Vehicle Distance and Relative Speed Estimation from an Observer Vehicle using Vanishing Point

Muhammad Naufal Fakhrizal
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
mnaufalfakhrizal@gmail.com

Rinaldi Munir
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
rinaldi@informatika.org

***Abstract*—One of the tasks included in vehicle automation system is recognition of its surrounding environment, including the presence of other vehicles. This also includes spatial relationship between the observer vehicle and other vehicles, such as their distance and speed. Various methods to detect vehicles from an image sequence and estimate their distance exist, including ones utilizing the position of vanishing point of the road. A vanishing point of a road exists in an image such as the boundaries of the road appear to converge at the vanishing point. In this research, a method is developed to estimate road vanishing point, detect and track surrounding vehicles, and measure their distance and relative speed from image sequences. Road region detection and vanishing point estimation is done by detecting road boundaries using Hough transform, and vanishing point estimation is based on intersections of road boundaries. Vehicle detection is done using support vector machine that receives histogram of oriented gradients of vehicle hypotheses. Measurement of vehicle distance from observer vehicle utilizes the position of vehicle and vanishing point in the input image, and relative speed is determined using linear regression of vehicle distance variable as a function of time. The solution estimates vanishing point position, detects vehicles, tracks them, and measure their distance and relative speed with processing rate of 38.481 frames per second.**

***Index Terms*—image processing, machine learning, speed estimation, vanishing point, vehicle detection.**

I. Introduction

Several factors involved in traffic accidents include humans, vehicle, and environment. Human errors are found to be the main cause for most traffic accidents. Several solutions have been proposed to alleviate or reduce the human factor. This includes vehicle automation, where some of the driver's tasks is redirected to or shared by this system. One of such task is to recognize the environment surrounding the vehicle equipped by this system. The information regarding the surrounding environment is important for decision making in related tasks, such as automotive driver assistance system (ADAS), or full-on self-driving vehicles.

One aspect of the surrounding environment to be aware of is the presence of other vehicles, especially front vehicles, around the equipped vehicle. Informations of a front vehicle can be gathered, including inter-vehicle distance and its speed in relation to the equipped vehicle.

Several works have developed vision-based front vehicle detection and inter-vehicle distance estimation. However, few has investigated front vehicle speed measurement. This paper presents a method to detect front vehicles and estimate their inter-vehicle distance and relative speed.

II. Related Works

Based on the usage of sensors, Vehicle detection and inter-vehicle methods is categorized into methods using active sensors and methods using passive sensors [1] Methods using active sensors utilizes signal emitted by the sensor that is reflected by front vehicle. This reflected signal is the received by a receiver in the sensor, indicating that it has detected the presence of the vehicle. The round-trip time of the signal from being emitted to being received can be used to measure the distance of the front vehicle. However, active sensors are fairly expensive, and they are prone to interference from other signal emitted by similar sensors.

The most popular approach is to use passive sensors, such as camera. Passive sensors are inexpensive and do not introduce interferences. Many works have developed methods to detect front vehicle and estimate their inter-vehicle distance based on the information in images captured by the camera installed on a vehicle dashboard or windshield. Some approaches includes measuring the position of detected vehicle in relation to vanishing point [2], measuring both the position and width of the detected vehicle and comparing it to vehicle width od a known distance [3], and approximation using exponential function [4].

On the other hand, vehicle speed measurements have been done in other context, such as traffic monitoring using a static camera. A detected vehicle is tracked and the time it required to pass two predetermined lines with known distance is measured, which can be used to measure speed [5].

III. Design and Implementation

Fig. 1 shows the flowchart of the proposed method, which consists of several steps. The system receives image sequences such as video of a front road scene. First, A frame of the input images is preprocessed. Second, the road boundaries in the image is detected and the position of the vanishing point of the road is estimated. Third, The road region is segmented from the image based on the position of the vanishing point and the boundaries. Fourth, The front vehicles is detected and

then tracked. Lastly, The distance and relative speed of the detected front vehicle is estimated.

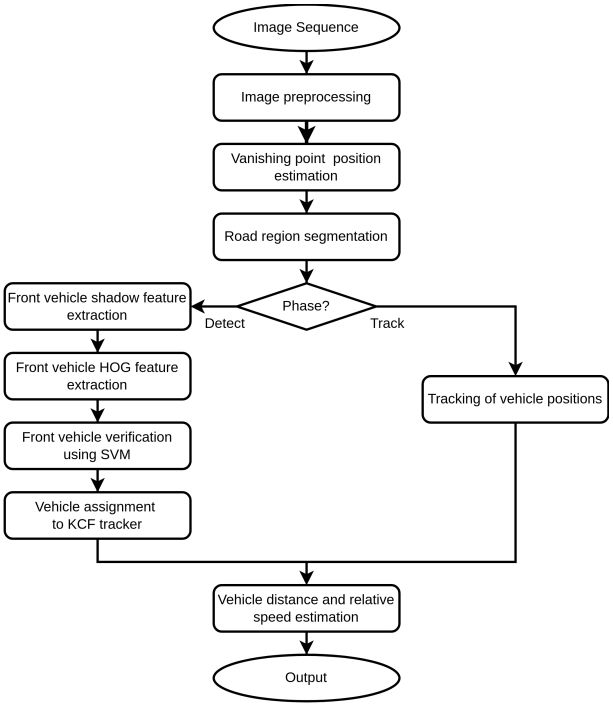

Fig. 1. Flowchart of the proposed method.

The vehicle detection and tracking are done in separate frames. n_{detect} frames of vehicle detection are then followed by n_{track} frames of vehicle tracking, completing a cycle of $n_{cycle} = n_{detect} + n_{track}$ frames.

A. Image Preprocessing

Some image frame properties are required for the following steps, which is achieved by preprocessing the image. First, the image is downsized to a predetermined size to reduce computation burden of the system. Then, the image is color-converted into grayscale.

Fig. 2. A preprocessed image.

B. Vanishing Point Estimation

First, Canny edge detection is performed on the preprocess frame, resulting in an edge image. Then, horizontal and vertical edges in the edge image are not needed and thus erased from the edge image. Hough transform is performed on the edge image to detect straight lines. From these lines, the candidates for left and right road boundaries are extracted such as the left candidates of road boundaries exist on the left half of the frame and go from bottom-left to top-right of the image, and vice versa. Then, intersection points of each left-right candidates of road boundaries pair is determined, and the point with which most line intersects is selected to be the candidate of vanishing point for this frame.

To estimate the position of the vanishing point, a number of vanishing point candidates from a number of subsequent frame are gathered. DBSCAN clustering is then performed with $\epsilon = 5$ and $MinPts = 3$ on the candidates, forming clusters. The largest cluster is then selected, and the center of mass of that cluster is selected as the position of the estimated vanishing point.

C. Road Region Segmentation

The estimated vanishing point position and the edge image are used to determine the true road boundaries. This is done by determining two lines traced from vanishing point to left and right edge of the image which go over the most amount of edge pixels.

D. Vehicle Detection

In general, vehicle detection process is divided into two step, hypothesis generation (HG) and hypothesis verification (HV). In HG step, vehicle hypotheses are generated using shadow features of the vehicle, as shown in Fig. 3. A shadow feature has lower pixel intensity than its surrounding area. First, shadow features of the front vehicles is extracted from road region segment by thresholding using Otsu's method. This thresholding is done twice to reduce false positives. Then, morphological processing is done to the shadow features to erase small noises and strengthen true shadow features. Small and narrow segments is erased using morphological erosion with 5×1 pixel rectangular structuring element. Then, morphological dilation is done twice using 3×3 pixel rectangular structuring element to restore and strengthen the remaining segments.

A bounding box is formed for each separate shadow feature segment. Then, bounding boxes that is too small and too large are removed. Shadow segments which ratio of their area to their bounding box area is lower than a certain threshold are also removed. Lastly, overlapping remaining bounding boxes are merged, resulting in the final bounding boxes of vehicle hypotheses.

In HV step, the object encompassed by a bounding box hypothesis is classified to be vehicle or non-vehicle object. First, a histogram of oriented gradient (HOG) of the part of the image encompassed by the bounding box is extracted. The HOG feature extraction is done on a 64×64 patches, using block size of 16×16 pixels, stride of 8×8 pixels, cell size of 8×8 pixels, and with 9 bins in range of 0 to 180 degrees, resulting in 1764-attributes HOG features. These features are

then fed to a support vector machine (SVM) with radial basis function (RBF) kernel and $C = 100$ to predict the classification of the hypotheses. The SVM is trained beforehand against KITTI dataset. The vehicle classified hypotheses are then kept, resulting in the detected vehicle positions as shown in Fig. 4.

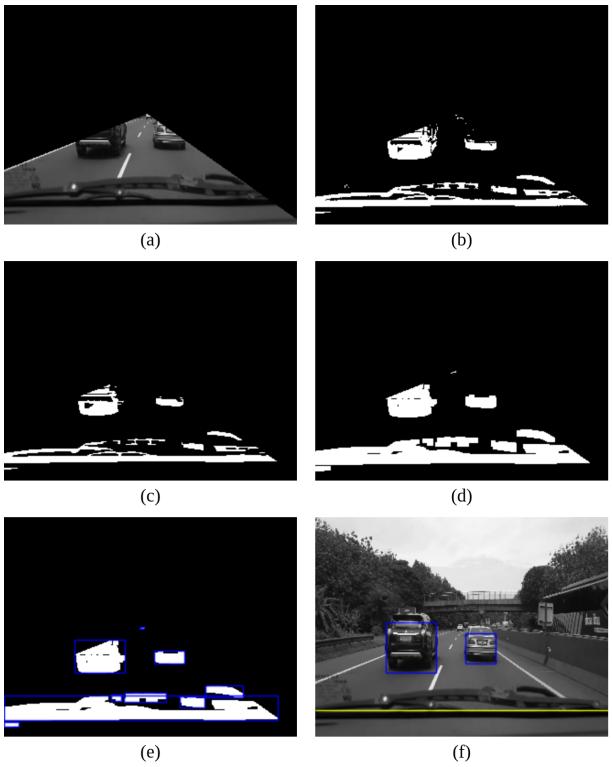

Fig. 3. The HG process of vehicle detection (a) road region segment, (b) shadow features, (c) eroded shadow features, (d) dilated shadow features, (e) initial bounding boxes, and (f) filtered bounding boxes.

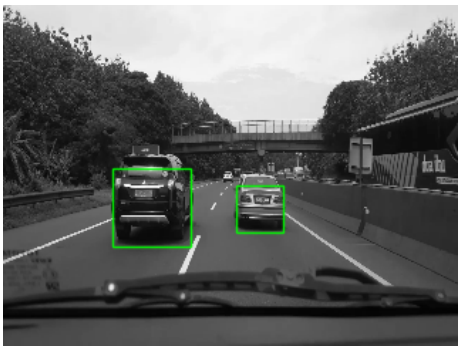

Fig. 4. The detected vehicle.

E. Vehicle Tracking

After a vehicle is detected, it was then assigned to a kernelized corellation filter tracker. In tracking frames, the tracker updates the position of the assigned vehicle for each frame. In the next detection frames, the newly detected vehicle need to be determined whether it belongs to an existing tracker or not. This was done by measuring the Jaccard distance of the new bounding box of the bounding boxes of existing trackers, assigning it to the closest tracker. If none exists that is closer than a certain threshold, a new tracker is created and the bounding box is assigned to it, as shown in Fig. 5. The Jaccard distance of two bounding boxes is computed using the following formula.

$$J_\delta(B_1, B_2) = 1 - \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|} \tag{1}$$

where $|B_1 \cap B_2|$ is the area of the intersection of both boxes, and $|B_1 \cup B_2|$ is the area of the union of both boxes.

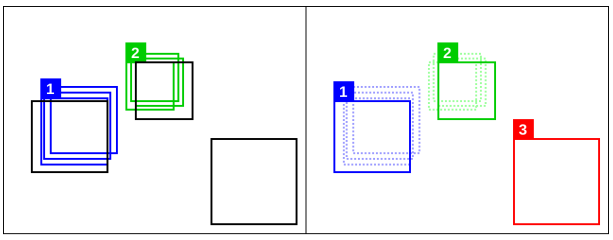

Fig. 5. Assignment of detection results to existing trackers (a) two trackers and three detected vehicles, and (b) two detected vehicle are assigned existing trackers, while the other is assigned a new tracker.

F. Inter-vehicle Distance and Relative Speed Estimation

Fig. 6 shows two instances of vehicle distance measurements from the position of bounding box and vanishing point. The camera is positioned at C, which sight line CC' is parallel to road surface. According to vanishing point theorem [6], two parallel lines appear to converge at a vanishing point. In this instance, the sight line and road surface appear to converge at vanishing point V in the image. An imaginary vision plane A is placed in front of camera at a distance of f. The camera's sight line CC' intersects with A at S', which appear at the image with Y axis position of y_{vp}, which is also the position of V.

In the first instance (Fig. 6(a)), a vehicle happens to be in front of camera at a distance of d_{veh}. the sight line CC' intersects with the back of the vehicle at point S_1. This point and the bottom of the vehicle B_1 forms line S_1B_1 of length l, that is projected into the image as $S'B'_1$ with length l_1. In the resulting image (Fig. 6(c)) of height H, The bottom of the vehicle is positioned at Y position of y_{veh}.

In the second instance (Fig. 6(b)), the vehicle appears at a distance of d_{base}. In the resulting image (Fig. 6(d)), the bottom of the vehicle appear to be touching the bottom of the image.

According to [6], the relationship of d_1 and d_2 is formulated by the following equation.

$$(f + d_1)l_1 = (f + d_2)l_2 \tag{2}$$

with

$$d_1 = d_{veh} - f \tag{3}$$

$$d_2 = d_{base} - f \tag{4}$$

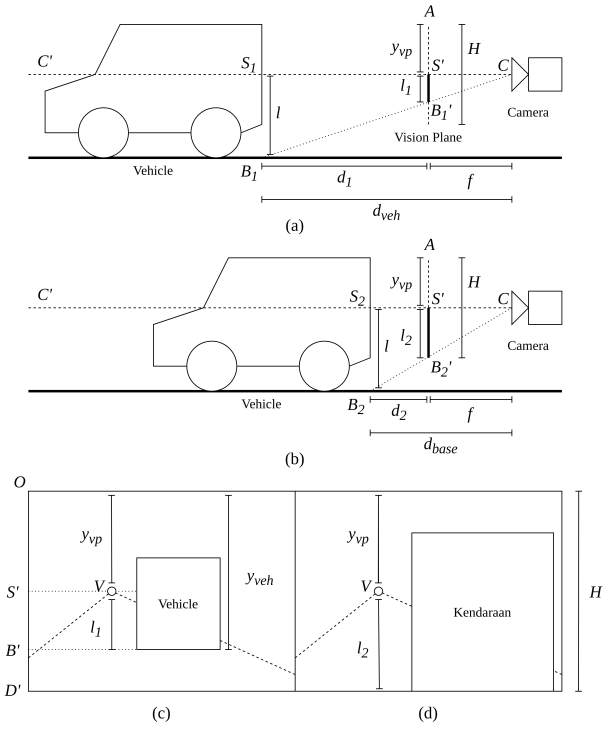

Fig. 6. Diagram of distance measurement (a) Side view of vehicle at a distance d_{veh} (b) Side view of vehicle at a distance d_{base} (c) The resulting image of (a), and (d) The resulting image of (b).

$$l_1 = y_{veh} - v_{vp} \tag{5}$$

$$l_2 = H - y_{vp} \tag{6}$$

resulting

$$(f+(d_{veh}-f))(y_{veh}-y_{vp}) = (f+(d_{base}-f))(H-y_{vp}) \tag{7}$$

$$d_{veh}(y_{veh} - y_{vp}) = d_{base}(H - y_{vp}) \tag{8}$$

$$d_{veh} = \frac{d_{base}(H - y_{vp})}{y_{veh} - y_{vp}} \tag{9}$$

The value of the d_{base} depends on the type of the equipped vehicle and the placement of the camera and is to be calibrated beforehand.

The estimation of vehicle relative speed is done using linear regression on d_{veh} measurements from a sequence of frames. This estimates a linear equation most fitting to the measured samples.

$$d_{veh} = v_{rel}t + c \tag{10}$$

with v_{rel} is the estimated relative speed and the coefficient of the equation, t is time, and c is the intercept.

IV. Experiment

The proposed system was implemented using Python 3.10.11 and run on a laptop with 3.2 GHz 8 core Intel i7-4702MQ and 8GB of RAM. Then, the performance of the system was measured and evaluated.

First, quantitative performance of the vehicle detection step of the system was tested and measured. The test was done on 9 input videos, showing road scenes of various conditions. The video was further categorized into two scenarios. The first scenario shows highway road scenes, while the second scenario shows urban road scenes at various times of day and traffic density. The test was done by comparing detection results on a frame with the vehicle position ground truth of that frame. This was done on detection frames, which was determined to be once every $n_cycles = 15$ frames. The ground truth (GT) was obtained by manually observing each of the test frames. Table I and II show the number of ground truths for each test video. The comparison resulted in the following quantitative metrics.

Fig. 7. The output images.

1) True positives (TP), which is the number of ground truths detected by the system.
2) False positives (FP), which is the number of detected objects not included in ground truths.
3) Recall score, which is TP/GT.
4) Precision score, which is $TP/(TP + FP)$.

TABLE I
The details of the first scenario videos.

Video	**frames**	**Test frames**	***Ground truths***
Video 1	1833	123	227
Video 2	1813	121	232
Video 3	948	64	190
Video 4	1001	67	100

TABLE II
The details of the second scenario videos.

Video	**frames**	**Test frames**	***Ground truths***
Video 5	1368	92	257
Video 6	1252	84	203
Video 7	1906	128	202
Video 8	2003	134	439
Video 9	1269	85	82

Table III shows the result of vehicle detection performance of the system.

TABLE III
Vehicle detection metrics on the first scenario videos.

Video	***Ground truths***	***TP***	***FP***	***Recall***	***Precision***
Video 1	227	188	39	0.828	0.979
Video 2	232	202	30	0.871	0.935
Video 3	190	139	51	0.732	0.959
Video 4	100	94	6	0.940	0.979
Total	749	623	126	0.832	0.960

TABLE IV
Vehicle detection metrics on the second scenario videos.

Video	***Ground truths***	***TP***	***FP***	***Recall***	***Precision***
Video 5	257	150	107	0.584	0.915
Video 6	203	97	106	0.478	0.933
Video 7	202	113	89	0.559	0.621
Video 8	439	210	229	0.478	0.737
Video 9	82	0	82	0.000	0.000
Total	1183	570	613	0.482	0.716

It was found that vehicle detection on highway scenes produces higher performance compared to urban road scenes. On urban scenes, especially congested area (Video 7), the road boundaries might be obstructed by other vehicle in a way such that it was not possible to extract road boundaries to estimate vanishing point, while straight lines detected on vehicles might cause interference in estimating the vanishing point.

In Video 9, the night road scene with nonuniform lighting caused the shadow feature extraction to fail. This was because there might be some unlit area that was darker than the shadow feature, which in turn might be illuminated by light sources such as observer vehicle's headlight.

Another performance to be measured was the processing rate of the implemented system. This was done by measuring the time needed for each step to complete on a frame, then computing the average for each step across all frames. Table V show the result of the average time measurement on the 9 test videos.

TABLE V
System processing rate measurement.

Step	**Avg. time**
Preprocessing	3.592ms
Vanishing point estimation and road region segmentation	9.708ms
Vehicle detection HG	2.461ms
Vehicle detection HV	18.542ms
Tracker assignment	1.959ms
Update tracker position	6.179ms
Distance and relative speed estimation	0.094ms
Postprocessing	0.117ms
Total	25.987ms
Processing rate (FPS)	38.481

It was found that in average, the system processes each frame in 25.987ms, which is equivalent to processing 38.481 frames per second. Using 30 FPS video as an input, the implemented system could achieve real-time processing.

V. Conclusion

The proposed method is able to detect vehicle using shadow features as hypothesis and verify the hypothesis using SVM to classify HOG features of the vehicle object. The proposed method is also able to estimate inter-vehicle distance of the detected vehicles using vanishing point as reference, and estimate their relative speed using linear regression on a sequence of distance measurement.

Acknowledgment

I'd like to thank Rinaldi Munir as the advisor of this research for the help, advice, and knowledge that made the completion of this research possible. I'd also like to thank my friends and family for their support during my study in Bandung Institute of Technology.

References

[1] Sun, Z., Bebis, G., & Miller, R. (2006). On-Road Vehicle Detection: A Review IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 694-711.

[2] Huang, D. Y., Chen, C. H., Chen, T. Y., Hu, W. C., & Feng, K. W. (2017). Vehicle Detection and Inter-Vehicle Distance Estimation Using Single-Lens Video Camera on Urban/Suburb Roads Journal of Visual Communication and Image Representation, Volume 46: 250-259.

[3] Kim, G. & Cho, J. S. (2012). Vision-Based Vehicle Detection and Inter-Vehicle Distance Estimation for Driver Alarm System. Optical Review, 19, 388-393.

[4] Fang, C. Y., Liang, J. H., Lo, C. S., & Chen, S. W. (2013). A Real-time Visual-based Front-mounted Vehicle Collision Warning System. 2013 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS) (pp. 1-8). IEEE.

[5] Nurwanto. (2016). Pendeteksian dan Perhitungan Jumlah Mobil di Jalan Raya dengan Memanfaatkan Algoritma *Background Substraction*. Tugas Akhir Program Sarjana, Institut Teknologi Bandung: Sekolah Teknik Elektro dan Informatika.

[6] Dong, S. P. (2003). Determination of 3-D position of particles in perspective picture with vanishing-point theorem. In Optical Technology and Image Processing for Fluids and Solids Diagnostics 2002 (Vol. 5058, pp. 104-112). SPIE.