Journal of Engineering Science and Technology
Vol. 13, No. 6 (2018) 1722 - 1734

A MODIFIED PINHOLE CAMERA MODEL FOR AUTOMATIC SPEED DETECTION OF DIAGONALLY MOVING VEHICLE

INDRABAYU[1]*, RIZKI YUSLIANA BAKTI[2], INTAN SARI ARENI[3]

[1]Informatics Study Program, Informatics Department, Universitas Hasanuddin
[2]Postgraduate Student of Electrical Engineering Department, Universitas Hasanuddin
[3]Electrical Study Program, Electrical Engineering Department, Universitas Hasanuddin
Jl. Poros Malino KM. 6 Gowa, 92171, Sulawesi Selatan, Indonesia
*Corresponding Author: indrabayu@unhas.ac.id

Abstract

This research applies the Pinhole (PH) Camera Model method to calculate car speed in a video sequence by taking object displacement distance in the frame. The main contribution in this paper is the new calculation method called Diagonal Pinhole (Diagonal-PH). It has a capability in measuring diagonal and zigzag movement. This research considers three scenarios, i.e. overtaking motion, lane changing path and straight-line movement. The performance of the proposed system is evaluated using Root Mean Square Error (RMSE). The results show that the diagonal-PH method provides smaller RMSE than a vertical-PH method for each scenario. This proves that the modified pinhole has better accuracy for car speed detection in all scenarios.

Keywords: Camera projection, Car speed, Intelligent transport system, Pinhole camera model.

1. Introduction

Intelligent Transportation System (ITS) is a vastly developing field in the world of transportation. An automatic detection of vehicle speed and acceleration is part of visual computing in ITS. Conventionally, vehicle speed can be measured in various ways. It could be done manually (calculating between spot marker) or by a laser gun [1]. The manual measurement could lead to error in counting particularly when the vehicle is moving at high speed and fast-changing acceleration. Meanwhile, the measurements of laser gun still require human to operate and should be fired directly at a moving vehicle. By applying Intelligent Camera Algorithm, the

Nomenclatures

B_l	Minimum path on the image
B_r	Maximum limit of the path on the image
$C_{P1(x)}$	Object coordinate point on the x-axis of previous frame
$C_{P2(x)}$	Object coordinate point on the x-axis of next frame
$\Delta C_{P(x)}$	Object displacement distance in the *x*-axis of the image
d_M	Actual object movement distance
d_{Mx}	Object distance in the *x*-axis
d_{My}	Object displacement distance in the *y*-axis
d_{RO}	Real object distance
d_x	The scale of the object displacement distance
d_θ	Distance from camera point to the initial limit capture area
F	Focal length
$frame_{(t)}$	Current frame
$frame_{(t-1)}$	Previous frame
H	Camera height
l_P	The length of the image pixels
K_T	A coefficient to convert m/s into km/h
N	The data number
v_{Fr}	Video speed
v_{PH}	Vehicle speed
v_{yi}	The real car speed
v_{yl}	System car speed
W_P	The width of image pixels
W_{PD}	The width of the road distortion
W_R	The actual width of the road
Y	Object pixel positions on the image

Greek Symbols

θ	Camera angle
Φ	Image projection angle

Abbreviations

GMM	Gaussian Mixture Models
HOG	Histogram of Oriented Gradient
ITS	Intelligent Transportation System
RGB	Red Green Blue
RMSE	Root Mean Square Error
ROI	Region of Interest

information on vehicle speed and acceleration can be automatically monitored by video. There are several methods to calculate the car velocity in a video, one of them is pinhole camera method. This method is using camera projection to calculate the distance moved by the object in the video frame. The research conducted by Lin et al. calculates the car speed based on the length horizontal blur of the car detected in an image and the relative position between the car and camera [2]. Herniaty et al. [3] followed a similar approach with the addition of blur parameter analysis using Fourier Transform. Both studies only considered the car horizontal movement, which was viewed from the side of the car. The same study was conducted by Nurhadiyatna et al. [4] comparing the Pinhole Camera Model and

Euclidean Distance to estimate the car speed. The pinhole camera model obtained the smallest average error. However, this study ignores the displacement of the car horizontally and only considers the vertical displacement of the vehicle with the object position seen vertically from the front of the car [5].

In a real situation, a car does not always move in a constant vertical straight line. Thus, this research proposed a new method to calculate the speed, which is called Diagonal Pinhole (Diagonal-PH) to accommodate nonstraight movement. This approach can count the vertical displacement and the horizontal displacement. This research is divided into three scenarios. The first scenario estimates the car speed when it is overtaking. The second scenario is when the car is changing lane motion. The third scenario is when the car is in a straight line. The first and second scenario is conducted because when the vehicle overtaking and changing the lane motion, it will happen diagonally vehicle movement so that results in a diagonal pixel shifted. All scenarios are used to compare the original pinhole camera model with Diagonal-PH in terms of measuring car’s speed.

2. Material and Methods

The first step in car speed estimation is vehicle detection. There are several methods, which can be used for vehicle speed detection. Some of the methods are Viola Jones, Histogram of Oriented Gradient (HOG) and Gaussian Mixture Models (GMM). Viola Jones and GMM algorithms proved to be simple and have smaller computation time [5]. Viola Jones and HOG require an enormous amount of datasets to improve the accuracy of the system but give higher accuracy compared to GMM [6, 7]. GMM method is used for moving objects speed detection in a video sequence and is capable of background and foreground extraction of videos. Next step is tracking an object by Kalman Filter. The last step is car speed estimation by Pinhole Camera Model method to calculate object displacement distance in the video frame.

2.1. Car detection

Segmentation is one of the stages of object detection. Generally, segmentation is performed using some methods. One of them is the growing methods. There are two types of region growing methods that are seeded and unseeded region growing methods. Region growing method works with the principle that, the neighbouring pixels within one region have similar pixel value compared with its neighbours. If there is a similarity between pixels then the pixel is said to belong to that region as one of its neighbour [8].

The Gaussian models in this research were described as an object segmentation technic. This segmentation converts the video frame data to binary 0 and 1 by applying the background and foreground concepts. The use of more models in each pixel will cause the background extraction process to be more adaptable because it can be modelled in every pixel. Compatibility of pixels to all Gaussian distribution models was constructed so pixels substitution can take place. If the compatibility process does not match, then the Gaussian model with the smallest probability will be deleted and replaced with the Gaussian model for the new pixel colour. The next step is to determine which pixels are included in the foreground and background objects. A background refers to a stationary object in the video frame, and a foreground is a moving object in the video frame. If the pixel colour is in the category of the gaussian model value, then it will be considered as background (pixels rated 0/black), else it will be considered as foreground (pixels rated one/white) [5].

Objects that are detected as foreground and marked by a blob (a set of pixels that have a neighbour relationship). The accuracy of detection system can be improved by optimizing the blob parameters. Because gaussian works on each pixel, it is done pre-processing to reduce computation to determine the Region of Interest (ROI) in the video frame.

Blob analysis is required to obtain the desired foreground object. Several blob parameters must be considered to optimize the detection results, which are minimum blob area and maximum blob area. These parameters must be set so that object detected is only car.

Figure 1 shows a blob analysis on car detection. Part (a) shows an **original image** of the detected car which is characterized by a bounding box in the original image. Part (b) shows a binary converted image of part (a). Even though several foreground objects are detected, only the intersect of the minimum blob area and maximum blob area is recognized as a car. The detected object is marked with a bounding box.

Figure 2 shows the example of ROI and Non-ROI in the video frame in both original and binary converted images. ROI in pre-processing is needed to limit the detection area and reduce unneeded pixels. ROI is an important area where a targeted object can be detected. While, Non - ROI is the outside area of the ROI. In this research, determination of ROI area is conducted manually by determining the coordinate points on the road that forms a trapezoid. Initialization of coordinate points is conducted to give a limitation on the object that will be detected. Moving objects are in the trapezoid area will be detected as a vehicle, while that is outside the trapezoid area will be eliminated or undetected.

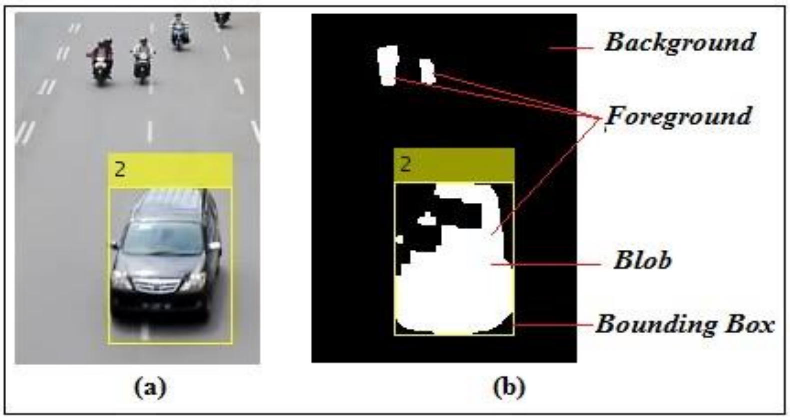

Fig. 1. Blob analysis on the car detection.

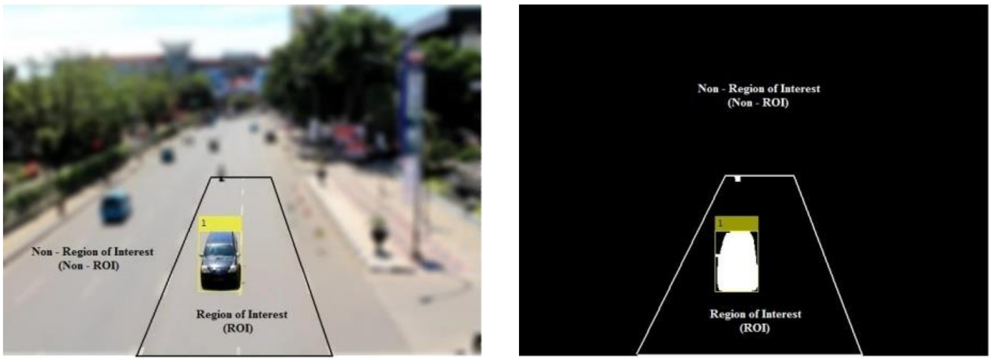

Fig. 2. ROI in pre-processing.

2.2. Tracking object

To find out the object position in the video, three types of object tracking were used (point tracking, kernel-based tracking and tracing by the silhouette). In this study, tracking point is used to locate the bounding box centre point of the detected object in every frame of video. So, the changing position and direction of car motion can be detected. One of the recursive object tracking methods is Kalman Filter [9, 10]. This method is used for object tracking and predicting the position and direction of object movement in a video. The estimated current position of the object is influenced by the previous object position. The direction of a car motion in the real conditions does not always move in a straight line. Therefore, three scenarios of car motion to estimate car speed are proposed in this paper as shown in Fig. 3, i.e., overtaking motion, changing lane motion and straight-line motion.

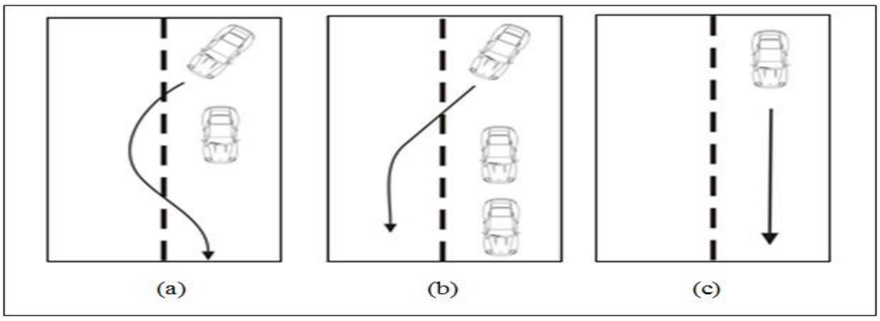

Fig.3. Scenarios. (a) Overtaking motion,
(b) Changing lane motion, (c) Straight line motion.

2.3. Car speed estimation

Car speed estimation can be obtained by knowing the object displacement distance from one frame to next frame using pinhole camera model with the *x*-axis and *y*-axis projections to see the vehicle movement in horizontal and vertical view. The illustration of *y*-axis projection camera is shown in Fig. 4 to figure out the real object distance (d_{RO}).

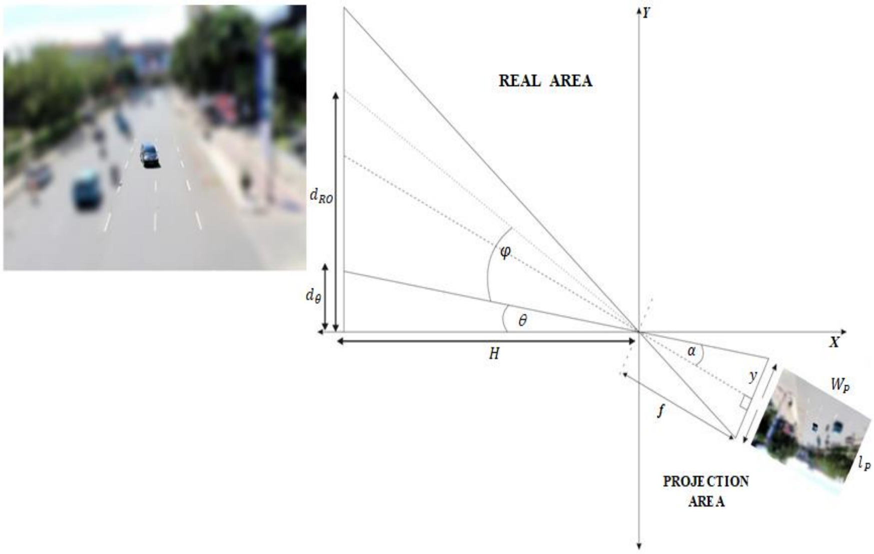

Fig. 4. The *y*-axis projection of pinhole camera model.

The d_{RO} is calculated by the following formula [4].

$$d_{RO} = H \times \tan(\varphi + \theta) \quad (1)$$

$$\varphi = tan^{-1}\frac{(l_P/2)}{f} - tan^{-1}\frac{(y - l_P/2)}{f} \quad (2)$$

$$\theta = \frac{d_\theta}{H} \quad (3)$$

where H is camera height, θ is camera angle, f (focal length) is the distance from the centre point of the lens to the image plane in pixel, y is an object pixel positions on the image, φ is image projection angle, W_P is the width of image pixels, l_P is the length of the image pixels and d_θ is the distance from camera point to the initial limit capture area. After calculating the d_{RO} value, the object displacement in each frame will be determined based on the illustration of *x*-axis projection camera as shown in Fig. 5.

Comparison of projection and real areas on the *x*-axis projection is shown in Figure 6 to get the scale of the object displacement distance (d_x) by using the following equation:

$$d_x = \Delta C_{P(x)} \times \frac{W_P}{W_{PD}} \quad (4)$$

$$W_{PD} = B_r - B_l \quad (5)$$

$$\Delta C_{P(x)} = \left|C_{P1(x)} - C_{P2(x)}\right| \quad (6)$$

where W_{PD} is the width of the road distortion that changes according to the image pixel number. $\Delta C_{P(x)}$ is the object displacement distance in the x-axis of the image. B_r is the maximum limit of the path on the image. B_l is the minimum path on the image. $C_{P1(x)}$ is the object coordinate point on the x-axis of the previous frame. $C_{P2(x)}$

is the object coordinate point on the x-axis of next frame. After that, the vehicle speed (v_{PH}) in km can be calculated by the following equation:

$$v_{PH} = \frac{d_M \times v_{Fr} \times K_T}{frame_{(t)} - frame_{(t-1)}} \tag{7}$$

$$d_M = \sqrt{d_{Mx}^2 + d_{My}^2} \tag{8}$$

$$d_{Mx} = d_x \times \frac{W_R}{W_P} \tag{9}$$

$$d_{My} = d_{RO1} - d_{RO2} \tag{10}$$

where v_{Fr} is video speed in fps unit, K_T is a coefficient to convert m/s into km/h of 3.6, $frame_{(t)}$ is the current frame, $frame_{(t-1)}$ is the previous frame, d_M is the actual object movement distance. d_{Mx} is the object displacement distance in the *x*-axis. d_{My} is the object displacement distance in the *y*-axis. W_R is the actual width of the road. d_{RO1} and d_{RO2} are the real object distance on the *y*-axis.

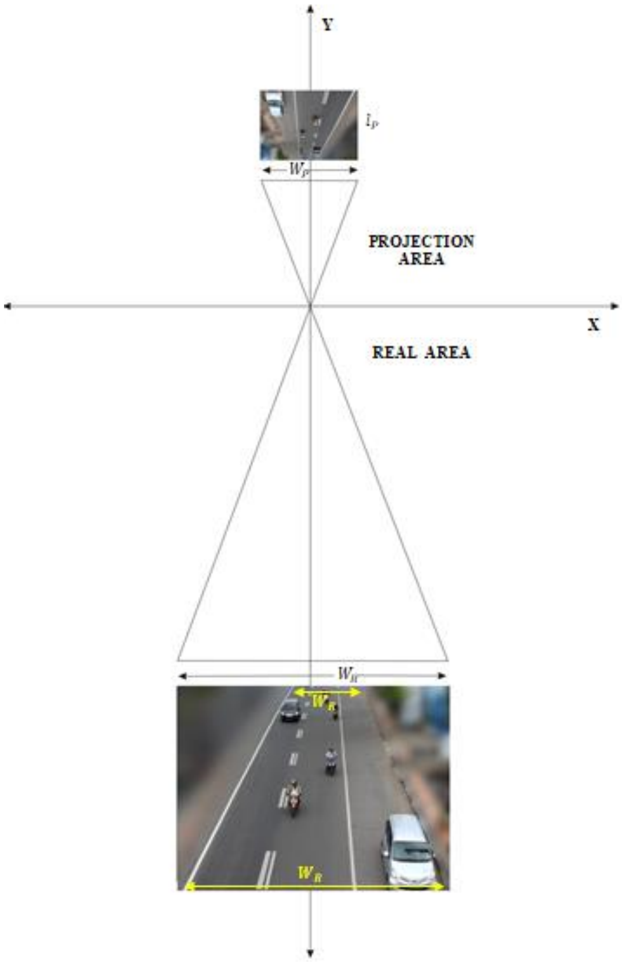

Fig. 5. The illustration of *x*-axis projection camera.

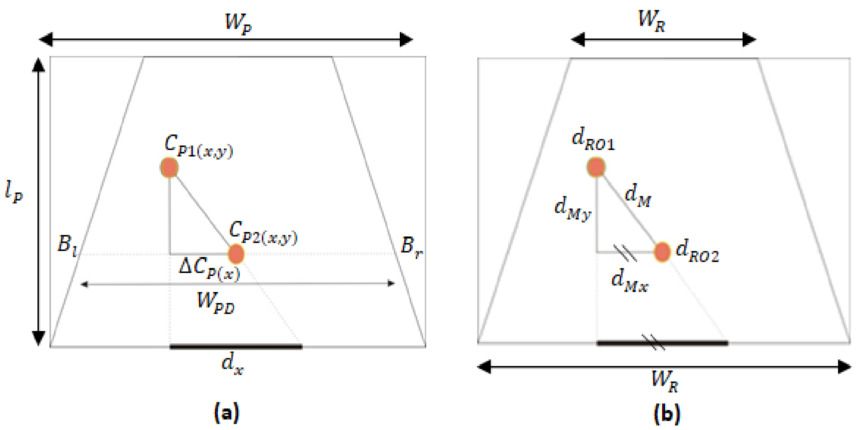

Fig. 6. *x*-axis projection at, (a) projection area, (b) real area.

3. Results and Discussion

In this research, the data retrieval technique is conducted by using a camera, which is placed on the pedestrian bridge. The time of data retrieval is conducted during the day from 10.00 am to 13.00 pm, because at this time, there is less traffic jam on the road and the daylight is supporting the detection process. The experiment is done using a test car for system validation. The recording is conducted when the test car passes through the area captured by the camera. This is conducted to create a condition where there is only one vehicle that passes on the street at the captured time. The aim is to see the performance of the proposed method when calculating vehicle speed that moving diagonally.

There are some values that should be determined before applying the proposed methods such as camera angle, camera height, distance from camera point to the initial limit capture area, and the actual width of the road.

Validation system is completed by comparing the proposed method (diagonal-PH) and the previous method (vertical-PH) with the real car speed, and the system performance is calculated by using RMSE as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(v_{yi} - v_{yl}\right)^2}{N}} \tag{11}$$

where v_{yi} is the real car speed, v_{yl} is the system vehicle speed, and N is the data number.

The illustration of car position in each frame is shown in Fig. 7 with the symbol of *A*, *B*, *C* and *D*. Meanwhile, the car displacement pixel position in each video frame is marked by 1, 2, 3 and 4. The car test is detected when entering the ROI area. The example of the car test detected is shown in Fig. 8. In that figure, the car started detected in frame 130 with the car speed of 0 km/hour because of no pixel position shifts in the frame. Vehicle speed starts to count when the object is in the frame 135, frame 140 and frame 145 until the last frame of the car is detected. This research used video data with 640×480 resolution and 25 fps frame rate for diagonal-PH and vertical-PH.

Figure 9 shows all scenarios, whereby the vertical-PH performance is different from the actual values in car spot position of 1, 2, 4, 5 and 6 due to car displacement with the diagonal position. For the car position at point 3, the diagonal-PH and vertical-PH is nearly coincided because of the car displacement in a straight position. The comparable result is also shown in Figs. 8 and 9.

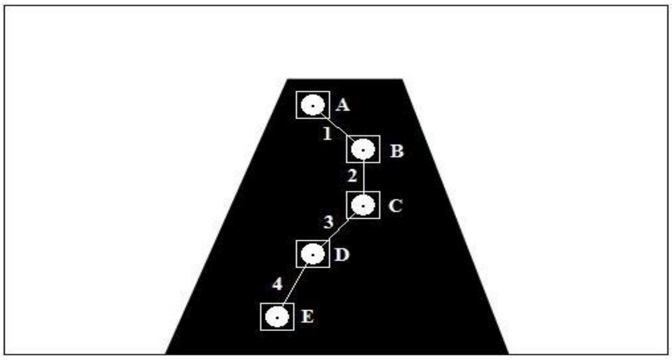

Fig. 7. Car spot position in video.

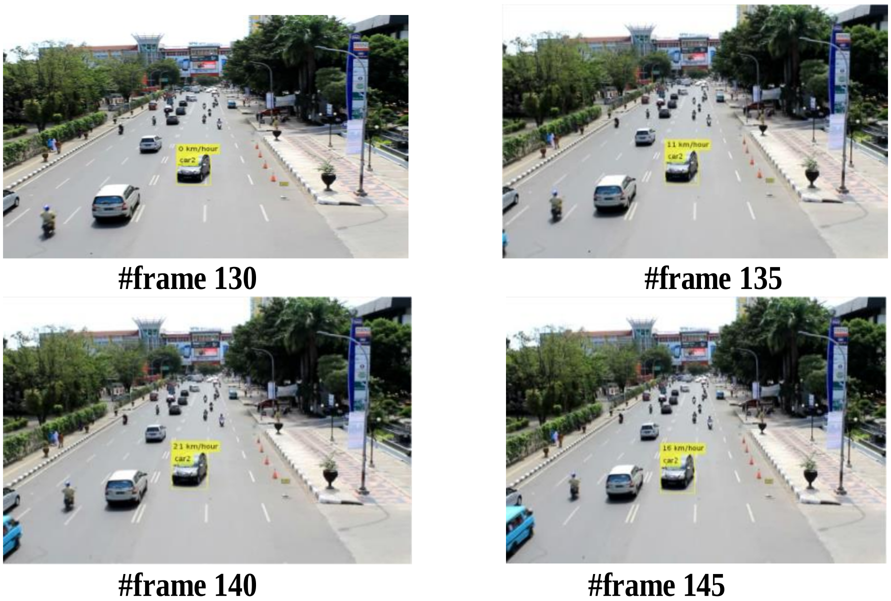

Fig. 8. The example of the car detected in each frame.

The comparison of RMSE values between diagonal-PH and vertical-PH for scenario 1, scenario 2 and scenario 3 are shown in Tables 1, 2 and 3, respectively. The results indicate that the RMSE values of diagonal-PH have a smaller value than the vertical-PH for all scenarios. This proves that the diagonal-PH obtained better performance than vertical-PH. Based on the results, the calculation of vehicle speed using the proposed method is almost the same as the real speed when the vehicle is detected correctly at the stage of vehicle detection.

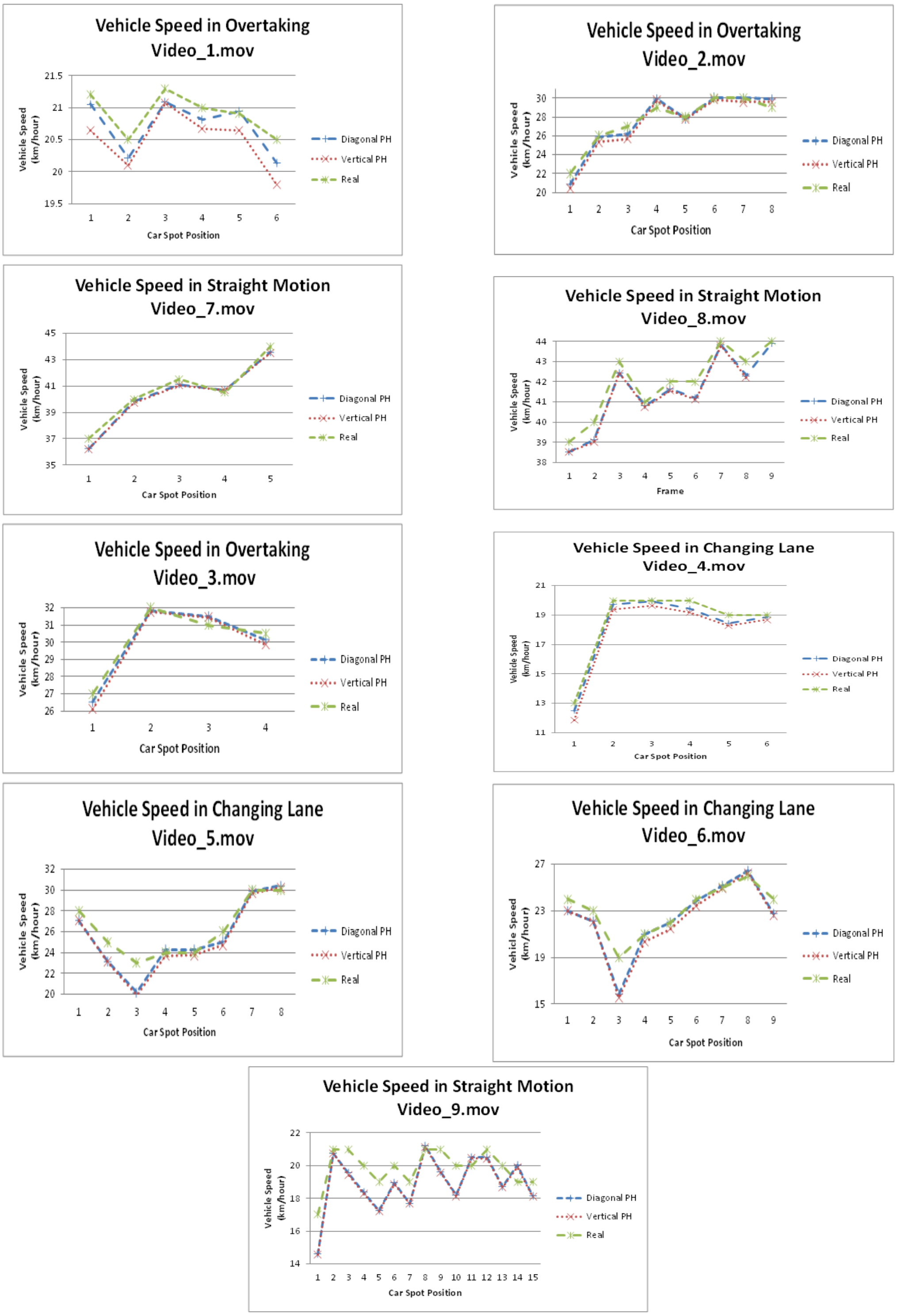

Fig. 9. Vehicle speed of car position scenarios

Table 1. Car speed scenario

Video name : Video_1.mov

Car spot position	V_{DPH} (km/hour)	V_{VPH} (km/hour)	V_{real} (km/hour)	V_{real} - V_{VPH}	V_{real} - V_{VPH}
1	21.1	20.6	21.2	0.01	0.36
2	20.2	20.1	20.5	0.09	0.16
3	21.1	21.1	21.3	0.04	0.04
4	20.8	20.7	21.0	0.04	0.09
5	20.9	20.6	20.9	0.00	0.09
6	20.1	19.8	20.5	0.16	0.49
RMSE				0.22	0.45

Video name : Video_2.mov

Car spot position	V_{DPH} (km/hour)	V_{VPH} (km/hour)	V_{real} (km/hour)	V_{real} - V_{VPH}	V_{real} - V_{VPH}
1	20.9	20.4	22.0	1.21	2.56
2	25.9	25.4	26.0	0.01	0.36
3	26.2	25.7	27.0	0.64	1.69
4	29.9	29.8	29.0	0.81	0.64
5	27.8	27.8	28.0	0.04	0.04
6	30.1	29.9	30.0	0.01	0.01
7	30.0	29.6	30.0	0.00	0.16
8	29.9	29.6	29.0	0.81	0.36
RMSE				0.66	0.85

Video name : Video_3.mov

Car spot position	V_{DPH} (km/hour)	V_{VPH} (km/hour)	V_{real} (km/hour)	V_{real} - V_{VPH}	V_{real} - V_{VPH}
1	26.5	26.1	27.0	0.25	0.81
2	31.8	31.8	32.0	0.04	0.04
3	31.5	31.4	31.0	0.25	0.16
4	30.1	29.9	30.5	0.16	0.36
RMSE				0.42	0.59

Table 2. Car speed scenario 2.

Video name : Video_4.mov

Car spot position	V_{DPH} (km/hour)	V_{VPH} (km/hour)	V_{real} (km/hour)	V_{real} - V_{VPH}	V_{real} - V_{VPH}
1	12.5	11.9	13.0	0.25	1.21
2	19.7	19.4	20.0	0.09	0.36
3	19.9	19.6	20.0	0.01	0.16
4	19.4	19.2	20.0	0.36	0.64
5	18.5	18.3	19.0	0.25	0.49
6	18.9	18.7	19.0	0.01	0.09
RMSE				0.40	0.70

Video Name : Video_5.mov

Car spot Position	V_{DPH} (km/hour)	V_{VPH} (km/hour)	V_{real} (km/hour)	V_{real} - V_{VPH}	V_{real} - V_{VPH}
1	27.1	27.1	28.0	0.81	0.81
2	23.2	23.1	25.0	3.24	2.89
3	20.2	19.9	23.0	7.84	9.61
4	24.3	23.7	24.0	0.09	0.09
5	24.3	23.7	24.0	0.09	0.09
6	25.0	24.7	26.0	1.00	1.69
7	29.9	29.7	30.0	0.01	0.09
8	30.4	30.2	30.0	0.16	0.04
RMSE				1.29	1.38

Video name : Video_6.mov

Car spot Position	V_{DPH} (km/hour)	V_{VPH} (km/hour)	V_{real} (km/hour)	V_{real} - V_{VPH}	V_{real} - V_{VPH}
1	23.0	23.0	24.0	1.00	1.00
2	22.1	22.1	23.0	0.81	0.81
3	15.8	15.5	19.0	10.24	12.25
4	21.0	20.3	21.0	0.00	0.49
5	22.0	21.4	22.0	0.00	0.36
6	23.9	23.4	24.0	0.01	0.36
7	25.1	24.9	25.0	0.01	0.01
8	26.5	26.2	26.0	0.25	0.04
9	22.7	22.6	24.0	1.69	1.96
RMSE				1.25	1.39

Table 3. Car speed scenario

Video name : Video_7.mov

Car spot position	V_{DPH} (km/hour)	V_{VPH} (km/hour)	V_{real} (km/hour)	V_{real} - V_{VPH}	V_{real} - V_{VPH}
1	36.2	36.2	37.0	0.64	0.64
2	39.8	39.7	40.0	0.04	0.03
3	41.1	41.0	41.5	0.16	0.25
4	40.7	40.7	40.5	0.04	0.04
5	43.6	43.5	44.0	0.16	0.25
RMSE				0.46	0.49

Video name : Video_8.mov

Car spot position	V_{DPH} (km/hour)	V_{VPH} (km/hour)	V_{real} (km/hour)	V_{real} - V_{VPH}	V_{real} - V_{VPH}
1	38.5	38.5	39.0	0.25	0.25
2	39.1	39.0	40.0	0.81	1.00
3	42.4	42.4	43.0	0.36	0.36
4	40.8	40.8	41.0	0.04	0.04
5	41.6	41.6	42.0	0.16	0.16
6	41.2	41.1	42.0	0.64	0.81
7	43.8	43.8	44.0	0.04	0.04
8	42.3	42.2	43.0	0.49	0.64
9	43.9	43.8	44.0	0.01	0.04
RMSE				0.56	0.61

Video Name : Video_9.mov

Car spot position	V_{DPH} (km/hour)	V_{VPH} (km/hour)	V_{real} (km/hour)	V_{real} - V_{VPH}	V_{real} - V_{VPH}
1	14.6	14.6	17.0	5.76	5.76
2	20.8	20.7	21.0	0.04	0.09
3	19.6	19.4	21.0	1.96	2.56
4	18.4	18.3	20.0	2.56	2.89
5	17.3	17.2	19.0	2.89	3.24
6	19.0	18.9	20.0	1.00	1.21
7	17.7	17.7	19.0	1.69	1.69
8	21.2	21.1	21.0	0.04	0.01
9	19.6	19.5	21.0	1.96	2.25
10	18.2	18.1	20.0	3.24	3.61
11	20.5	20.4	20.0	0.25	0.16
12	20.5	20.4	21.0	0.25	0.36
13	18.7	18.7	20.0	1.69	1.69
14	20.0	19.9	19.0	1.00	0.81
15	18.2	18.1	19.0	0.64	0.81
RMSE				1.29	1.35

References

1. Adnan, M.A.; Sulaiman, N.; Zainuddin, N.I.; and Besar, T.B.H.T. (2013). Vehicle speed measurement technique using various speed detection instrumentation. *Proceedings of the Business Engineering and Industrial Applications Colloquium*. Langkawi, Malaysia, 668-672.
2. Lin, H.-Y.; Li, K.-J.; and Chang, C.-H. (2008). Vehicle speed detection from a single motion blurred image. *Image and Vision Computing*, 26(10), 1327-1337.
3. Herniaty, A.P.; Ningrum, I.P.; and Dewi, A.P. (2015). Estimasi kecepatan kendaraan menggunakan citra motion blur. *SemanTIK*, 1(1), 45-52.
4. Nurhadiyatna, A.; Hardjono, B.; Wibisono, A.; Sina, I.; Jatmiko, W.; Ma'sum, M.A.; and Mursanto, P. (2013). Improved vehicle speed estimation using gaussian mixture model and hole filling algorithm. *Proceedings of the International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Bali, Indonesia, 451-456.
5. Indrabayu; Basri.; Achmad, A.; Nurtanio, I.; and Mayasari, F. (2015). Blob modification in counting vehicles using gaussian mixture models under heavy traffic. *Asian Research Publishing Network (ARPN) Journal of Engineering and Applied Sciences*, 10(16), 7157-7163.
6. Djamaluddin, D.; Indrabulan, T.; Andani; Indrabayu; and Sidehabi, S.W. (2014). The simulation of vehicle counting system for traffic surveillance using Viola Jones method. *Proceedings of the Makassar International Conference on Electrical Engineering and Informatics (MICEEI)*. Makassar, Indonesia, 130-135.
7. Bakti, R.Y.; Indrabayu.; and Areni, I.S. (2015). Klasifikasi bertingkat untuk deteksi mobil. *Prosing Seminar Nasional Teknik Informatika (SNATIKA). Yogyakarta, Indonesia.*
8. Biradar, V.B.; Jayaramaiah, G.V.; and Vijayakumar, R.H. (2015). Review of image co segmentation technique. *International Journal for Scientific Research & Development*, 3(5), 981-984.
9. Santosh, D.H.H.; Venkatesh, P.; Poornesh, P.; Rao, L.N.; and Kumar, N.A. (2013). Tracking multiple moving objects using gaussian mixture model. *International Journal of Soft Computing and Engineering*, 3(2), 114-119.
10. Indrabayu; Bakti, R.Y.; Areni, I.S. and Prayogi, A.A. (2016). Vehicle detection and tracking using gaussian mixture model and kalman filter. *Proceedings of the International Conference on Computational Intelligence and Cybernetics (CYBERNETICSCOM).* Makassar, Indonesia, 115-119.