Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 157 (2019) 255–265

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), 12–13 September 2019

# Detection of Vehicle Position and Speed using Camera Calibration and Image Projection Methods

Alexander A S Gunawan[a,*] , Deasy Aprilia Tanjung[a], Fergyanto E. Gunawan[b]

[a]*Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480*
[b]*Industrial Engineering Department, BINUS Graduate Program - Master of Industrial Engineering, Bina Nusantara University, Jakarta, Indonesia 11480*

**Abstract**

Traffic congestion is the main problem faced by big cities, such as Jakarta. One approach to reduce congestion levels is to improve traffic management that regulates and controls the number of vehicles. To evaluate the impact of traffic management before direct implementation on the highway, traffic modeling can be carried out. Parameters in modeling traffic must be determined from a calibration process where the vehicle is accurately measured for its position and speed. This study aims to propose an efficient calibration procedure with accurate results, based on recorded vehicle movement in perspective view. First, the road image is projected using the Direct Linear Transformation (DLT) method, then the vehicle position is detected using the Background Subtraction and tracked using Mixture of Gaussian (MoG) to determine the vehicle speed. Finally, we develop a prototype of Automated Traffic Flow Monitoring based on Python programming. In the experiment results, the accuracy of vehicle position detection is evaluated based on the Euclidean distance. The average difference between the results of position detection with ground-truth is 12.07 pixels with a camera angle 40 °. The percentage of speed measurement accuracy using the DLT projection method is 96.14%.


Peer-review under responsibility of the scientific committee of the 4th International Conference on Computer Science and Computational Intelligence 2019.

*Keywords:* Music recommender system; Convolutional recurrent neural network; Similarity distance;

* Corresponding author. Tel.: +62-21- 5345830.
*E-mail address:* aagung@binus.edu

1877-0509 © 2019 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the 4th International Conference on Computer Science and Computational Intelligence 2019.
10.1016/j.procs.2019.08.165

## 1. Introduction

Traffic congestion is the main problem faced by Jakarta city as a metropolitan which is the center of the economy and trade in Indonesia. One approach to reduce congestion levels is by improving traffic management where the level of traffic is regulated and controlled. The traffic management often requires forecasting of the impacts of implementing various traffic management regulation by conducting traffic modeling [1]. Traffic modeling is means to evaluate the effects of the traffic management interventions prior the actual the implementation. Parameters in modeling traffic must be determined from a calibration process to record the vehicle position and speed accurately. This study aims to create an efficient calibration procedure with accurate results based on based on recorded vehicle motions in video camera. The vehicle is detected and tracked using computer vision techniques. In here, we concern to increase the accuracy of vehicle position and speed results. First, the road image is projected using the Direct Linear Transformation (DLT) method, then the vehicle position is detected using the Background Subtraction and tracked using Mixture of Gaussian (MoG) to determine the vehicle speed.

Advanced Traffic Management System (ATMS) is one of the developments in Intelligent Transportation System (ITS). ATMS's input data is obtained by CCTV and provides real-time information about traffic to road users [2]. Real-time traffic data obtained from CCTV videos needs to be processed first so that it becomes useful information. With the calibration procedures developed in our study, accurate lower-cost real-time traffic data processing can be processed by the Traffic Management Center (TMC) for monitoring traffic on the road. Therefore, it can reduce the accidents and traffic congestion. One approach for implementation can be found in Osman et al work [3] which aimed to produce a speed detection camera system or Speed Detection Camera System (SDCS) as an alternative to radar. SDCS can be divided into several steps, that is: object detection, tracking objects and object speed calculations. Fig. 1 shows the design of SDCS to estimate the vehicle speed.

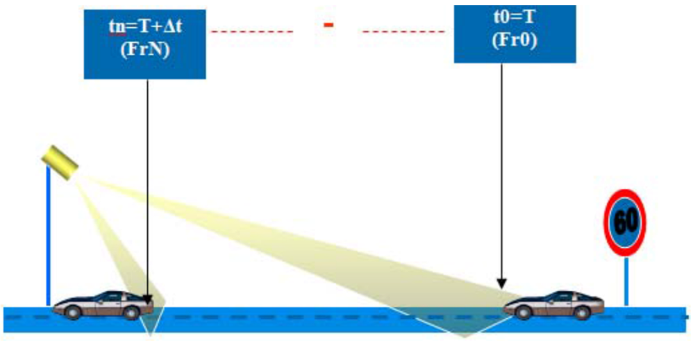



Fig. 1. The design of Speed Detection Camera System (SDCS)

Our initial research [4] aimed to find the coefficients of the car-following model at a lower cost. While all the existing calibration methods need many preparations on the equipment and the cost is also very expensive. In that study, vehicle movements were digitally recorded and tracked with computer vision techniques: multilayer and Eigen background subtractions. This method is used to track the trajectory of a vehicle moving on a short straight line. Next, the obtained position of the vehicle is projected on the road plane by using orthographic projection technique [5]. The camera is used to record the movement of a vehicle passing a straight line which is limited by four points from a certain height. The position data of the four points is used to project data from the image to the road plane. In here, we would like to create an efficient calibration procedure by improving the image rectifying algorithm. Therefore, the research focus is to construct the better projection technique.

This research is inspired by Wang et al work [6] which aims to calculate the estimated speed of pedestrians using Direct Linear Transformation (DLT) method [7]. In the study, the calculation of pedestrian speed can be equivalent to analyzing the traces of an object movement. The problem can be analyzed in two parts. The first part is the change of coordinates between the world coordinate system and the coordinate system in the image obtained through the "Distance-to-Pixel" mapping table. The second part is by calculating the actual movement distance. The mapping table is obtained by calibrating the camera using the DLT method. After that, trajectory in 3D space can be obtained by changing the trajectory into an 2D image and divided into several parts so that each part is nearly linear. The estimated speed is the average of the speed in each linear part. Our research focus is different to other researches [8, 9] because we try to construct more better projection algorithm in here. It is not directly related to increase the prediction of vehicle speed.

We follow Wang et al [6] in using DLT method with the aim of analyzing calibration and projection methods, consequently we wish the vehicle position detection and vehicle speed calculations will be more accurate. Camera calibration is a method of settling the camera model, which is a geometric relationship between 3D space and 2D image. In here, the purpose of camera calibration was to calculate the target position in real 3D space. Our algorithm starts by projecting the road image using the Direct Linear Transformation (DLT) method, then the vehicle position is detected using the Background Subtraction and tracked using Mixture of Gaussian (MoG) to determine the vehicle speed. Finally, we develop a prototype of Automated Traffic Flow Monitoring based on Python programming and its libraries, such as OpenCV and Kivy.

The remainder of the paper is composed as follows: first we discuss Direct Linear Transformation (DLT) in section 2, and then is followed by explanation of Automated Traffic Flow Monitoring, in section 3. In section 4, we discuss the experiment results to analyze calibration and projection process. Finally, we concluded our work with suggestions for the future research in section 5.

## 2. Direct Linear Transformation (DLT)

There are many camera calibration methods, which their requirements vary along with the application scene. In 1971, Direct Linear Transformation (DLT) method was first proposed by Abdel-Aziz YI et al [10]. DLT is a traditional camera calibration method based on homogenous coordinate and without considering object distortion. It solves a set of variables from a set of similarity relations by solving system of linear equations. What makes DLT problem distinct from system of linear equations case is the fact that the right and left sides of the defining equation can differ by an unknown multiplication factor [7]. In order to be able solved as system of linear equations, the similarity relations are rewritten as proper linear homogeneous equations. The combination of rewriting the similarity equations as homogeneous linear equations and solving them by standard methods of linear equation system is called as DLT algorithm.

In camera calibration, the relation is between 3D points in a scene and their projection onto the image plane of camera. Nevertheless, the vertical coordinate is always constans in our case, thus we can regard 3D points as just 2D points. It simplifies the camera problem being a matter of homography. If given a pair of point mapping: 2D to 2D $\{x_i \Leftrightarrow x_i'\}$ as much as $n,\ n\ \geq 4$, it can be determined H matrix [7] that fulfills $x_i' = Hx_i$ , where:

$$H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} = \begin{bmatrix} \boldsymbol{h^{1T}} \\ \boldsymbol{h^{2T}} \\ \boldsymbol{h^{3T}} \end{bmatrix} \qquad \boldsymbol{x_i'} = \begin{bmatrix} x_i' \\ y_i' \\ w_i' \end{bmatrix} \qquad \boldsymbol{x_i} = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \tag{1}$$

We can rewrite the equation $x'_i = Hx_i$ in cross-product form $x'_i \times Hx_i = 0$ as follow:

$$x'_i \times Hx_i = \begin{pmatrix} y'_i\boldsymbol{h^{3T}}x_i - w'_i\boldsymbol{h^{2T}}x_i \\ w'_i\boldsymbol{h^{1T}}x_i - x'_i\boldsymbol{h^{3T}}x_i \\ x'_i\boldsymbol{h^{2T}}x_i - y'_i\boldsymbol{h^{1T}}x_i \end{pmatrix} \tag{2}$$

The above equation gives a set of three equations for H in the following equation:

$$\begin{bmatrix} 0^T & -w_i'x_i^T & y_i'x_i^T \\ w_i'x_i^T & 0^T & -x_i'x_i^T \\ -y_i'x_i^T & x_i'x_i^T & 0^T \end{bmatrix} \begin{pmatrix} \boldsymbol{h^1} \\ \boldsymbol{h^2} \\ \boldsymbol{h^3} \end{pmatrix} = 0 \tag{3}$$

It should be noted that only two linear ones are independent in above set of three equations. Thus, it is standard practice while using the DLT algorithm to ignore the third equation while solving for H, as follow:

$$\begin{bmatrix} 0^T & -w'_i x_i^T & y'_i x_i^T \\ w'_i x_i^T & 0^T & -x'_i x_i^T \end{bmatrix} \begin{pmatrix} \boldsymbol{h}^1 \\ \boldsymbol{h}^2 \\ \boldsymbol{h}^3 \end{pmatrix} = 0 \tag{4}$$

This gives us the equation $A_i \boldsymbol{h} = 0$, where $A_i$ is $2 \times 9$ matrix. This equation applies to all involved homogeneous coordinates. Each correspondence $\{x_i \Leftrightarrow x_i'\}$ gives two independent equations for homograph H. Given $n \geq 4$ correspondence will get a set of $2n$ equations $\mathbf{Ah} = 0$ where $\mathbf{A}$ is built from row matrix $A_i$. Next, to solve $2n \times 9$ $\mathbf{A}$ matrix, we can calculate Singular Value Decomposition (SVD) of $\mathbf{A}$ matrix and take an eigenvector that produces the smallest singular value as a solution and thus determine the homograph H which governs mapping between $x_i$ and $x'_i$.

The DLT method aims to obtain an H matrix that is used to map source points to destination points. In brief, the algorithm of DLT method can be described as follows:

1. The input is an array of source points (fp) and an array of destination points (tp). We must validate the size of both arrays must be the same.
2. Normalize fp and tp arrays, so that the means are zero and the standard deviations are one. To do normalization process for fp array, we calculate a $3 \times 3$ C1 matrix with the main diagonal containing $[1/maxstd, 1/maxstd, 1]$ with $maxstd$ is standard deviation of fp array in $x$ and $y$ axes. The value of $C1_{13}$ is obtained by dividing $(-m_{11})$ with $maxstd$ and the value of $C1_{23}$ obtained by dividing $(-m_{21})$ with $maxstd$. To get the normalization of fp array, we perform the dot product operation between C1 and fp. The same procedure is carried out to normalize tp array.
3. Construct the $2n \times 9$ $\boldsymbol{A}$ matrix, with $n$ is the number of points of fp.
4. Factorize the A matrix by using SVD method to get matrix of eigen vectors $V$as follows: $\boldsymbol{A} = USV^T$.
5. Construct homograph matrix H from last row of $V$ matrix, which is an eigenvector that produces the smallest singular value, in form a $3 \times 3$ matrix.
6. Do decondition to the obtained H matrix by multiply it to $C1_{fp}$ and $C1_{tp}$.
7. Finally, we normalize the H matrix by dividing each element with $H_{33}$.

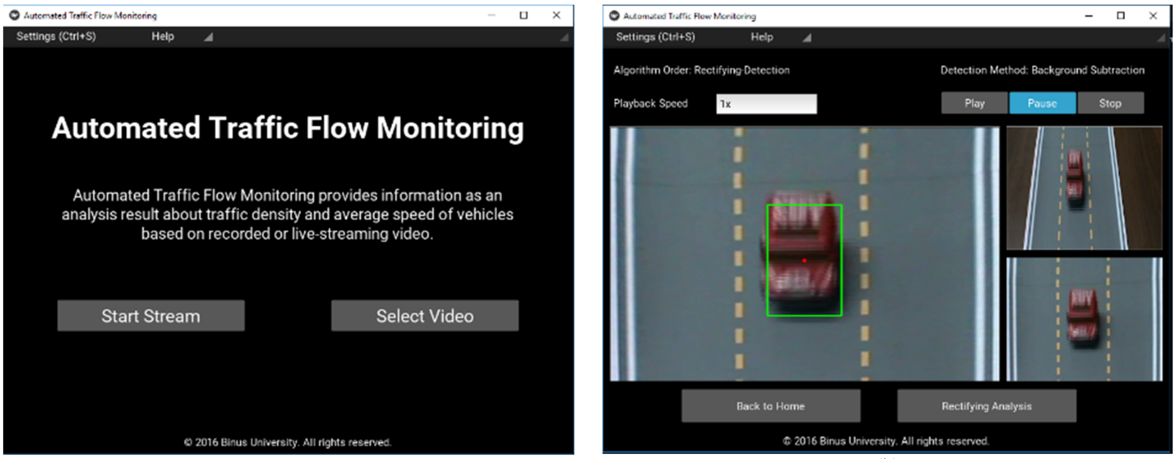
## 3. Automated Traffic Flow Monitoring

Automated Traffic Flow Monitoring (ATFM) is a desktop application that serves to display the process of calibrating camera for traffic simulation to obtain information on the average speed of a vehicle (see Fig 2a). It is based on Python programming and its libraries, such as OpenCV and Kivy. In ATFM, the recorded traffic video will be converted into frames and displayed after DLT processing. Projection transformation processing using DLT method consists of normalization of points, arranging the matrix $\mathbf{A}$ using corresponding points, and factoring matrix $\mathbf{A}$ using SVD. Finally, the last line on the matrix V is formed into a homograph matrix H. The H matrix will be used for image warping to obtain destination points.

The output of video processing are frames that have been projected with the DLT method, including the contour area of the vehicle object and vehicle position (see Fig 2b). For initial setting, it is used 4 source points that have been determined in x and y coordinates. Projection video will look as recorded from the top of the road, and lines on the left and right of the road will appear parallel (see Fig 3a). In addition, there is result analysis, which shows the information about the position track of vehicle and its average speed (see Fig 3b). In the settings menu, the user can specify the number of known source points to form parallel lines. The number of source points provided on the analysis system of ATFM is 4, 8, 12, 16, and 32 points. If the user determines the number of source points based on the standard setting of the system, the destination points will be obtained from the corresponding system automatically. Furthermore, users can also fill in input points for x and y coordinates manually and they will be stored as an array. If the user fills in input points manually, the source points will be sorted by x and y coordinates to top left, top right, bottom right and bottom left, then the destination points will be calculated corresponding to source points based on frame width and height.

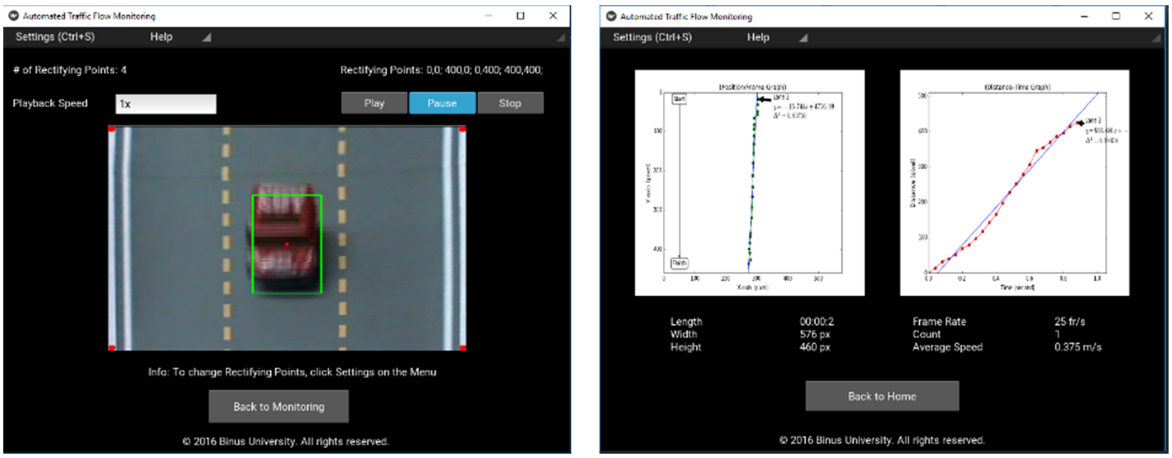
Real-time segmentation of moving regions in image frames is an essential step in vehicle detection. One of renown methods for detecting moving objects is background subtraction. One of the successful background subtraction methods is background mixture model [11], which is a Gaussian mixture-based background/foreground segmentation algorithm. It proposed a method to construct background pixel model by a mixture of K Gaussian distributions (K = 3 to 5). By analyzing the weights of the mixture, which represent the time of colors in the scene, we can obtain the background, which colors stay longer and more static. Furthermore, adaptive background mixture model [12] is proposed to improve the previous method by selecting the appropriate number of gaussian distribution for each pixel automatically. Recursive equations that are used to constantly update the parameters of a Gaussian mixture model. This method gives better adaptability to facing illumination changes in the scenes. To detect the vehicle, we used adaptive background mixture model [12] by finding foreground mask in each frame. The foreground mask is obtained from the current frame minus the background model, then the different result is filtered by using a threshold. For moving objects, the obtained contour is combined with its convex hull. Finally, vehicle detection is performed at the midpoint of the bounding rectangle of convex hull for each frame (see Fig 3a).

The results of vehicle detection will be displayed as position curve in the x-y axis and as distance curve after the vehicle detection data is filtered from outliers. Based on the distance curve, we can calculate a trendline equation by using the first-order curve fitting. The average speed of a vehicle is obtained by derivation of the trendline equation over time (see Fig 3b).




(a) (b)

Fig. 2. Automated Traffic Flow Monitoring (ATFM) (a) Home (b) Monitoring view




(a) (b)

Fig. 3. The results of ATFM (a) rectifying result (b) detection result

## 4. Experiment Results

### *4.1. Analysis of Number of Projection Points*

DLT method produces different H matrix values following the number of input points. Therefore, we conducted an analysis to find out the optimal number of input points for obtaining the most accurate results. To analyze and determine the number of the most efficient projection points, we have to specify the reference image. The reference image is created using Adobe Photoshop CS6 and the input image is obtained by projected it with 10° perspective angle. The reference and input image have 480 × 720 pixels with 5 pixels line size (see Fig 4a). Next, we record the position of the intersection points in the input and reference image. These points are used for the source points (fp) and the destination points (tp) in DLT algorithm. The example of projection image from DLT method with 14 input points is shown in Fig 4b.

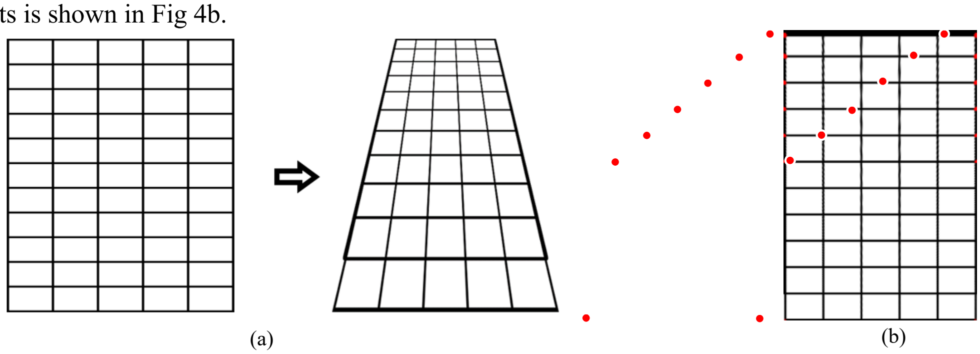



Fig. 4. (a) The reference image (left) and the input image (right) (b) projection result of DLT Method with 14 input points

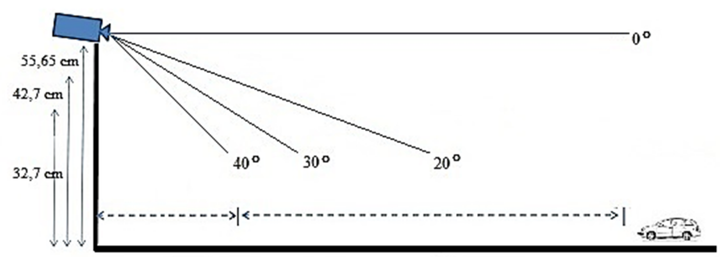
We analyze the projection images from DLT method by comparing to the reference image using Structural Similarity Index Metric (SSIM). SSIM [13] is used for measuring the similarity between two images, which considers image degradation as perceived change in structural information. The difference with respect to Mean Square Error (MSE) is this approach estimate absolute errors, which is not scale invariance as is SSIM. We make experiments with 4 to 24 input points, and compare the projection results to the reference image by using SSIM (see Table 1). The sequence of adding points starts from the top left, upper right, lower right, and lower left respectively. It is also showed the processing time to compute DLT algorithm in our computer. In Table 1, the SSIM values increase according to the addition of the number of input points. The bigger SSIM value, the more similar the projection imageto the reference image. SSIM value equals to 1 means both images are exactly the same. On the other hand, the addition of the number of input points will cause longer processing time. Based on Table 1, the optimal number of input points to get the most accurate result is 24 points with a processing time of 0.066 seconds.

Table 1. SSIM value and processing time corresponding to number of input points

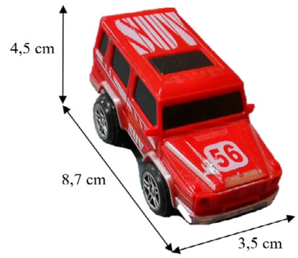
| Number of input point | SSIM | Time (sec) |
|---|---|---|
| 4 | 0,69 | 0,030 |
| 6 | 0,74 | 0,033 |
| 8 | 0,75 | 0,035 |
| 10 | 0,80 | 0,035 |
| 12 | 0,81 | 0,040 |
| 14 | 0,84 | 0,046 |
| 16 | 0,85 | 0,050 |
| 18 | 0,86 | 0,052 |
| 20 | 0,86 | 0,058 |
| 22 | 0,86 | 0,061 |
| 24 | 0,87 | 0,066 |

### 4.2. Analysis of Projection on Simulation Videos

In order to evaluate DLT algorithm when detecting vehicle, we conducted a controlled experiment by making simulations using toy cars and printed roads using several A4-sized HVS papers. The sketch of the experiment and the toy cars can be seen in Fig. 5a and 5b. The angle and height variations of camera setting which carried out in experiment are also shown in Fig 5a. In addition, we use a tripod and Sony DCR-SX65 camcorders with a resolution of 576 × 460 pixels to record simulation videos.




(a)




(b)

Fig. 5. (a) The experiment sketch with several camera setting (b) toy car as experiment object

There are 4 video simulation for our controlled experiment. Video 1 and video 2 are recorded with the same camera angle to determine the effect of camera height on the projection results. While video 1 and video 3 are recorded with the same camera height to determine the effect of the camera angle. Video 4 has a role as a comparison to other videos, is recorded with the camera angle and height that might give the most optimal projection results.

Table 2. Variations of camera angle and height used in controlled experiments

| Video name | Camera angle (degree) | Camera height (cm) |
|---|---|---|
| Video 1 | 20 | 32,7 |
| Video 2 | 20 | 42,7 |
| Video 3 | 40 | 32,7 |
| Video 4 | 30 | 55,65 |

With the difference in camera angle and height, the shape and length of the road that appears on the camera is different. Fig. 6 shows the shape of the road that appears on the first frame of each video with the camera angle and camera height according to Table 2 respectively. We need source and destination points to make projections from video input into video projection. In our experiments, source points consist of 4 points, namely upper left, upper right, lower right, and lower left. Each video will be projected on the same destination points, that is: upper left (2, 2), upper right (574, 2), lower right (574, 458), and lower left (2, 458).

(1)

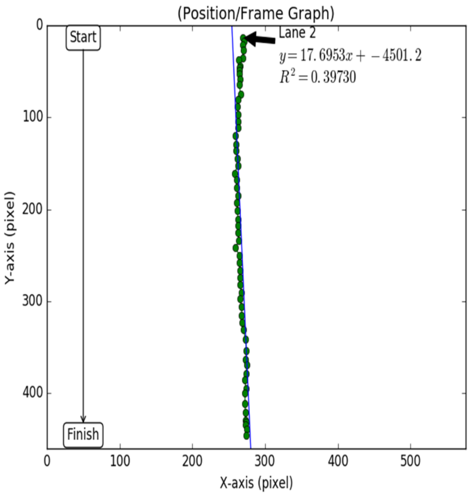
(2)

(3)

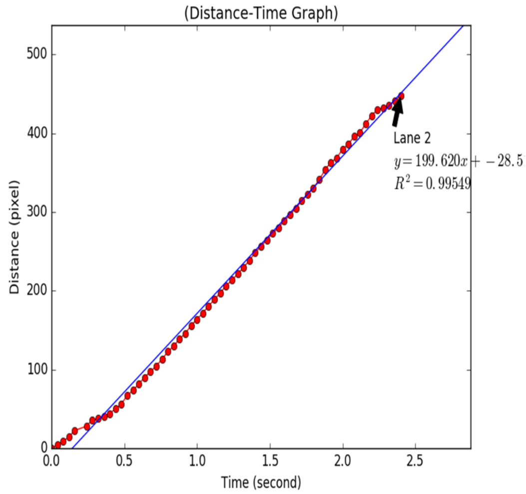
(4)

Fig. 6. The first frame in experiment videos 1 to 4 in Table 2

Next, the vehicle position is detected using the Background Subtraction and tracked using Mixture of Gaussian (MoG). Figure 7a and 7b show a graph of vehicle position and a cummulative distance graph over time from experiment video 2. The position graph consists of the x and y axis of the position coordinates. While the cummulative distance contains the length of distance which increases over time with constant speed.




(a)




(b)

Fig. 7. (a) Graph of vehicle position and (b) Graph of cummulative distance from experiment video 2

Based on the cummulative distance curve (see Fig 7b), we can calculate a trendline equation by using the first-order curve fitting. The average speed of a vehicle is obtained by derivation of the trendline equation over time. After obtaining the average speed, we calculate the relative error by comparing to the actual speed: 0.41 meters/sec and compute the accuracy in each video. Table 3 contains information on the average speed and its relative error and accuracy for each video. The average accuracy of our method is about 96%.

Table 3. Relative error and speed accuracy from vehicle position detection

| Video | Average Speed (meter/sec) | *Relative Error* (%) | Accuracy (%) |
|---|---|---|---|
| Video 1 | 0,422 | 1,89 | 98,11 |
| Video 2 | 0,429 | 3,46 | 96,54 |
| Video 3 | 0,392 | 5,47 | 94,53 |
| Video 4 | 0,434 | 4,60 | 95,40 |
| Average | 0,41 | 3,86 | 96,14 |

In Table 4, we compare the vehicle position detection based on adaptive background mixture model [12] to the ground truth position. The ground truth is done by changing the video file into a sequence of frames and recording the vehicle position from the object midpoint in each frame manually. Table 4 contains the average, maximum, and minimum of the position difference between the detection results and the ground truth. Based on Table 4, it can be seen that the vehicle position in video 3 is the most accurate because its average Euclidean distance is the smallest. Consequently, the camera angle and height that gives the most accurate results is in video 3 which has camera angle of 40 ° and camera height of 32.7 cm from the road.

Table 4. Euclidean Distance Analysis between Detecting Background Subtraction and Truth Ground

| Video | Average *Euclidean Distance* (*pixels*) | Maximal *Euclidean Distance* (*pixels*) | Minimal *Euclidean Distance* (*pixels*) |
|---|---|---|---|
| Video 1 | 13,84 | 18,20 | 4,53 |
| Video 2 | 12,46 | 27,66 | 2,50 |
| Video 3 | 10,82 | 23,24 | 4,27 |
| Video 4 | 11,14 | 21,03 | 4,92 |
| Average | 12,07 | 22,53 | 4,06 |

In the previous research [4], vehicle tracking was performed by using multilayer background subtraction [14] and Eigen background subtraction [15] methods. which the results were compared with speed data on the speedometer. The accuracy of vehicle speed with the multilayer background subtraction is 88.9% and 84.3% for the Eigen background subtraction method. The accuracy of the vehicle speed in our current research is 96.14% which is the average of speed accuracy in 4 videos.

### *4.3. The Results of Projection on Real Traffic Video*

To evaluate our method to real traffic video, we used Kopo toll road video [16]. Videos are cut before being analyzed, so that there is only one vehicle in one lane. We analyze qualitatively toll road as show in Fig 8, where there was one black car drove in the middle lane. In general, the vehicle can be detected properly by our methods, but there are obstacles, that is: the detection is influenced by the vehicle shadow, which is considered as part of the vehicle object. It causes the wrong detected position of the vehicle. In here, we cannot measure the speed accuracy because there is no information about the vehicle speed by using speedometer.

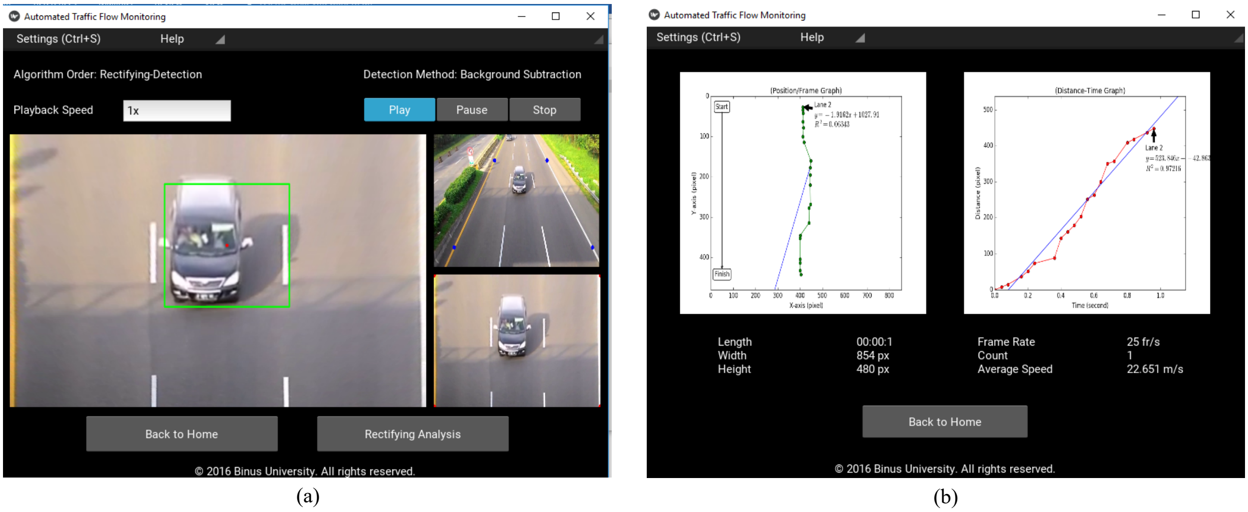



Fig. 8. (a) Vehicle object detection on Kopo toll road video (b) Detection results of vehicle position and speed

## 5. Conclusions

Our proposed DLT projection method has 7.24% more accurate results for vehicle position detection and speed measurement in controlled experiments compared to our previous research [4]. The average accuracy of the speed using our DLT projection method is 96.14% and the average Euclidean distance error of position is 12,07 pixels. Finally, the position graph of the vehicle based on our approach can show that the detection of vehicle movements is in accordance with the actual movement.

The experiments in real dataset show that the research can be continued to solve the fact that the detection is influenced by the vehicle shadow. It can be tried alternative vehicle object detection methods, which are more reliable so that vehicle position detections are more accurate. Another possibility of research is multi vehicle detection problems. By using multi vehicle detection, the application can be applied on real traffic to determine the speed of each vehicle with the number of vehicles more than one vehicle per lane road

## Acknowledgements

This work is supported by Directorate General of Research and Development Strengthening, Indonesian Ministry of Research, Technology, and Higher Education, as a part of Penelitian Terapan Unggulan Perguruan Tinggi (PUPT) Research Grant to Bina Nusantara University with contract number: 12/AKM/PNT/2019 and contract date: 27 March 2019.

## References

1. Gunawan FE, Suharjito , Gunawan AAS. Simulation Model of Bus Rapid Transit. EPJ Web of Conferences. 2014; 68(00021).
2. Suyuti R. Implementasi “Intelligent Transportation System (ITS)” Untuk Mengatasi Kemacetan Lalu Lintas di DKI Jakarta. Konstruksia. 2012;: p. 14-15.
3. Ibrahim O, ElGendy H, Ahmed M. ElShafee. Speed Detection Camera System using Image Processing. International Journal of Computer and Electrical Engineering. 2011 December ; 3(6): p. 1-8.
4. Elysia , Gunawan FE. Vehicle Tracking using Computer Vision Technique for Car-Following Model. ; 2015.
5. Burger W, Burge MJ. Principles of Digital Image Processing: Core Algorithms: Springer; 2009.
6. Wang G, Li J, Zhang P, Zhang X, Song H. Pedestrian Speed Estimation Based on Direct Linear. In International Conference on Audio, Language and Image Processing; 2014; Shanghai. p. 1-5.

7. Hartley R, Zisserman A. Multiple View Geometryin Computer Vision New York: Cambridge University Press ; 2003.
8. Kim H. Vehicle Detection and Speed Estimation for Automated. Tehnicki Vjesnik. 2019; 26(1): p. 87-94.
9. Drews P, Williams G, Goldfain B, Theodorou EA, Rehg JM. Vision-Based High-Speed Driving With a Deep Dynamic Observer. IEEE Robotics and Automation Letters. 2019; 4(2).
10. Abdel-Aziz YI, Karara DHM. Direct Linear Transformation from Comparator. Coordinates into Object Space Coordinates in. Close-Range Photogrammetry. Photogrammetric Engineering & Remote Sensing. 2015 February; 81(1): p. 103–107.
11. KaewTraKulPong P, Bowden R. An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. In Remagnino P, Jones GA, Paragios N, Regazzoni CS, editors. Video-Based Surveillance Systems.: Kluwer Academic; 2002. p. 135-144.
12. Zivkovic Z, Heijden Fvd. Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction. Pattern Recognition Letters. 2006; 27(7): p. 773-780.
13. Gandhi SA, Kulkarni CV. MSE Vs SSIM. International Journal of Scientific & Engineering Research. 2013 July 7; 4(7): p. 930.
14. Yao J, Odobez JM. Multi-Layer Background Subtraction Based on Color and Texture. In IEEE Conference on Computer Vision and Pattern Recognition; 2007; Minneapolis.
15. Gritti T. Robust Background Subtraction with Incremental Eigen Models. Technical Note. Philips Research Europe; 2008. Report No.: PR-TN 2008/00531.
16. Firdaus A. Kopo Toll Road Video. [Online].; 2016. Available from: https://youtu.be/OcKtb6reBl0.