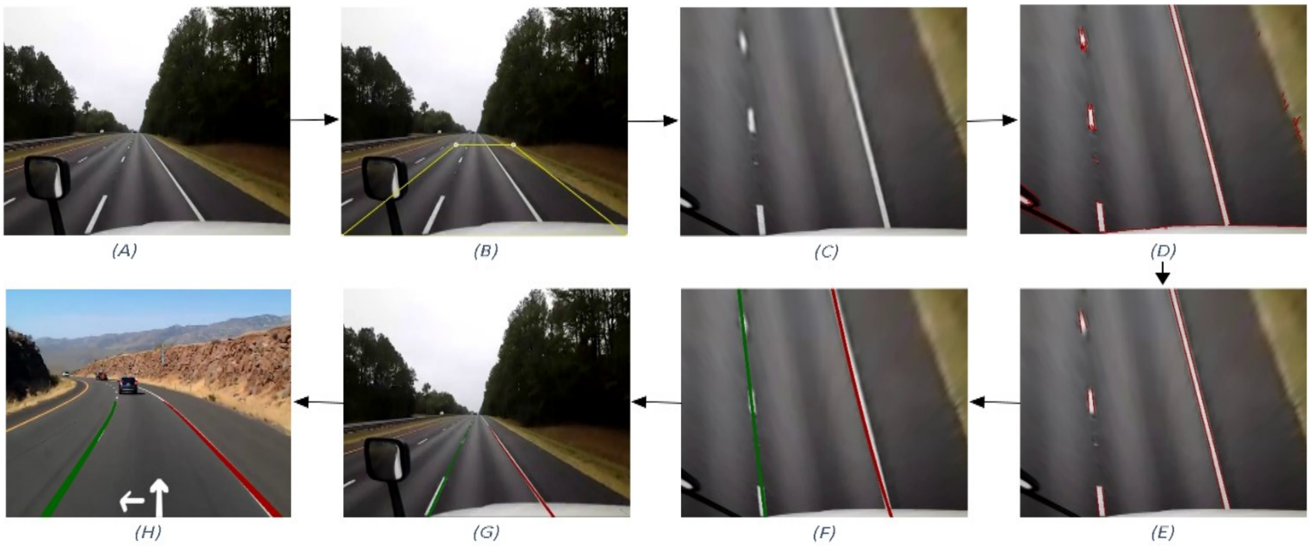

Fig. 8. Proposed land detection algorithm results

Two features were proposed after detection which are to visually differentiate between solid and dashed lines and lane keeping assist.

For differentiating between dashed and solid lines a condition was applied on the results of LSD blocks such as, if the average length of lines in one region (Right or Left) exceeded a certain threshold then this line is solid and colored with red color, else if it is a dashed line and colored with green to indicate possibility of changing lane through that line.

Lane keeping assist was achieved by measuring the distance (in pixels) between the center of the lane and both right and left lines. If the driver deviates to one side with more than a defined threshold (in pixels), a guiding arrow is displayed to direct driver back in the center of lane.

IV. CONCLUSION

In this paper, we proposed a real-time lane detection method based on line segment detection. Detection based on multiple frames was proposed to solve the problem of no detection and false curvature of lane lines due to small detected lines in some frames. Also, introduces a more stable drawn lines of the lane as it counts on both current and previous detections.

ACKNOWLEDGMENT

This work was partially funded by Mentor Graphics, and ONE Lab at Cairo University and Zewail City of Science and Technology.

REFERENCES

[1] Accidents https://en.wikipedia.org/wiki/List_of_countries_by_traffic-related_death_rate, 2018.

[2] NTSB, https://www.ntsb.gov/Pages/default.aspx, 2018.

[3] M. Yousef, A. Hosny, W. Gamil, M. Adel, H. M. Fahmy, S. M. Darweesh, and H. Mostafa, "Dual-Mode Forward Collision Avoidance Algorithm Based on Vehicle-to-Vehicle (V2V) Communication", in *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS 2018)*, Windsor, Ontario, Canada, pp. 739-742, 2018.

[4] H. Fahmy, G. Baumann, M. AbdelGhany, and H. Mostafa, "V2V-Based Vehicle Risk Assessment and Control for Lane-Keeping and Collision Avoidance", in *IEEE International Conference on Microelectronics (ICM 2017)*, Beirut, Lebanon, pp. 61-65, 2017.

[5] Dajun Ding, Chanho Lee, Kwang Lee, "An Adaptive Road ROI Determination Algorithm for Lane Detection," in *Proceedings of the TENCON 2013–2013 IEEE Region 10 Conference*; pp. 22–25, 2013.

[6] Pallavi V. Ingale and Prof. K. S. Bhagat, "Comparative Study of Lane Detection Techniques," in *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 4, no. 5, 2016.

[7] Ammu M Kumar and Philomina Simon, "Review of lane detection and tracking algorithms in advanced driver assistance system," in *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 7, no. 4, 2015.

[8] Mohamed Aly, "Real Time Detection of Lane Markers in Urban Streets," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 4–6, 2008.

[9] Von Gioi R.G., Jakubowicz J., Morel J.-M., Randall G. LSD, "A line segment detector," in *Image Processing Line*, vol. 2, pp. 35–55, 2012. doi: 10.5201/ipol.2012.gjmr-lsd.

[10] Toan Minh Hoang, Hyung Gil Hong, Husan Vokhidov and Kang Ryoung Park, "Road Lane Detection by Discriminating Dashed and Solid Road Lanes Using a Visible Light Camera Sensor," in *Division of Electronics and Electrical Engineering*, pp. 15-18, 2016.

[11] Wang Y., Shen D., and Teoh E, "Lane detection using spline model," in *Pattern Recognition Letters*, vol. 21, pp. 677-689, 2000.