International Journal of Innovative
Computing, Information and Control
Volume **15**, Number **4**, August **2019**

ICIC International ©2019 ISSN 1349-4198
pp. **1397–1411**

CAR-FOLLOWING MODEL: A COMPUTER-VISION-BASED CALIBRATION METHOD

Fergyanto Efendy Gunawan

Industrial Engineering Department, BINUS Graduate Program – Master of Industrial Engineering
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
fgunawan@binus.edu

Received August 2018; revised December 2018

Abstract. *Car-following model is a mathematical model that regulates the movement of a vehicle in the longitudinal direction in microscopic level. Commonly, the model describes the vehicle movement as a function of the vehicle relative position and velocity with respect to the leading vehicle. One of the widely used models is the Gazis-Herman-Rothery model, which is characterized by coefficients m and l. The values of the coefficients vary depending on many aspects such as vehicle type and road condition. The coefficients are usually determined from a calibration test where the vehicle position, velocity, and acceleration are measured accurately. So far, a few calibration methods have been proposed; some modern methods are by using remote sensing and RTK GPS. In the current work, the vehicle movement is recorded in a perspective view from an elevation. The recorded vehicle movement is analyzed for the vehicle position using computer vision methods. Then, the position is transformed to the actual vehicle position. Two computer vision methods are evaluated: multilayer- and Eigen-background-subtraction methods. The proposed method is evaluated to track the movement of a vehicle traveling in a short-straight distance. The results show that the tracking accuracy of the multilayer-background-subtraction method is better than the Eigen-background-subtraction method. The multilayer-background-subtraction method has 96.6% for position accuracy and 88.9% for velocity accuracy, while the Eigen-background-subtraction method has 92.9% for position accuracy and 84.3% for velocity accuracy. The most reliable car following parameters are estimated with 3.2% of error. The obtained parameter m is 0.4 and l is 1.2. Various values of m and l have been proposed by many researchers where the reliable values are in the range of 0-2.7 for m and 0-2.8 for l. Our findings are within the range.*

Keywords: Car-following model, Vehicle tracking, Computer vision, Micro simulation

1. **Introduction.** Traffic congestion is a major issue faced by many metropolitan cities including the city of Jakarta in Indonesia. Indonesia as a whole has the road networks with a total length of 508 000 km approximately and the total number of vehicles is about 104 118 969 units including trucks, buses, passenger cars, and motorcycles. Over the last 10 years, the number of vehicles has increased by about 14.6% per year but the road length has only increased by about 3.5% per year [1]. Thus, the traffic density has increased significantly. As a result, traffic congestion often occurs particularly in the capital city.

[2] proposed a long-term strategic development for transportation systems in metropolitan cities. He suggested that metropolitan cities had to structure their public transportation systems on the basis of their operational speeds. A low operational speed of public transportation should be used to serve closely spaced stations. For this purpose, Jakarta

DOI: 10.24507/ijicic.15.04.1397

administration adopted a bus-based system, TransJakarta, in 2003 [3]. The operational efficiency of the system has been the topic of a number of research works [4, 5, 6].

Another approach to lessen the congestion level can be achieved by means of better traffic management where the level of demand is regulated and controlled. The traffic management often requires forecasting of the impacts of implementing various traffic management measures. Testing and evaluating the measure in the actual road are less effective and efficient as it may severely disrupt the traffic condition. Traffic simulation is another means to evaluate the effects of the traffic management interventions prior to the actual implementation.

One of the traffic modeling techniques is micro simulation. Micro simulation shows the movement of every vehicle that is traced through a road network over time at a small time increment of a fraction of a second [7]. There are many types of computer modelings in micro simulation that look at the interaction of individual "units" such as households, people or vehicles. This analytical tool can perform highly detailed analysis of activities for better understanding of the current policies and their effects in the real world. For example, a traffic micro simulation model can be used to evaluate the effectiveness of widening a lane.

In the microscopic scale, car-following model is one of the important components for modeling how a vehicle interacts with other vehicles. The generalized car-following model has the conception that "a driver's response is determined by sensitivity and stimuli" [8]. One of the widely used car-following models is Gazis-Herman-Rothery (GHR) model, which has the following mathematical form [9]:

$$a_n(t) = \alpha \frac{v_n^m(t) \cdot \Delta v_n(t - t_d)}{\Delta x_n^l(t - t_d)}, \tag{1}$$

where $a_n(t)$ is the vehicle acceleration, $v_n(t)$ is the vehicle velocity, and Δv_n and Δx, respectively, are the relative velocity and the relative position with respect to the leading vehicle. The model has three parameters α, m, and l, and has a delay of t_d. The driver delay or response time t_d is usually about 1 s; α is the driver sensitivity [9].

The central issue for the use of the micro simulation model is finding the model parameters for each type of vehicles, road types, traffic conditions, and environment. The parameters are usually determined via a calibration process. There are few existing calibration methods for the car-following model. In the calibration process, it is essential to record the vehicle and its leading vehicle's position and velocity.

The first experiment to identify the car-following model parameters was performed by Chandler et al. [10]. They utilized a fine wire reel and power unit mounted on a small platform, which was fastened on the front bumper of the test vehicle. Wire tension was maintained constant by means of a slipping friction clutch. The information of $x_l - x_f$, $v_l - v_f$, v_f, and a_f is recorded simultaneously by an oscillograph installed in back of the test vehicle seat. The subscripts l and f respectively denote the leader and follower vehicles.

[11, 12] proposed a remote sensing technique to identify the car-following model parameters. They demonstrated the use of this technique by analyzing the traffic along a 280-m road segment of a Dutch motorway. However, given the limited stability of the helicopter, only 210 m of the motorway could be used for vehicle detection and tracking. The resolution of the data collection was about 22 cm. This method clearly requires rather expensive equipments such as helicopter and high resolution digital camera, whereas the limited stability of helicopter will be a problem in collecting the image.

Other calibration method is to use an extremely expensive Real Time Kinematic (RTK) GPS to track the movement of each vehicle. Each studied vehicle should be equipped with

an RTK GPS receiver with a position accuracy of 10 mm [13]. RTK GPS is able to measure the vehicular movement at an outstanding level of accuracy and much more conveniently than ever before but the GPS price is very high.

All of that existing calibration methods need many preparations on the equipment and the cost is also very expensive. For this reason, this research intends to establish a significantly low-cost calibration procedure where the vehicle movements will be recorded digitally, tracked with a computer vision technique, and projected with an orthography projection technique.

This manuscript is organized as the following. Section 2, Research Methods, describes methods to track the vehicle position, a method to map the vehicle position on a perspective plane to those on a road plane, a metric to evaluate prediction accuracy, and the experimental procedure. Section 3, Results, presents tracked vehicle positions on the perspective and road planes, comparisons of the vehicle positions and velocities of the current methods and of those of numerical integration, the estimated GHR model coefficients, and the estimation errors. Section 4, Conclusions, briefly presents the salient aspects of the research.

2. **Research Methods.**

In this work, the vehicle movement is recorded using a video camera from an elevation. The vehicle is assumed to move along a lane, which is marked on its four corners. The position of these four-corner point should be measured accurately because they will be used to project the vehicle recorded position to the actual position. This section discusses a number of basic theories that supports the current works.

2.1. **Computer vision.**

The `openCV` library that is used for vehicle tracking is `cvBlob`. The `cvBlob` library is a computer vision library that is used to detect connected regions in binary digital images [14]. The `cvBlob` can perform a labeling and features extraction especially for the object tracking. The working steps in the vehicle tracking using `openCV` are as follows.

1) A background subtraction would be performed. The background subtraction methods are the multilayer-background-subtraction method [15] and the Eigen-background-subtraction method [16].
2) The masked foreground binary image would be submitted to `cvBlob` or `OpenCVBlobsLib`.
3) The `cvBlob` detects the connected regions in binary digital images. In each binary image, the white connected pixels are grouped together. These are called blobs. The library of the `cvBlob` provides the methods for tracking the moving object. The methods can provide the centroid position of every object and the identifier number of every moving object in a single screen.
4) A bounding box would be drawn into the images where the moving object is detected. The position of bounding box is obtained by minimal x, minimal y, maximal x, and maximal y point.
5) The tracking data of every blob would be written into a text file.

2.2. **Orthorectification.**

In the current approach, the vehicle trajectory is initially determined in the perspective plane. Then, the trajectory is mapped to an orthogonal plane, which provides the actual vehicle position.

The following transformation is used to map the data from the perspective plane to the road plane [17]. We consider a unit square S_1 and an arbitrary quadrilateral Q, which is governed by four corner points: x'_1, x'_2, x'_3, and x'_4. These four points have one-to-one relations with those four points in the unit square, see Figure 1. The following matrix $\mathbf{T}$

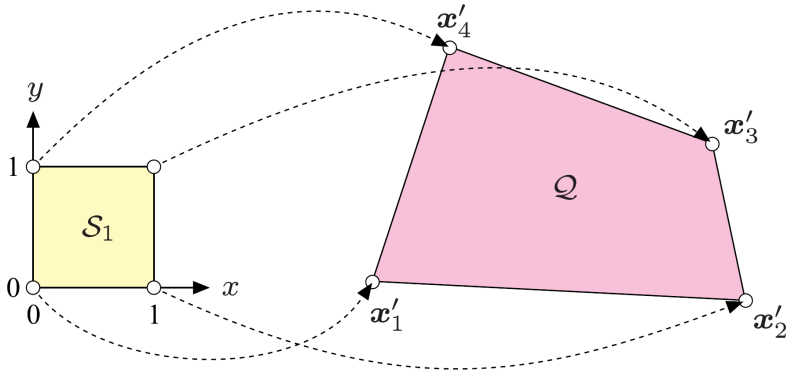

FIGURE 1. Projective mapping from the unit square S_1 to an arbitrary quadrilateral Q [17]

projects any point on the unit square to a point on the quadrilateral, or mathematically: $S_1 \xrightarrow{\mathbf{T}} Q$, where the transformation $\mathbf{T}$ is

$$\mathbf{T} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \tag{2}$$

where

$$a_{31} = \frac{(x'_1 - x'_2 + x'_3 - x'_4)(y'_4 - y'_3) - (y'_1 - y'_2 + y'_3 - y'_4)(x'_4 - x'_3)}{(x'_2 - x'_3) \cdot (y'_4 - y'_3) - (x'_4 - x'_3) \cdot (y'_2 - y'_3)},$$

$$a_{32} = \frac{(y'_1 - y'_2 + y'_3 - y'_4)(x'_2 - x'_3) - (x'_1 - x'_2 + x'_3 - x'_4)(y'_2 - y'_3)}{(x'_2 - x'_3) \cdot (y'_4 - y'_3) - (x'_4 - x'_3) \cdot (y'_2 - y'_3)},$$

$$a_{11} = x'_2 - x'_1 + a_{31}x'_2 \qquad a_{12} = x'_4 - x'_1 + a_{32}x'_4 \qquad a_{13} = x'_1,$$

$$a_{21} = y'_2 - y'_1 + a_{31}y'_2 \qquad a_{22} = y'_3 - y'_1 + a_{32}y'_4 \qquad a_{23} = y'_1 \qquad a_{33} = 1.$$

Inversely, we can project any point on the Q plane to the S_1 plane by the inverse of the transformation matrix $\mathbf{T}^{-1}$.

Now, we consider three planes, the perspective plane Q_1, the unit plane S_1, and the road plane Q_2, depicted in Figure 2. We can map points on the perspective plane Q_1 to points on the road plane Q_2 in two steps: $Q_1 \xrightarrow{\mathbf{T}_1^{-1}} S_1$, and followed by $S_1 \xrightarrow{\mathbf{T}_2} Q_2$. The transformation can be directed from the perspective plane Q_1 to the road plane Q_2 by:

$$\mathbf{T} = \mathbf{T}_2^{-1}\mathbf{T}_1. \tag{3}$$

2.3. **Numerical integration.** In the current work, the vehicle movement is also recorded using an accelerometer, which is placed inside the test vehicle. The recorded acceleration data are numerically integrated to provide the vehicle velocity and position. A comparison is made to these data to measure the accuracy of the current proposed method. The numerical integration is performed by use of the trapezoidal rule.

2.4. **Performance measure.** We assess the accuracy of a real-value vector $\hat{\mathbf{x}}$ with respect to a reference vector $\hat{\mathbf{x}}$ by

$$\text{Relative Error} = \frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|} \times 100\%, \tag{4}$$

where $\|\cdot\|$ denotes the Euclidean norm. In the current work, $\hat{\mathbf{x}}$ may denote the vehicle position tracked by a computer vision technique and $\mathbf{x}$ denotes the vehicle position obtained by a numerical integration of the accelerometer data.

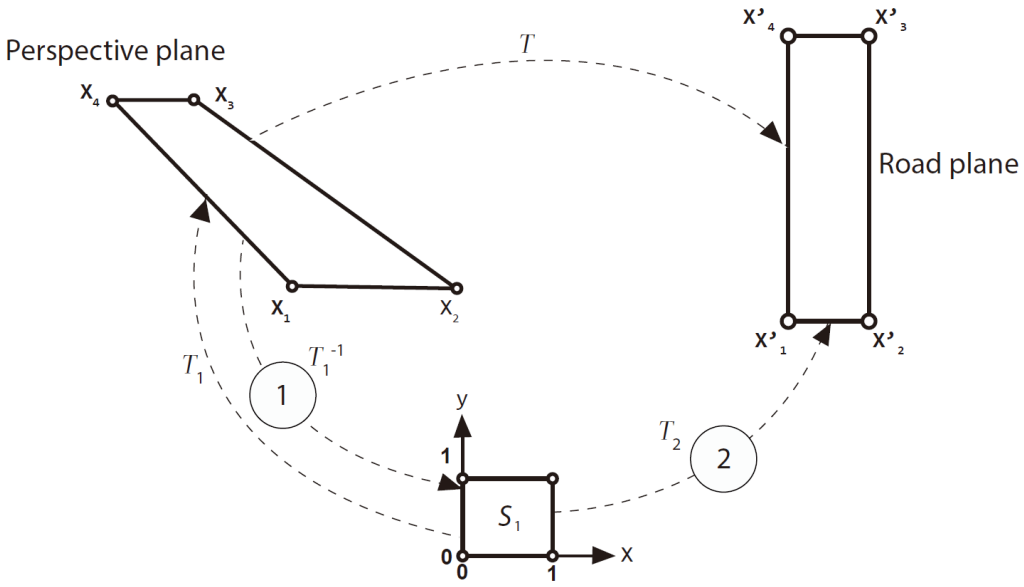

FIGURE 2. Two-step projective transformation between perspective plane and road plane. In the first step, the perspective plane Q_1 is transformed to the unit square S_1 by the inverse mapping function $\mathbf{T}_1^{-1}$. In the second step, $\mathbf{T}_2$ transforms the square S_1 to the road plane Q_2. Transformation from the perspective plane Q_1 to the road plane Q_2 can be directed by $\mathbf{T} = \mathbf{T}_1^{-1}\mathbf{T}_2$ [17].

3. **Results.** The proposed method is evaluated with a simple experiment described following. The experiment only involves a vehicle. The vehicle is set to travel along a straight trajectory for a distance around 23 m. It is difficult to control the vehicle exact position. Figures 3 and 4 respectively show the vehicle initial and final positions.

Four cones are placed on the four corners of the vehicle trajectory; see Figure 3. These four cones are separated by 22.9 m distance longitudinally and 3.5 m distance laterally. The four-cone positions are later important for orthorectification of the vehicle position.

The vehicle movement is recorded by two means: a video camera and an accelerometer. The video camera records the vehicle movement at the rate of 25 fps and image size of 720×576 pixels. The accelerometer records the data at the sampling time of 0.1 s.

The experiment of vehicle movement along the straight trajectory is repeated seven times. On each experiment, the vehicle is slowly accelerated from the initial position, and the vehicle break is exerted when the vehicle position is near to the intended final position. At the first four cases, the vehicle maximum speed is about 10 km/h. For the remaining cases, the maximum speed is about 20 km/h. This speed is estimated by the driver.

From the tracked vehicle position, we derive the vehicle's velocity and acceleration. For this purpose, firstly, we fit the vehicle position with a cubic spline function with smoothing. Then, the function is differentiated once and twice to provide the vehicle velocity and acceleration. Finally, the parameters m and l are obtained by minimizing the following error function:

$$\text{Error}(t) = a_n(t) - \alpha \frac{v_n^m(t) \cdot \Delta v_n(t - t_d)}{\Delta x_n^l(t - t_d)}. \tag{5}$$

Two sets of data are necessary for this study. The first set is obtained from the deployed computer-vision-based vehicle tracking methods. The second set is from the accelerometer.

FIGURE 3. The vehicle initial position and the four cones used for orthorectification

FIGURE 4. The vehicle final position

Two different background subtractions in computer vision will be used separately: multilayer- and Eigen-background-subtraction methods. The computer vision vehicle tracking provides the data of frame number, blob number, blob area, x blob-centroid, y blob-centroid, and the position of the bottom-right corner of the blob. The frame number and the frame rate data are used to calculate the time associated with the frame number by:

$$\text{Frame Time } t = \frac{\text{Frame Number}}{\text{Frame Rate}}.$$

The frame rate is fixed at 25 fps. Only one vehicle is used in the experiment and it has dimensions of 4.7 m length, 1.8 m width, and 1.8 m height.

Since the computer vision vehicle tracking provides many types of data, the question is: which data are good to determine the vehicle position. The candidates are rather clear; they are the centroid of the blob or its bottom-right corner position. We should note that the vehicle movement along its trajectory is subjected to a perspective view that makes

the vehicle looks bigger when its position is near the camera. This phenomenon is clearly depicted in the left panel of Figure 5, which shows that the blob area quickly increases with the frame number.

When using the multilayer-background-subtraction method, the blob area abruptly drops when the vehicle nearly reaches its final position. This is because only a big blob initially exists depicting the vehicle and this blob becomes two blobs having smaller size when the vehicle is about to stop. The blob area data after that time are not related to the vehicles but are noises. In the right panel of the figure, the evolution of the blob area, prior to the area of the blob drops, is fitted with a quadratic function. Theoretically, the blob area should increase quadratically with the frame number as the blob width and height increasing linearly. We obtain the level of fitness of $R^2 = 0.9785$, which indicates high level of agreement.

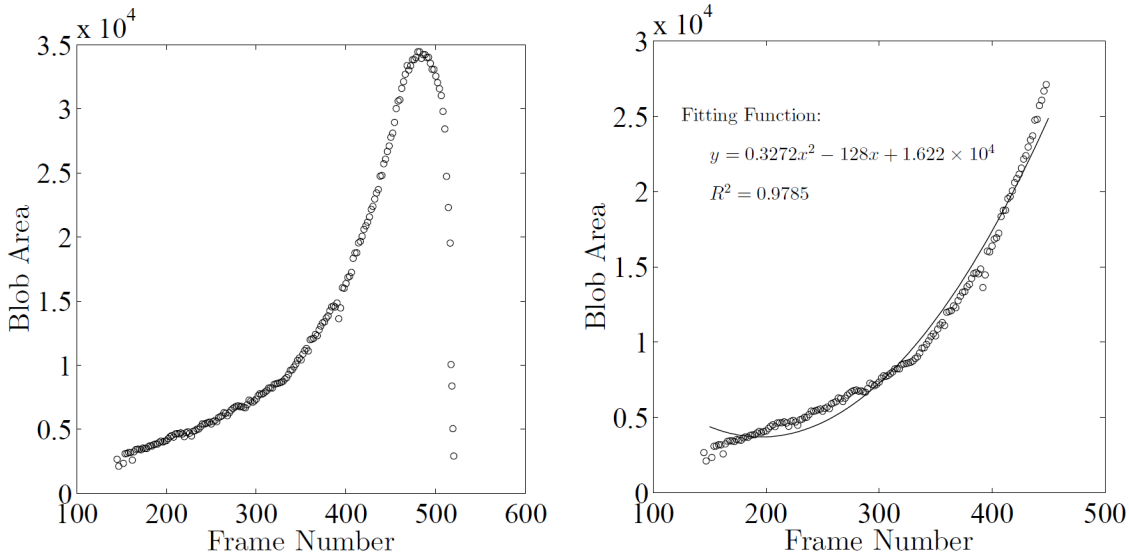

FIGURE 5. The evolution of the size of the blob area during the vehicle movement along a straight trajectory (left panel); a portion that is fitted with a quandratic equation (right panel)

The tracked vehicle position according to its blob centroid and blob right-bottom corner is shown in Figure 6 using the video data from the first six experiments. The results of the seventh experiment are not displayed for the compactness of the presentation. We should note that since the vehicle is moving on a straight path, its trajectory in the perspective plane should also be straight. We observe in the figure that the centroid trajectory and the bottom-right corner-point trajectory are rather straight. Few data are off from the trajectory particularly at the end of the trajectory when the tracked vehicle has stopped.

We fit the trajectories with a straight line and evaluate the fitness using the coefficient of determination R^2, which is computed by $R^2 = 1 - \mathrm{SS}_{\mathrm{res}}/\mathrm{SS}_{\mathrm{tot}}$, where $\mathrm{SS}_{\mathrm{res}} = \sum(y_i - \hat{y}_i)^2$, $\mathrm{SS}_{\mathrm{tot}} = \sum(y_i - \bar{y})^2$, y_i is the ith data, $\hat{y}$ is the data average, and $\hat{y}_i$ is the ith predicted data. The results are depicted in Table 1. These fitness data show that the tracking using the blob right-bottom corner point data is better in all of the tests in the sense that it has higher values of the coefficient of determination R^2. In the following step, the vehicle position in the perspective plane depicted in Figure 6 is projected to the actual road plane. This projection is achieved by use of the projection matrix $\mathbf{T}$, which is computed by Equation (3). Four corner points on the perspective plane are required to compute $\mathbf{T}_1$, and four corner points on the road plane are required for $\mathbf{T}_2$. Those points are tabulated

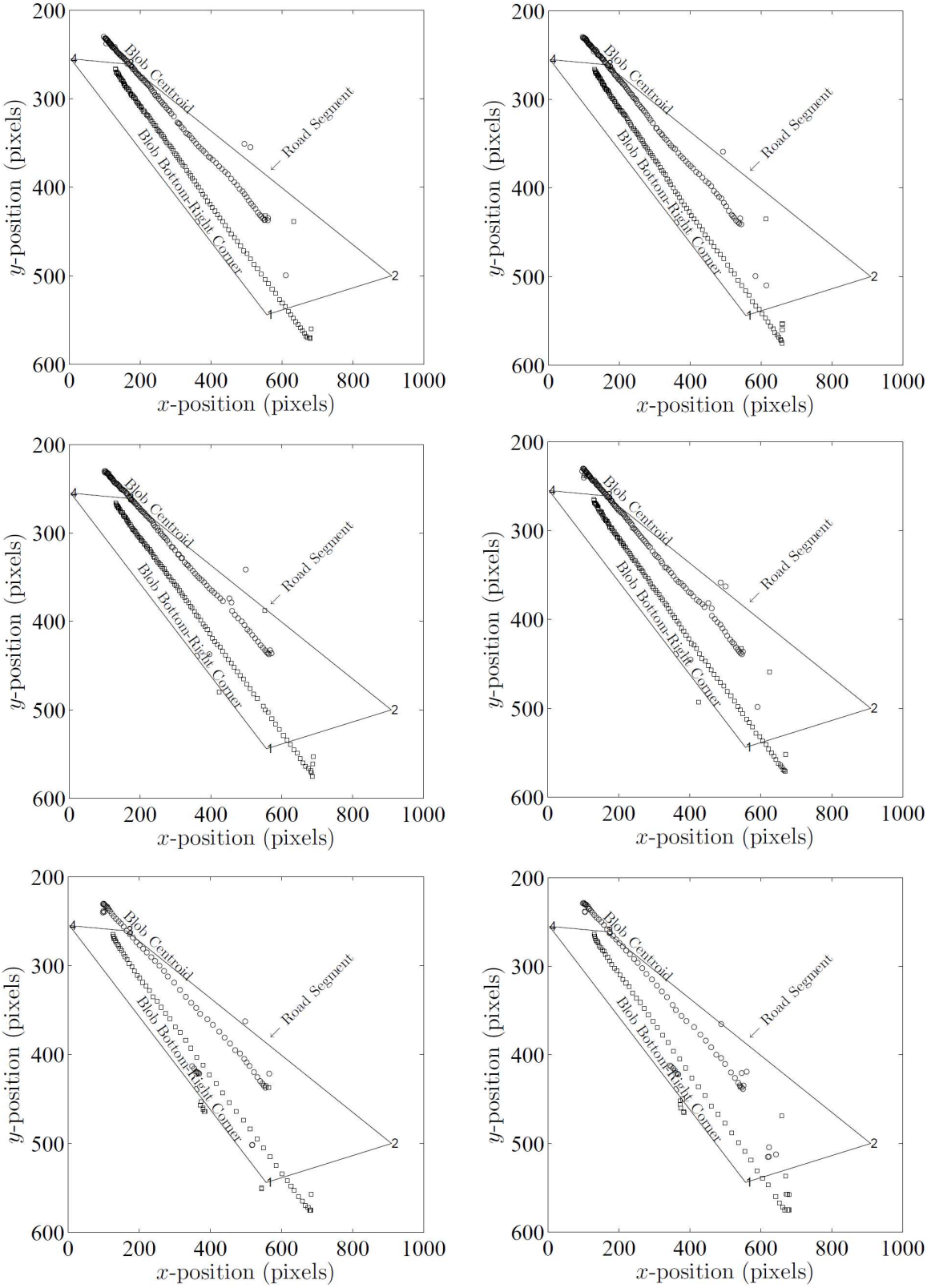

FIGURE 6. The comparison of the vehicle trajectory using blob centroid position and blob right-bottom corner position for the six experiments

TABLE 1. The level of fitness of the blob centroid position and blob right-bottom corner position with the straight line for the seven experiments

Experiment	Fitting Function		Fitness (R^2)	
	Corner Data	Centroid Data	Corner Data	Centroid Data
1	$y = 0.5445x + 199.5$	$y = 0.4519x + 185.8$	0.9914	0.9885
2	$y = 0.5630x + 197.5$	$y = 0.4726x + 183.7$	0.9854	0.9837
3	$y = 0.5422x + 197.5$	$y = 0.4523x + 183.7$	0.9894	0.9781
4	$y = 0.5555x + 198.1$	$y = 0.4663x + 184.2$	0.9843	0.9685
5	$y = 0.5450x + 208.3$	$y = 0.4678x + 188.8$	0.9626	0.8991
6	$y = 0.5313x + 207.7$	$y = 0.4826x + 182.0$	0.9437	0.9205
7	$y = 0.5570x + 201.5$	$y = 0.4688x + 184.5$	0.9739	0.9331

TABLE 2. The positions of the four cones in the image (pixel) and in the actual road (meter)

Cone No.	Corner Points (Cone Positions)			
	x (pixel)	y (pixel)	x (m)	y (m)
1	557	544	0.0	0.0
2	910	500	3.5	0.0
3	164	261	3.5	22.9
4	4	255	0.0	22.9

in Table 2. The results of this projection are position data in the actual road plane that can be seen in Figure 7.

The accuracy of computer vision vehicle tracking is measured by comparing the tracking positions and velocities to those of the accelerometer data. These results can be seen in Figures 8-11.

The estimated vehicle's position by the multilayer-background-subtraction method during the experiment is depicted in Figure 8. The figure also shows the vehicle position estimated by using the accelerometer sensor. In general, the estimated positions by the two methods agree very well for the majority of the experimental duration. However, discrepancy between the two methods exists during the initial time. We note that the test vehicle in the experiment is slowly accelerated from a rest condition. The accelerometer sensor clearly captures this process better than the vision-based method. Even, in a few cases, the vision-based method seems to suggest that the test vehicle is moving backward, which is misleading. At the initial time, the vehicle is positioned farthest from the camera. We expect that the estimated vehicle position will be rather low. We conclude that the discrepancy of the accuracy during the initial time is due to the error in the multilayer-background-subtraction method. Similar problem also occurs in the Eigen-background-subtraction method as presented in Figure 9.

The estimated vehicle velocity is depicted in Figure 10 for the multilayer-background-subtraction method and Figure 11 for the Eigen-background-subtraction method. In general, the discrepancy in predicting the velocity is much larger than that of the position.

The relative error, absolute error, and accuracy have been computed for each experiment; see Table 3. The results show that the tracking accuracy using the multilayer-background-subtraction method is better than the Eigen-background-subtraction method.

The multilayer-background-subtraction method has 96.6% of position accuracy and 88.9% of velocity accuracy; meanwhile the Eigen-background-subtraction method has 92.9% of position accuracy and 84.3% of velocity accuracy.

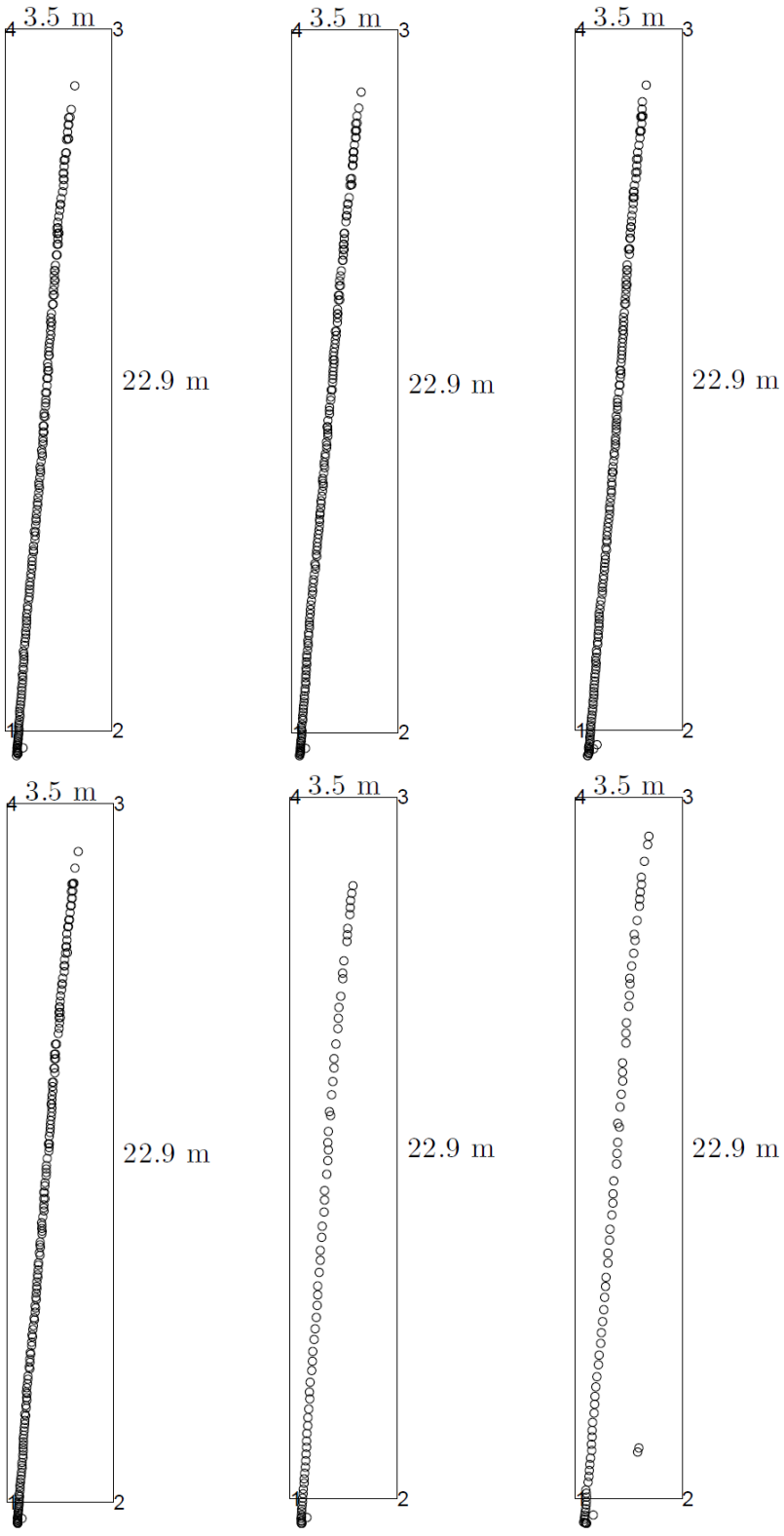

FIGURE 7. The reconstructed vehicle trajectory from the six experiments

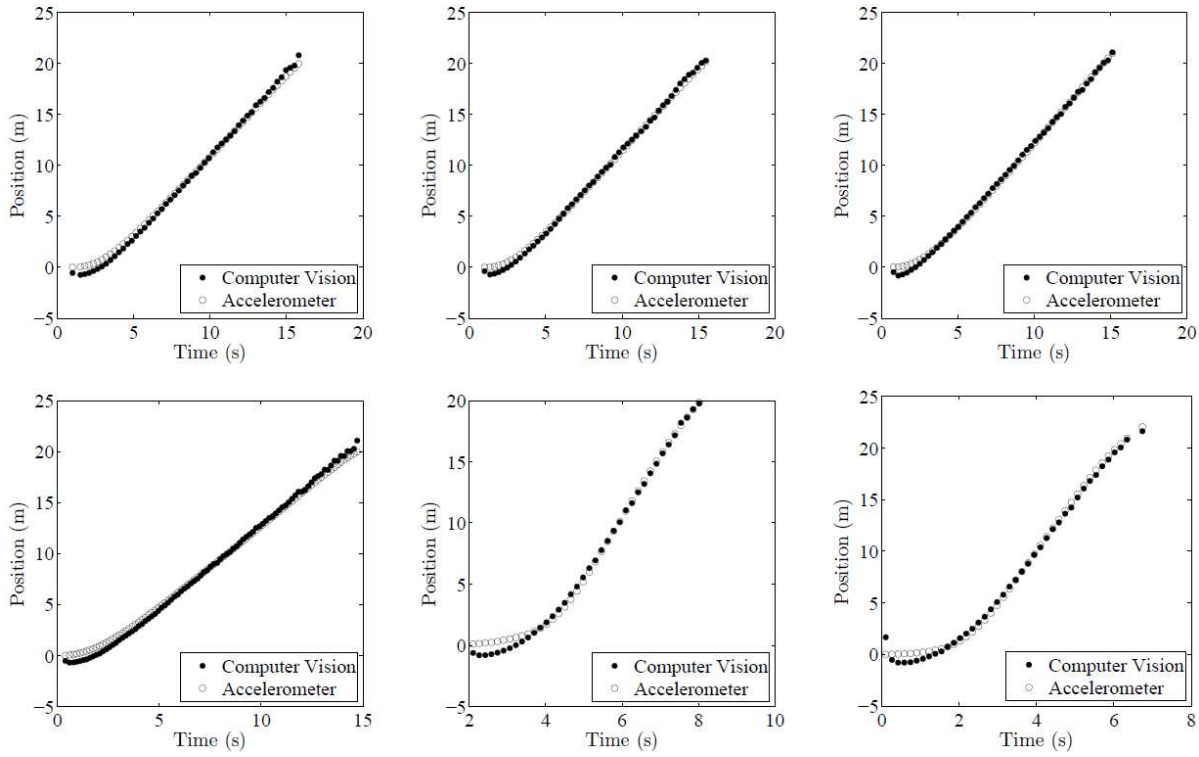

FIGURE 8. Comparisons of the vehicle position-time histories reconstructed by using accelerometer and computer vision (multilayer-background-subtraction method)

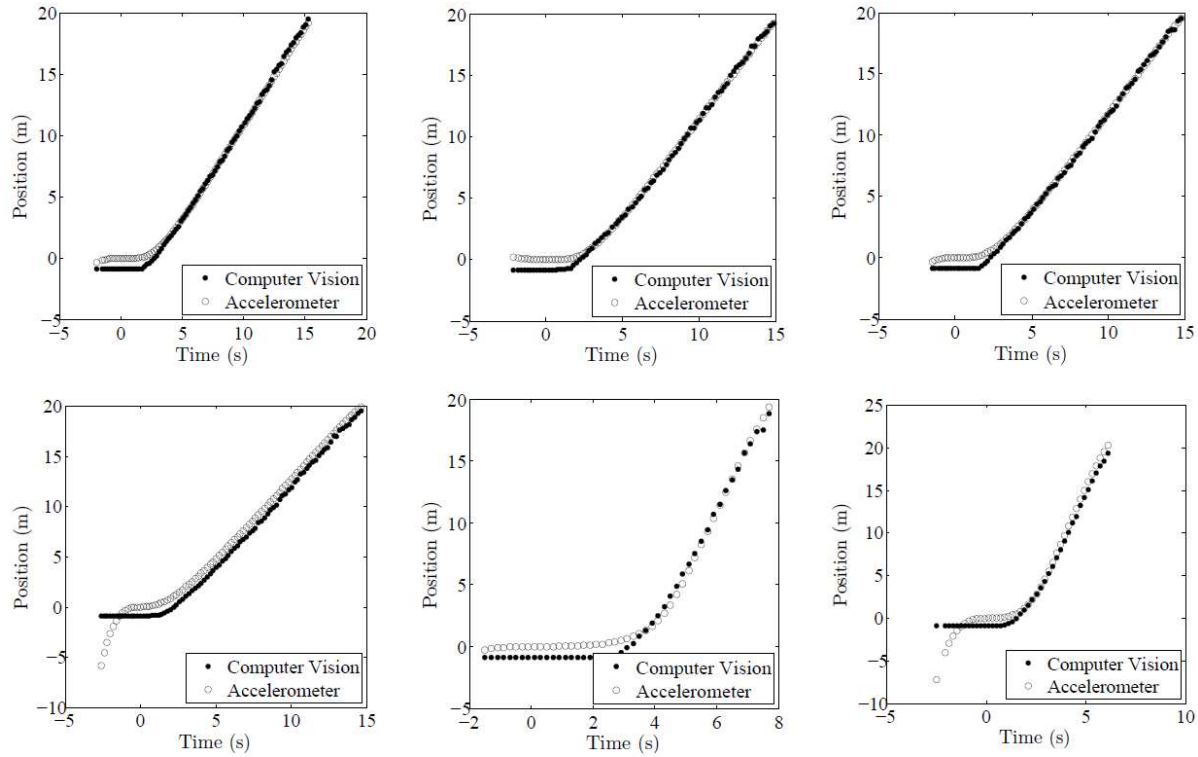

FIGURE 9. Comparisons of the vehicle position-time histories reconstructed by using accelerometer and computer vision (the Eigen-background-subtraction method)

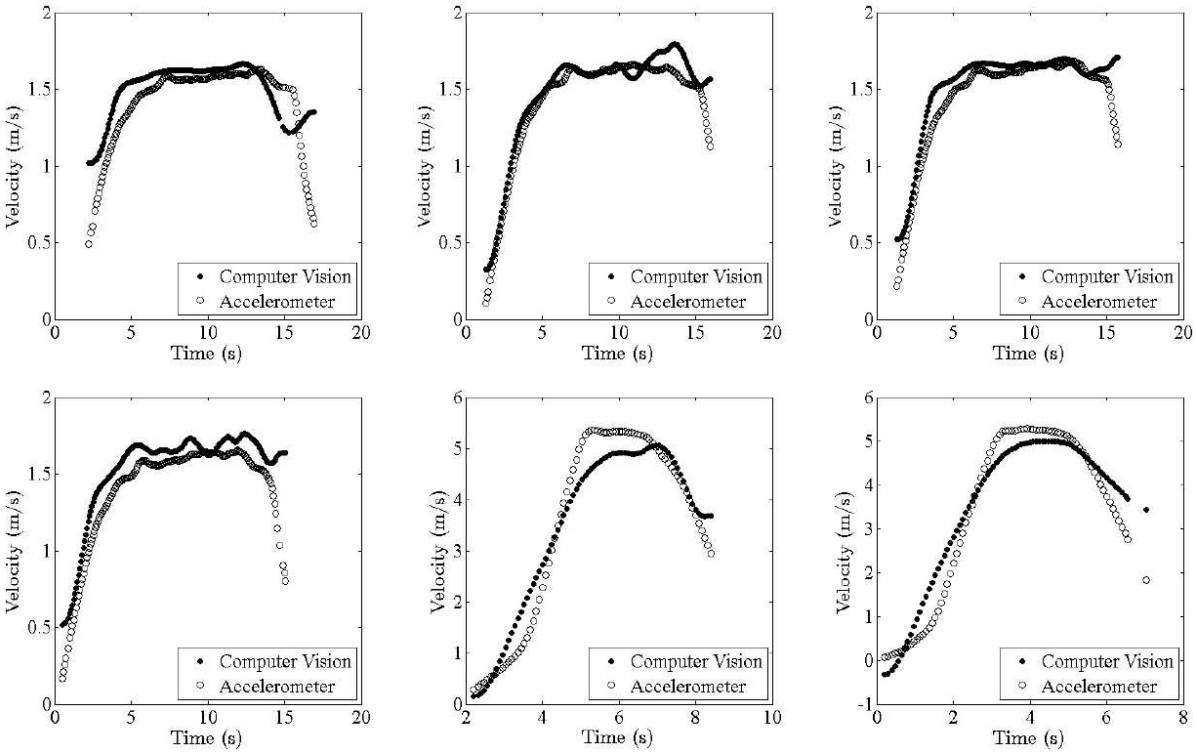

FIGURE 10. Comparisons of the vehicle velocity-time histories reconstructed by using accelerometer and computer vision (multilayer-background-subtraction method)

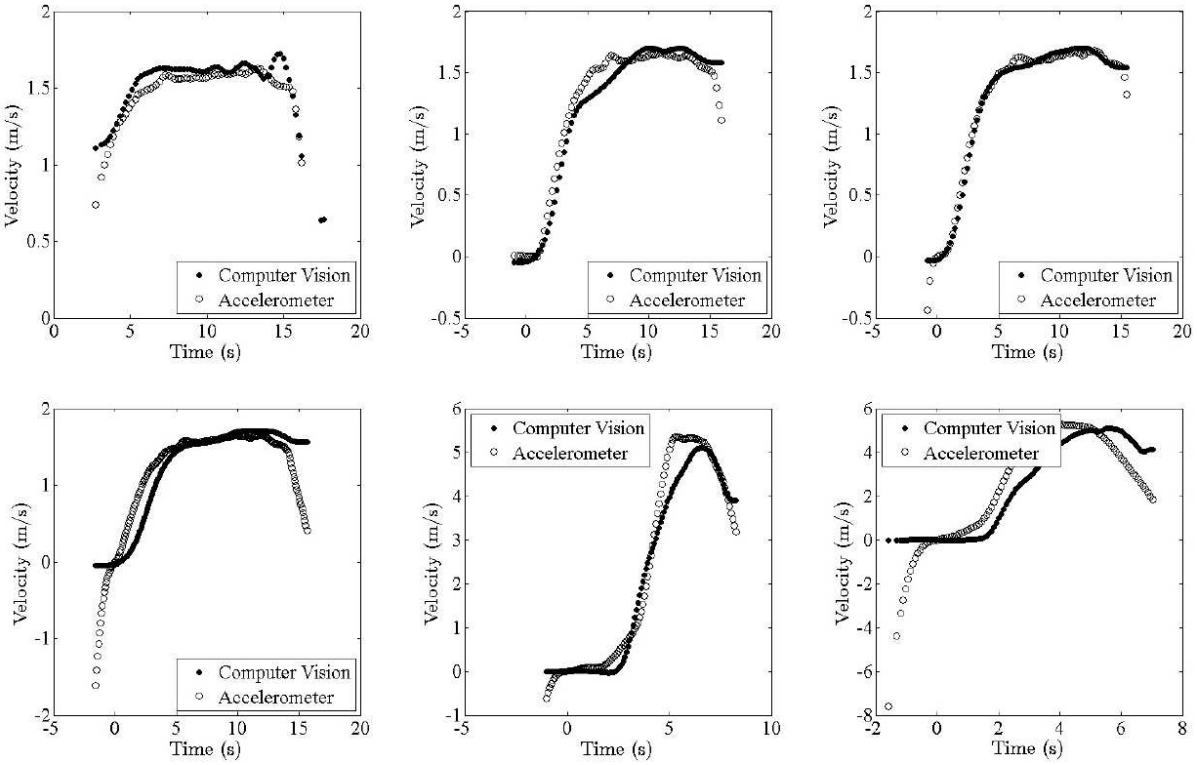

FIGURE 11. Comparisons of the vehicle velocity-time histories reconstructed by using accelerometer and computer vision (the Eigen-background-subtraction method)

TABLE 3. The relative error of the tracked vehicle position by the computer vision techniques (multilayer- and Eigen-background-subtraction methods) with respect to the vehicle position obtained by integrating the accelerometer data. The relative error is defined by Equation (4).

Experiment No	Relative Error (%)			
	Position (m)		Velocity (m/s)	
	Multilayer	Eigen	Multilayer	Eigen
1	3.41	2.9	13.1	6.9
2	2.49	4.6	6.5	9.1
3	2.40	4.4	8.4	4.0
4	3.84	10.5	12.5	26.3
5	4.37	9.4	10.9	13.1
6	4.30	12.7	12.5	39.3
7	3.07	5.2	13.5	11.2

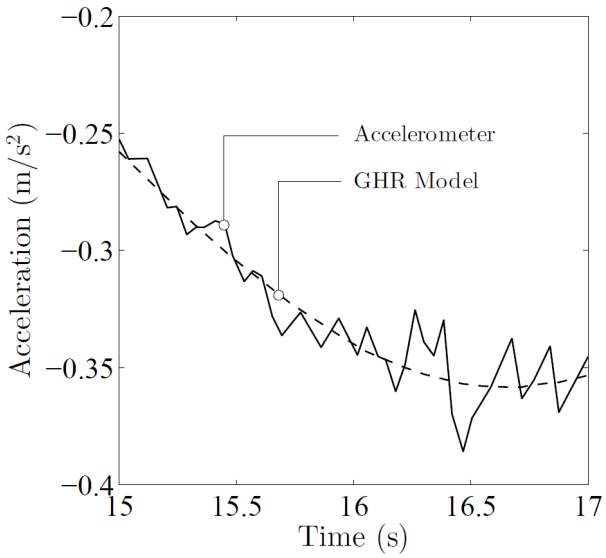

FIGURE 12. A comparison of the vehicle accelerations between accelerometer data and GHR model prediction

In Figure 12, we compare the vehicle acceleration computed by GHR model and the measured one. The estimated GHR car-following parameters are $m = 0.4$ and $l = 1.2$ at 3.2% of relative error. These estimated parameters are within the range of most reliable estimated parameters according to findings of other researchers for GHR model [18].

4. **Conclusions.** The car-following model has important application in traffic and safety engineering. Unfortunately, finding the model parameters are often costly. For the reason, this research proposes a significantly low-cost method to determine the model coefficients. In the current method, the vehicle movement is recorded digitally and tracked by a computer vision technique. The obtained vehicle position is projected to the road plane with an orthographic projection technique. In the current experiment, to evaluate its accuracy, the method is used to track trajectory of a vehicle moving in a short-straight lane. From comparisons to the data obtained from an accelerometer, it is concluded that the current method is reasonably accurate. The results show that the

tracking accuracy of the multilayer-background-subtraction method is better than the Eigen-background-subtraction methods. The multilayer-background-subtraction method has 96.6% of position accuracy and 88.9% of velocity accuracy; meanwhile, the Eigen-background-subtraction method has 92.9% of position accuracy and 84.3% of velocity accuracy. The method estimates the car-following parameters to be $m = 0.4$ and $l = 1.2$ with 3.2% relative error. These estimated parameters are within the range of the most reliable GHR model parameters according to many references.

In the current work, the tracked car position is compared with the position obtained by integrating the car acceleration recorded by an accelerometer sensor. Both measurement methods clearly contain errors from different sources. The accelerometer does not only record the car acceleration but also its vibration. Therefore, the assessment of the accuracy of the proposed method, which is established on the basis of the accelerometer data, is not entirely accurate, and is compromised by the error. For future research, the accuracy of the method should be assessed on the basis of the data collected by a more reliable method such as an RTK GPS.

REFERENCES

[1] BPS, *Increasing in the Number of Vehicles According to Types during 1987-2012*, http://www.bps.go.id/linkTabelStatis/view/id/1413, 2013.

[2] S. Morichi, Long-term strategy for transport system in Asian megacities, *Journal of the Eastern Asia Society for Transportation Studies*, vol.6, pp.1-22, 2005.

[3] F. E. Gunawan, Empirical assessment on factors affecting travel time of bus rapid transit, *International Journal of Engineering and Technology*, vol.7, no.1, pp.327-334, 2015.

[4] F. E. Gunawan, Suharjito and A. A. S. Gunawan, Simulation model of bus rapid transit, *International Conference on Advanced Science and Contemporary Engineering*, Jakarta, Indonesia, 2013.

[5] J. Lumentut, F. E. Gunawan, W. Atmadja and B. S. Abbas, A system for real-time passenger monitoring system for bus rapid transit system, *The 7th Asian Conference on Intelligent Information and Database Systems (ACIIDS)*, Bali, Indonesia, 2015.

[6] J. S. Lumentut, F. E. Gunawan and Diana, Evaluation of recursive background subtraction algorithms for real-time passenger counting at bus rapid transit system, *Intenational Conference on Computer Science and Computational Intelligence (ICCSCI2015)*, Bina Nusantara University, Jakarta, Indonesia, 2015.

[7] D. Walsh, D. Stewart, J. Luk and J. Tay, The use and application of microsimulation traffic models, *Austroads Publications Online*, https://www.onlinepublications.austroads.com.au/items/AP-R286-06, 2006.

[8] J. Ryu, C. Kim, M. Chang, Y. Kim and S. Bae, Simulation and speed classification of car following models, *Proc. of the Eastern Asia Society for Transportation Studies*, vol.3, no.2, pp.123-133, 2001.

[9] F. E. Gunawan, Two-vehicle dynamics of the car-following models on a realistic driving condition, *International Journal of Transportation Systems and Information Technology*, vol.12, no.2, pp.77-83, 2012.

[10] R. Chandler, R. Herman and E. Montroll, Traffic dynamics: Studies in car following, *Operation Research*, vol.6, p.165, 1958.

[11] S. Hoogendoorn, H. V. Zuylen, B. Gorte, G. Vosselman and M. Schreuder, Microscopic traffic data collection by remote sensing, *Transportation Research Record*, vol.1855, pp.121-128, 2007.

[12] S. Hoogendoorn and M. Schreuder, Tracing congestion dynamics with remote sensing: Toward a robust method for microscopic traffic data collection, *The 84th Annual Meeting of the Transportation Research Board*, Washington, D.C., 2005.

[13] P. Ranjitkar, T. Nakatsuji and A. Kawamura, Car-following models: An experiment based benchmarking, *Journal of the Eastern Asia Society for Transportation Studies*, vol.6, pp.1582-1596, 2005.

[14] Crist, *Blob Library for Opencv. Google Code*, https://code.google.com/p/cvblob/wiki/FAQ, 2012.

[15] J. Yao and J. M. Odobez, Multi-layer background subtraction based on color and texture, *IEEE Conference on Computer Vision and Pattern Recognition*, no.1, pp.1-8, 2007.

[16] N. M. Oliver, B. Rosario and A. P. Pentland, A Bayesian computer vision system for modeling human interactions, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.22, no.8, pp.831-843, 2000.

[17] W. Burger and M. Burge, *Principles of Digital Image Processing: Core Algorithms*, Springer-Verlag, 2009.
[18] M. Brackstone and M. McDonald, Car-following: A historical review, *Transportation Research Part F*, vol.2, pp.181-196, 1999.