Chapter

Game Theory-Based Autonomous Vehicle Control Via Image Processing

Mustafa F.S. Zortul, Tayyab Waqar and Sezgin Ersoy

Abstract

Self-driven vehicles slowly but surely are making the transition from a distant future technology to current luxury by slowly becoming a part of our everyday life. Due to their self-driving ability, they are making our travels efficient. However, they are still a work in progress as they require many software and hardware-based improvements. To address the software part of this issue, an image processing-based solution has been proposed in this study. The algorithm estimates the real-time positions and predicts the possible interaction of the objects, such as other moving vehicles, present in the vicinity of the driven autonomous vehicle in determined environmental conditions. Cameras and related peripheral present on autonomous vehicles are used to obtain data related to the real-time situation for predicting and preventing possible hazardous events, such as accidents, using these data.

Keywords: autonomous drive, game theory, vehicle control, image processing

1. Introduction

The continuous advancements in technology are constantly making our life more and more efficient. To this end, autonomous vehicles are one of the important pillars according to FIA [1] and the reports from IPSOS also show that the trends will be in the favor of the adaption of autonomous vehicles once the standardization issues are addressed [2].

Autonomous vehicles use sensors, such as radar, lidar, and camera, to implement computer vision and odometry techniques for sensing the surrounding environment and make the travel without any human intervention possible [3]. When we look into the history of autonomous vehicles, we see that first autonomous projects appear in the 1980s [4]. The first prototype came to life with navlab and ALV projects that ran by Carnegie Mellon University [4]. These prototypes were followed by project Eureka Prometheus by Mercedes-Benz and Bundeswehr partnership in 1987 [4, 5]. After these, many companies have manufactured autonomous vehicles and some of those vehicles have also found a place in active traffic in some countries. It is believed that in the near future, autonomous vehicles will cause an unprecedented change in economic, social, and environmental areas of life [6].

However, still there remains work to be done before the roads are filled with safe self-driving cars. Researches have been done to make the autonomous vehicles safer and environment friendly so that they can be as close to perfect as possible. A real-time lane detection system for autonomous vehicles is proposed by Abdulhakam A. M. Assidiq et al. utilizing the images captured by the onboard camera installed on a moving vehicle to perform real-time lane detection the lane via image processing using edge detection and Hough Transform [7]. Yen-Lin et al. presented a night-time vehicle detection method for two vehicles, incoming and outgoing, using their head and tail lights [8]. They used image segmentation and spatial clustering processes to perform rule-based vehicle detection and distance estimation by using the head and tail lights of the vehicle. Image processing techniques have also been applied to detect the current state of traffic light by Guo et al. They used Histogram of Oriented Gradient (HOG) features to implement Support Vector Machine (SVM) for the real-time detection of the current state of traffic lights for self-driving vehicles [9]. Phanindra et al. presented a method for detecting obstacles and lane using the data collected by a LIDAR sensor and a fisheye camera [10]. A vector fuzzy connectedness-based algorithm has been proposed by Lingling Fang et al. to detect the boundary of the lane using the captured images by the camera [11].

Based on these results, in this study, an image processing software has been proposed that can autonomously extract data related to factors of the vehicle environment. The predictions regarding the self-driving vehicle's environment are carried out on the basis of the proposed three models which are built using the video footage taken from an onboard camera placed inside the vehicle. The first model is an SVM that can detect vehicle positions using the inputted image. The second model fits a linear-parabolic function for the right and left lane boundaries of the vehicle. The third model approximates a bird-view version of the inputted image and carries out a more realistic approximation of steer angle on the bird-view image, rather than original.

2. Methods: research and application steps

2.1 Detection and tracking of lane boundaries

The model developed in this study is designed to work in a highway environment. Model performance is invariant to the color of the road and is directly proportionate to the amount of contrast between the road and lane boundary color, quality, and continuity of lane boundaries. However, the model assumes that the vehicle which is guided is in the center of the lane that the vehicle is present according to the horizontal plane and conducts its operations on this assumption. The rest of this subsection covers the following topics: the detection of the lane boundaries, the approximation of edge distribution function, weighted Hough transform which is used to determine the location of lane boundaries in images and lastly description of both linear and parabolic functions that we approximated for every single lane boundary. The images used in this study are taken by a vehicle-centered camera [12], and all studies are continued iteratively as a development and test loop on these images. The size of the image array (720, 1280) and different environmental aspects that it contains are shown below (**Figure 1**).

2.2 Initial detection of lane boundaries

In the first image of the application, we aimed to calculate the slopes of right and left lane boundaries in the image according to the vehicle-centered camera view. Firstly, image is converted to grayscale form and with the help of Sobel filter vertical and horizontal gradient images are obtained. These gradient images are

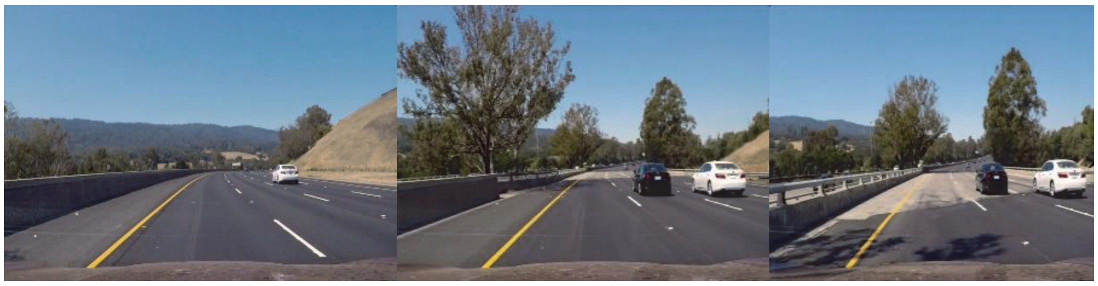


Figure 1.
Image samples used in the study. Images contain distortions that are caused by lane boundaries of various colors, different environmental aspects, and change of road color that is caused by various light conditions and color differences which all push the model to make errors.

then used to form a magnitude matrix and orientation matrix which contains slope calculation of every pixel in the image. With these matrices, an edge distribution function is formed, and local maxima points of edge distribution function gave us the center-lane boundary slopes on the image. Hough transform is applied with known edge slopes, and global maxima points for every edge are found with weighted voting. The locations of lane boundaries are found after slope and distance from origin calculation.

2.3 Edge distribution function (EDF)

To decrease the computational complexity, images are converted to grayscale. Function (30) is used for this conversion. The grayscale image is named as $I(x, y)$. With vertical and horizontal differentials of this function, we obtain $\nabla I(x, y)$ function [13].

$$\nabla I(y, x) = \left(\frac{\partial I}{\partial y}, \frac{\partial I}{\partial x}\right)^T = \left(D_y, D_x\right)^T \tag{1}$$

D_x and D_y represent vertical and horizontal differentials, respectively. D_x and D_y are used to calculate the magnitude matrix which contains a magnitude of every pixel in the image. Eq. (2) is used for that.

$$|\nabla I(y, x)| = |D_x| + |D_y| \tag{2}$$

After this D_x and D_y matrices are used to calculate the orientation matrix which contains orientation of every pixel in the image. In the calculation of the orientation matrix, a lookup table is used to decrease arctangent values.

$$\theta(\mathrm{y}, \mathrm{x}) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{3}$$

A voting routine in which every $I(i, j)$ pixel votes weighted as their magnitude $|\nabla I(i, j)|$ took place. After voting, a histogram of 90 bins with $(-90, 90)$ minimum, maximum values are created.

For extracting local maxima points in the histogram, we first plotted the histogram as a piecewise function. Then smoothed the plot with a one-dimensional Gaussian filter (**Figure 2**). Filter width of 7 seemed enough to eliminate false positives after some number of tests we did. However, we also tried kernel density estimation to smooth the histogram. But it caused some meaningful local maxima points to shift from its original value or completely disappear, so we removed kernel density estimation from the workflow.

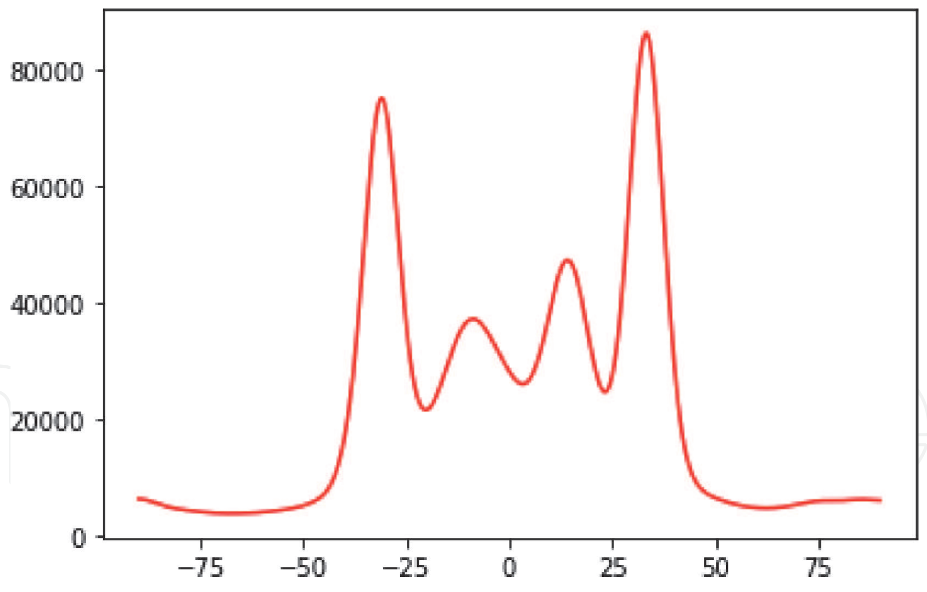
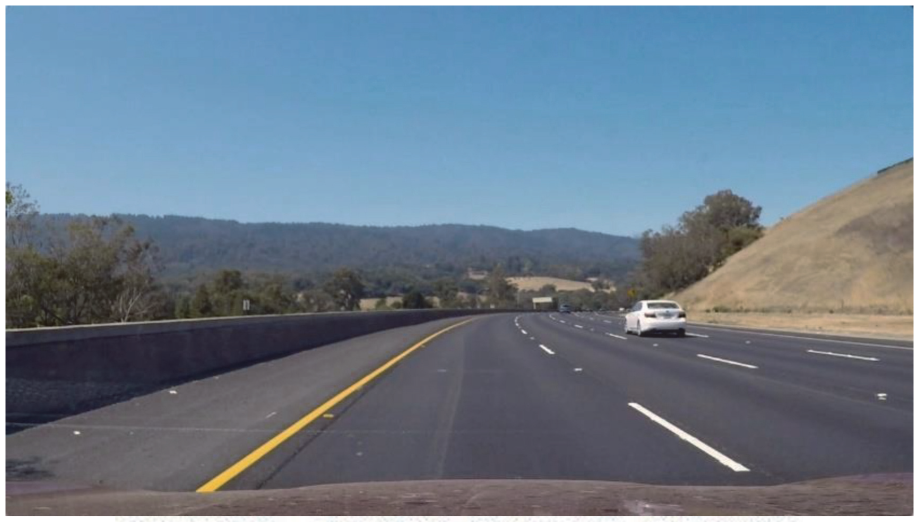


Figure 2.
Smoothed edge distribution function. Pixels between (470, 650) in the y-axis are assumed as the image part that contains the road from a (720, 1280) sized original image and converted to grayscale. Sobel filter is applied and magnitude and orientation matrices are obtained. Weighted voting took place, results smoothed with 1-d Gaussian filter.



```
[-30.8356546   -8.77437326  14.28969359  33.84401114]
```

Figure 3.
The image that edge distribution function generated from and local maxima points. Local maxima points on edge distribution function are detected and within these points, the possibility of being lane boundaries of the center lane is found as a result of thresholding.

When lane and road boundaries are considered as only linear objects in the image, we can assume that there will be two local maxima that are visibly voted higher than others. The situation on roads with multiple lanes can slightly differ from this and more local maxima points can appear in the edge distribution function. **Figure 2** above is generated from a multiple lane road. Below we discussed the local maxima points from this function (**Figure 3**).

When we consider that degrees shown above represent the degree value of horizontal axis lane boundaries, we can see that −30.8356546 is the right boundary of the center lane, −8.77437326 is the right boundary of the right lane, 14.28969359 is left road boundary, and 33.84401114 is left boundary of the center lane.

Generally, the center orientation of the center lane boundaries will be numerically closer to the vertical axis and have approximately the same absolute values.

To extract center lane boundary orientations and to eliminate false positive local maxima points which can be caused by organic variations in the road, we applied thresholding. α_1 and α_2 are minimum and maximum degree values from local maxima points, respectively. We applied thresholding to these values with Eq. (1). 15, 20, 25 are tested as threshold values and we saw that 20 shows the best performance.

$$|\alpha_1 + \alpha_2| \leq T_1 \tag{4}$$

The α_1 and α_2 couples that do not meet the requirements of this equation are removed from the local maxima list and we continue to apply the equation to the rest of the values. When there are no local maxima points left in the list, the orientation couple that closest to the vertical axis are selected as center lane boundary orientations. But it should be noted that in most cases, local maxima points of edge distribution function contain true positives only and this process is just a formality.

At this point, despite we know the orientations of center lane boundaries, we still do not know the exact locations of these boundaries. To find exact locations, we first threshold the magnitude image according to Eq. (5).

$$g(y,x) = f(x) = \begin{cases} |\nabla I(y,x)|, & \mathit{if}\ |\theta(y,x) - \alpha| < T_2 \\ 0, & \mathit{else} \end{cases} \tag{5}$$

To find the exact locations, we first get indexes of the pixels that have the same orientation as each center lane boundaries, then we zero out all the pixels except these indexes and pixels within a minor threshold value. By the end of this process, we have a magnitude matrix for each center lane boundary (**Figure 4**). As a result, we found two-lane boundary magnitude matrices and we will proceed to find exact locations of boundaries from these matrices.

2.4 Hough transform

Applying Hough transform on a two-dimensional matrix gives us a two-dimensional C (ρ, θ) accumulator matrix just as our input (**Figure 5**).

Slope and distance from image origin of found edges are obtained from the analysis of the accumulator matrix. But in this study, we already found the slopes of lane boundaries with techniques that we described in earlier subsections. Thus, we do not need to do our analysis on a two-dimensional matrix. As a result, our accumulator matrix will be independent of θ dimension and voting will take place only for potential ρ values which are only one dimension. Applying Gaussian filter

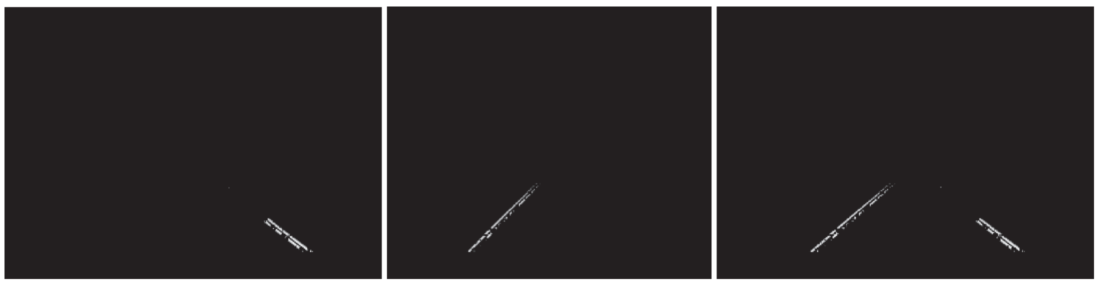


Figure 4.
Extracted edge images. These images are extracted from earlier images in this paper. A threshold is applied to a magnitude matrix to extract edge images. Two degrees is used as a threshold value in all tests. Centre lane boundaries are found with −30° (left), 33° (right). The image on the right is the sum of the edge images we found.

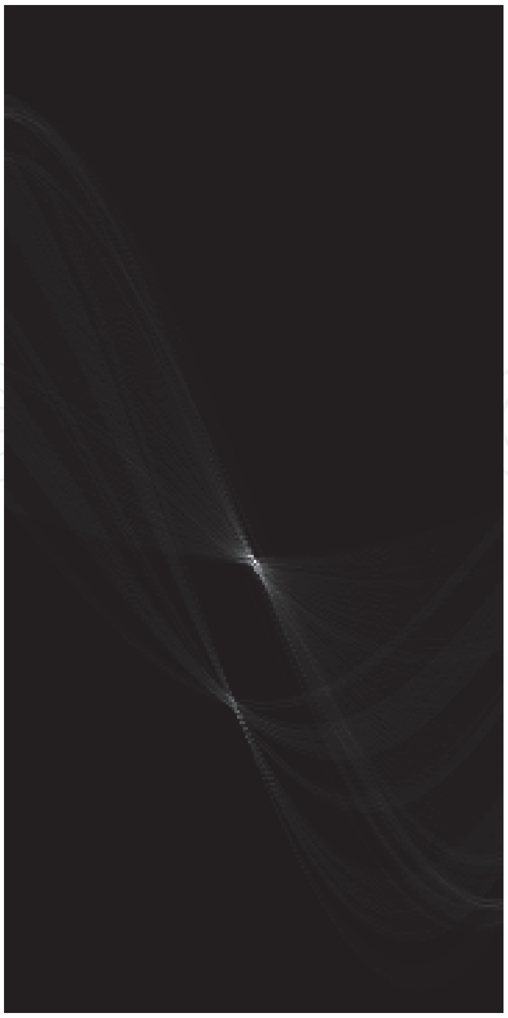


Figure 5.
Example Hough accumulator matrix. Hough transform is carried out as conversion of every non-zero pixel in the original image to Hough space (ρ, θ). Every pixel in the original image equals to a sinusoidal curve in Hough space. With representations of all pixels of the original image on the accumulator matrix, there will be local maxima points (ρ, θ) from straight lines of the original image. Thus, the locations of edges in the original image are found.

on these accumulator arrays will give us global maxima points. Gaussian filter windows size as 30 is enough in this part to eliminate multiple maxima points and to extract the true ρ value we are searching for which is the global maxima. After extracting global maxima points from each accumulator array, we now have the exact locations of center lane boundaries in the image. After we have exact locations, we will use these locations to specify a region of interest to be used in the next frame and edge search that we carried out will only take place in this region of interest in the next frame which will improve efficiency (**Figure 6**). Extension of 20 pixels to left and right is chosen as the width of the region of interest. We assume that lane boundary locations found in the last frame will not change drastically in the next frame of image array, this assumption is the main reason we specify a region of interest. Apart from that we also assumed that just as its location, orientation of lane boundary will not change drastically from the last frame processed to the next frame of the image array. So, we applied a threshold to the orientation search as well. We only searched in values that differ slightly from the last orientation we found. After some tests, we decided 4° as the threshold value is optimal.

2.5 Lane tracking

After the initial detection of lane boundaries from the first frame, lane detection should continue in following frames. A linear model is chosen for the initial frame.

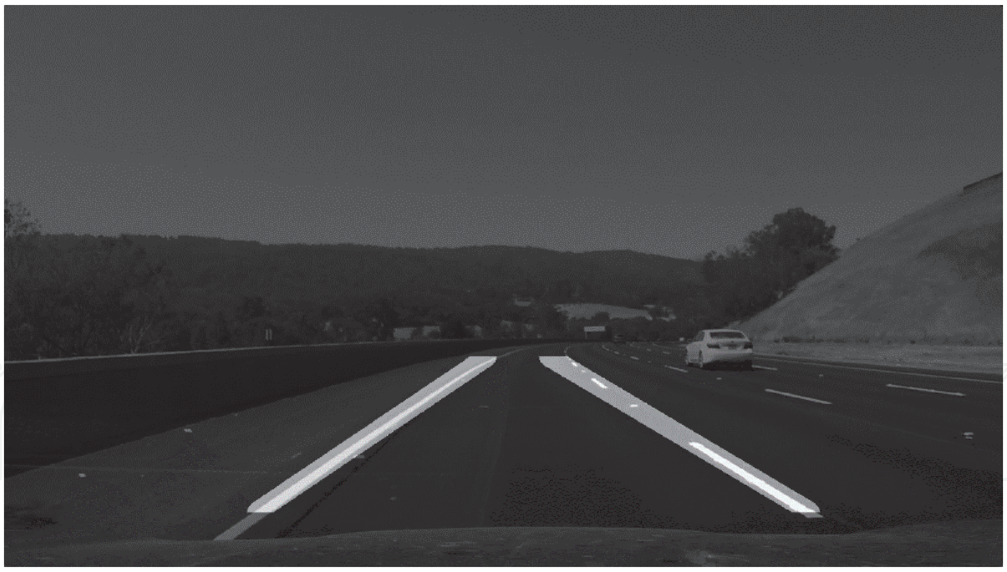


Figure 6.
Region of interest to be used in the next frame. After Hough transform, we have exact locations and orientations of center-lane boundaries. A region of interest is specified according to this information and the process is applied only within the region of interest. The image above is the sum of the region of interest and grayscale version of the original image.

But in the following frames, we already have prior information on lane boundaries which is the region of interest. So, we can work on a more complex model to predict, a linear-parabolic piecewise function is decided as a model. For this model, horizontal line that separates linear and parabolic parts of function is chosen as approximately 15 m in front of the vehicle and area below and above this boundary named as near and far sections, respectively. The model will act linear in the near section and will act parabolic in the far section. The mathematical representation of the model is shown below Eq. (6).

$$f(x) = \begin{cases} a + bx, & if\ x > x_m \\ c + dx + ex^2, & if\ x \leq x_m \end{cases} \tag{6}$$

The model needs to meet two equations to have continuity and differentiable features Eq. (7).

$$\begin{aligned} f(x_m^+) &= f(x_m^-) \\ f'(x_m^+) &= f'(x_m^-) \end{aligned} \tag{7}$$

Below equations were obtained when assumed requirements of Eq. (7) are met Eq. (8).

$$\begin{cases} a + bx_m = c + dx_m + ex_m^2 \\ b = d + 2ex_m \end{cases} \tag{8}$$

If equations in (8) are solved within themselves for c and e parameters piecewise function below is what we have Eq. (9):

$$f(x) = f(x) = \begin{cases} a + bx, & if\ x > x_m \\ \dfrac{2a + x_m(b-d)}{2} + dx + \dfrac{b-d}{2x_m}x^2, & if\ x \leq x_m \end{cases} \tag{9}$$

When we assume that both right and left center lane boundaries are found in the last frame, we can expect that both right and left boundaries will be in their corresponding region of interests in the next frame. Before we try to detect center lane boundaries in the next frame threshold application below should be applied within the region of interest to eliminate various noise sources that could appear in the region of interest Eq. (10).

$$g(y,x) = \begin{cases} |\nabla I(y,x)|, & if\ |\nabla I(y,x)| > 0.5 g_{avg} \\ 0, & else \end{cases} \tag{10}$$

We assume that usually, lane boundaries will have bigger values in magnitude matrices than any other source in the next frame and this assumption is the reason why we apply threshold within the region of interest. Hence, the majority of the pixels that above the threshold from Eq. (10) will belong to lane boundary we try to detect, and the majority of the pixels that below the threshold will be independent of lane boundary and can be considered as noise as a whole. As we remove these magnitudes from the magnitude matrix, we can carry out a more realistic approximation of the real lane boundary.

Let (x_{ni}, y_{ni}) be the near section non-zero pixels from threshold applied lane boundary magnitude matrices and let $\left(x_{fj}, y_{fj}\right)$ be the far section non-zero pixels from threshold applied lane boundary magnitude matrices. Also, let $M_{fj} = g\left(x_{fj}, y_{fj}\right)$ be the corresponding magnitude of matrices elements.

This situation gives us three unknown equation below.

$$\begin{cases} a + bx_{ni} = y_{ni}, & if\ i = 1, 2, \dots, m \\ \dfrac{2a + x_m(b-d)}{2} + dx + \dfrac{b-d}{2x_m} x_{fj}^2 = y_{fj}, & if\ j = 1, 2, \dots, n \end{cases} \tag{11}$$

A solution is approximated with the normal equation method for the equation above. The below function is used to represent error and tried to minimize it.

$$E = \sum_{i=1}^{m} M_{ni} \left[y_{ni} - f(x_{ni}) \right]^2 + \sum_{j=1}^{n} M_{fj} \left[y_{fj} - f\left(x_{fj}\right) \right]^2 \tag{12}$$

Function E could be written as a matrix multiplication below.

$$E = (b - Ac)^T W (b - Ac) \tag{13}$$

Variables which are stated in Eq. (13) are specified below.

$$\mathbf{A} = \begin{bmatrix} 1 & x_{n_1} & 0 \\ \vdots & \vdots & \vdots \\ 1 & x_{n_m} & 0 \\ 1 & \dfrac{1}{2x_m}\left(x_{f_1}^2 + x_m^2\right) & -\dfrac{1}{2x_m}\left(x_{f_1} - x_m\right)^2 \\ \vdots & \vdots & \vdots \\ 1 & \dfrac{1}{2x_m}\left(x_{f_n}^2 + x_m^2\right) & -\dfrac{1}{2x_m\left(x_{f_n} - x_m\right)^2} \end{bmatrix},$$

$$W = diag\left(M_{n1,\dots,}M_{nm}, M_{f1,\dots,}M_{fn}\right)$$

$$c = [a, b, d]^T\ and\ b = \left[y_{n1,\dots,}y_{nm}, y_{f1,\dots,}y_{fn}\right]^T$$

To solve the Eq. (13), we isolate variable b with matrix multiplication below.

$$A^TWAc = A^TWb \tag{14}$$

If we draw a plot with found parameters from Eq. (14), we get a straight line in near section and parabolic line in the far section as seen below (**Figure 7**).

2.6 Steer angle estimation

Steer angle estimation is prone to errors because as the distance from the camera increases the distortion of the road in the image also increases. For this reason, we first estimated the bird view version of the image with inverse perspective mapping (IPM) [14] (**Figure 8**).

We cropped the IPM image in a way that it only contains pixels from the center lane boundaries. Later we extract Canny edges from it (**Figure 9**).

After that, we apply Hough transform on the Canny image and get a two-dimensional accumulator matrix. Two highest local maxima points from the accumulator matrix give us the angle between the vertical plane and center lane boundaries. We averaged the two angle values and we found to estimate steer angle to keep the car on its lane.

2.7 Vehicle detection and tracking

We used a mix of GTI and KITTI datasets as the dataset. Dataset as a whole contains 8792 positive and 8968 negative and 17,760 examples in total. Every example is a (64, 64) size, RGB encoded, png image.

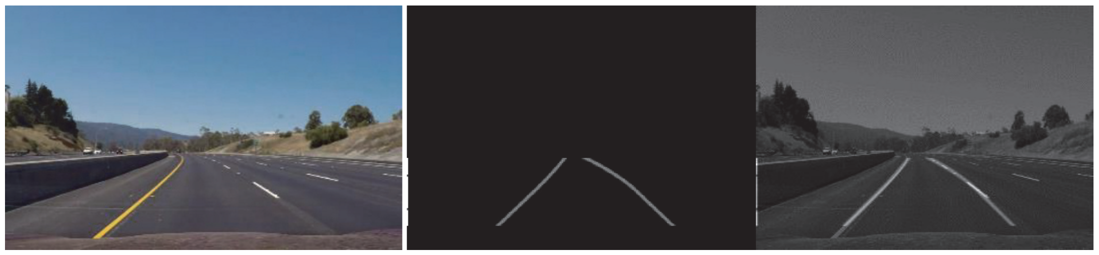


Figure 7.
Plot fitting example. Original image (left) went through the process covered in recent subsections. For found center lane boundaries, we fit a plot with the normal equation method which is covered in this subsection. Two-line image (middle) found as output and finally added to the grayscale version of the original image for comparison.

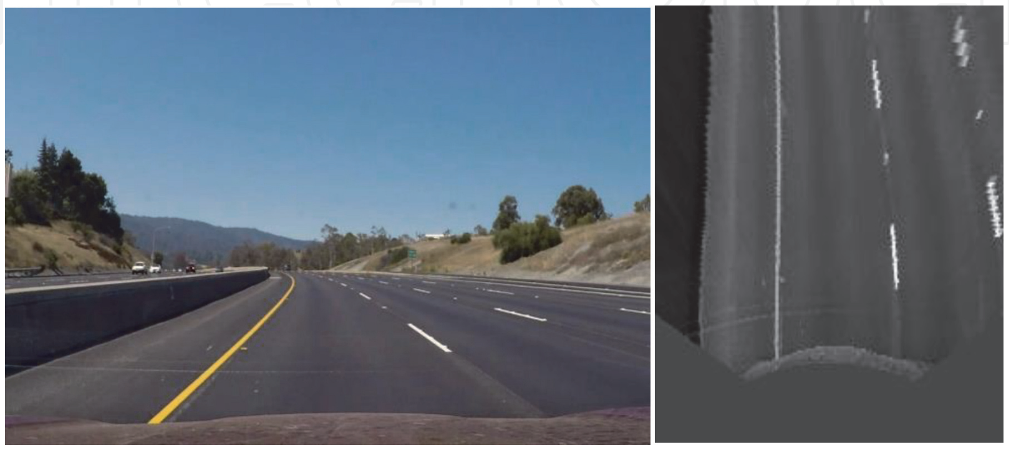


Figure 8.
Inverse perspective mapping example. Average camera parameters for bird view image of the original image (left) manually selected and bird view image (right) is found. The original image size is (720, 1280) and the bird view size is (200, 200).

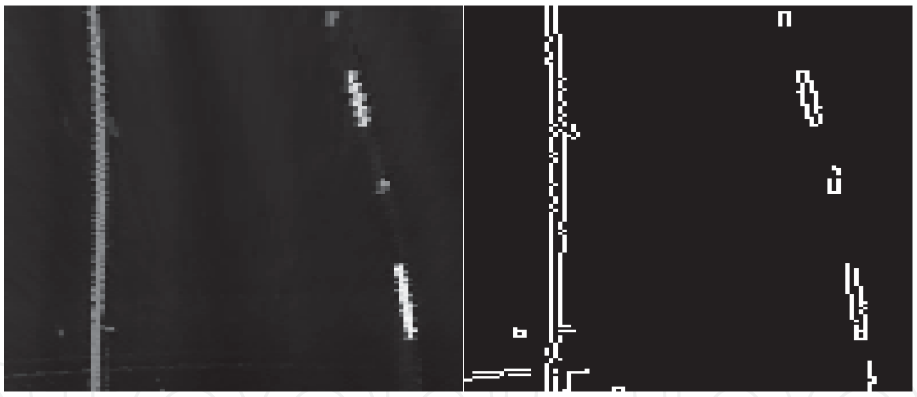


Figure 9.
Canny edges from bird view image.

First, we extracted histogram of oriented gradients in order to be used in training. Histogram of oriented gradients basically puts a grid on image. A magnitude weighted voting takes place for each bin. Then, a histogram is created from voting results.

To shrink the histogram of oriented gradients features space, the principal component analysis is applied. Then the support vector machine is trained. After training, images size of (720, 1280) scanned for vehicle detection with a sliding

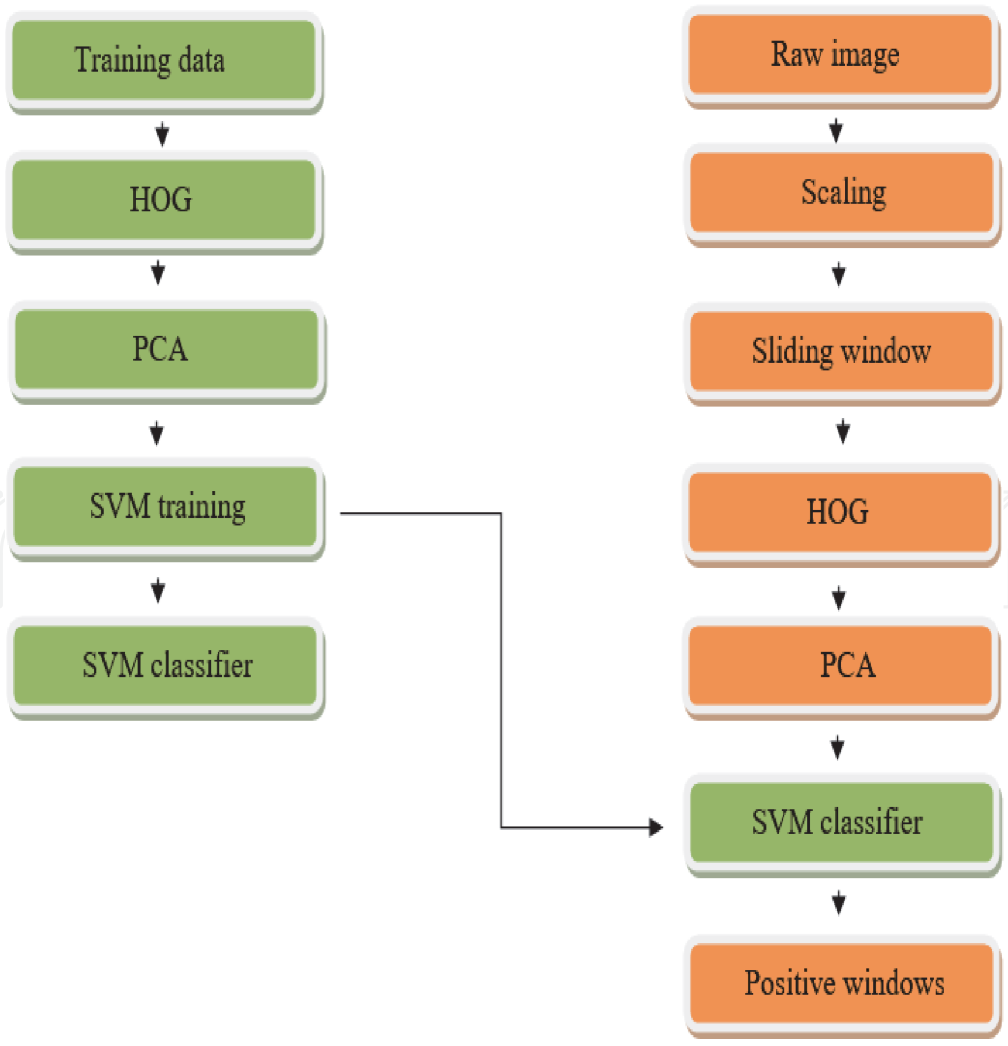


Figure 10.
Support vector machine flow chart. Green blocks represent training and orange blocks represent the test part of the application.

window. Calculated vehicle positions are used to form a heat map and the thresholded heat map is used to find a singular location for each vehicle (**Figure 10**).

3. Conclusion

This paper presents an approach based on image processing using edge distribution function and Hough transform for lane detection and tracking, steer angle estimation and vehicle detection and tracking for the autonomous vehicles. It was found that the instant change in the image feed is one of the most challenging parts in lane detection, and tracking part of this study as the model was vulnerable to instant changes in image feed. In order to prevent that, a temporal filter was applied to the region of interest which allowed shifting and hence increased the model's resistance to the aforementioned instant changes. According to the results of the tests, it was concluded that that the application of the temporal filter alone on the region of interest will not be ample; therefore, a filter was applied to the orientation of lane boundaries too, and the changes of greater than 2° in one frame. The model was found to be less dynamic, and an increase in its overall prediction accuracy was observed after those added aforementioned additions. One other issue that was discovered for this model is that it could be affected by the color changes. The model was found to be affected by the color changes on the road, shades of other vehicles, and trees on the side of the road. Errors caused by this situation usually hits the parabolic side of the estimation rather than linear.

Steer angle estimation model has a similar problem to the lane detection and tracking model. Considering the techniques that both the models share, it can be said that this was expected. This model is affected by color changes on road and shades also. If the same solution is applied, error can be lowered in color dynamic parts of the road. Another issue that was discovered was related to the parts of the road. The changes in the slope of the road cause an additional error. Because the model uses an inverse perspective mapping method that uses camera parameters as inputs, even if place occupied in Euclidean space by the camera does not change, the vanishing point which is one of the input parameters of IPM changes and this causes the model to make false predictions. To prevent this situation, image processing techniques or additional sensors can be used to estimate road slope. After predicting road slope, a model that can adapt to vanishing points shifts in the image can be developed and the error rate can be decreased drastically.

Author details

Mustafa F.S. Zortul, Tayyab Waqar* and Sezgin Ersoy
Mechatronics Engineering, Marmara University, Istanbul, Turkey

*Address all correspondence to: tayyabwaqar@marun.edu.tr

References

[1] Autonomous Vehicles. Available from: https://www.fia.com/autonomous-vehicles [Accessed: 19 June 2020]

[2] The future of mobility: Autonomous, electric and shared. Available from: https://www.ipsos.com/sites/default/files/ct/publication/documents/2019-11/the-future-of-mobility-autonomous-electric-shared.pdf [Accessed: 19 June 2020]

[3] Kato S, Takeuchi E, Ishiguro Y, Ninomiya Y, Takeda K, Hamada T. An open approach to autonomous vehicles. IEEE Micro. 2015;**35**(6):60-68

[4] Clark B, Parkhurst G, Ricci M. Understanding the Socioeconomic Adoption Scenarios for Autonomous Vehicles: A Literature Review. Project report. Bristol: University of the West of England; 2016

[5] Asadi BS, Tavana M, Asadi M, Oliver T. Autonomous vehicles: Challenges, opportunities, and future implications for transportation policies. Journal of Modern Transportation. 2016;**24**(4):284-303

[6] Azmat M, Schuhmayer C, Kummer S. Innovation in mobility: Austrian expert's perspective on the future of urban mobility with self-driving cars. In: Innovation Arabia 9: Quality and Business Management Conference (Business Innovation-Imperative for Knowledge Economy). Vol. 9. HBMSU Publishing House; 2016. pp. 142-160

[7] Assidiq Abdulhakam AM, Khalifa OO, Islam MR, Khan S. Real time lane detection for autonomous vehicles. In: 2008 International Conference on Computer and Communication Engineering. IEEE; 2008. pp. 82-88

[8] Chen Y-L, Chen Y-H, Chen C-J, Bing-Fei W. Nighttime vehicle detection for driver assistance and autonomous vehicles. In: 18th International Conference on Pattern Recognition (ICPR'06). Vol. 1. IEEE; 2006. pp. 687-690

[9] Mu G, Xinyu Z, Deyi L, Tianlei Z, Lifeng A. Traffic light detection and recognition for autonomous vehicles. Journal of China Universities of Posts and Telecommunications. 2015;**22**(1): 50–56

[10] Amaradi P, Sriramoju N, Dang L, Tewolde GS, Kwon J. Lane following and obstacle detection techniques in autonomous driving vehicles. In: 2016 IEEE International Conference on Electro Information Technology (EIT). IEEE; 2016. pp. 0674-0679

[11] Fang L, Wang X. Lane boundary detection algorithm based on vector fuzzy connectedness. Cognitive Computation. 2017;**9**(5):634-645

[12] Cortes C, Vapnik V. Support-vector networks. Machine Learning. 1995; **20**(3):273-297

[13] Lee JW. A machine vision system for lane-departure detection. Computer Vision and Image Understanding. 2002; **86**(1):52-78

[14] Bertozzi M, Broggi A. GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. IEEE Transactions on Image Processing. 1998;**7**(1):62-81