



MULTI VEHICLE SPEED DETECTION USING EUCLIDEAN DISTANCE BASED ON VIDEO PROCESSING

Budi Setiyono ¹⁾, Dwi Ratna Sulistyaningrum ²⁾, Soetrisno ³⁾,
Danang Wahyu Wicaksono ⁴⁾

Department of Mathematics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
masbudisetiyono@gmail.com, dwratna@gmail.com, soetrisno@gmail.com, <https://www.its.ac.id>

Paper history:

Received 15 November 2018

Received in revised form 04 August 2019

Accepted 11 November 2019

Available online 31 December 2019

Keywords:

Smart City;

Vehicle speed estimation;

Gaussian Mixture Model;

Region Of Interest;

Euclidean Distance;

Abstract: One component of smart city is smart transportation, known as Intelligent Transportation Systems (ITS). In this study, we discuss the estimation of moving vehicle speed based on video processing using the Euclidean Distance method. In this study, we examine the effect of camera angles on the video acquisition to speed estimation accuracy. In addition, Region of Interest (ROI) will be designed into three parts to determine which area is the most appropriate to be chosen, so that the estimated vehicle speed will be better. These approaches have never been studied by previous researchers. The separation between the background and foreground is conducted using Gaussian Mixture Models method. By comparing the displacement distance and the number of frames per second (fps), we obtain speed estimate for each vehicle. According to the experimental results, our system can estimate the speed of the vehicle with an accuracy of 99.38%.

Copyright © Research Institute for Intelligent Computer Systems, 2019.
All rights reserved.

1. INTRODUCTION

Smart city is defined as a concept of developing and managing cities by utilizing Information and Communication Technology (ICT) to connect, monitor, and control various resources within the city. These resources include administration, education, health, public safety, real estate, transportation, and other utilities to be smarter. Besides that, the interconnection between these various resources becomes more efficient and effective in maximizing services to its citizens and supporting sustainable development [1].

One component of smart city is smart transportation, better known as Intelligent Transportation Systems (ITS) [2], [3]. With ITS, it is expected that traffic engineering can be carried out automatically, such as automation of traffic lights to be adaptive, automation in finding alternative routes to parse congestion, and automation in detection of speed limit violations. The basic requirement for such automation process is knowing the speed of each vehicle that crosses the highway. Image and video processing can be used to detect speed on vehicles moving on the highway. Videos can be

obtained from surveillance cameras that have been installed on many roads.

In previous studies, analysis on the angle of shooting and division of Region Of Interest (ROI) areas has never been studied, even though it can affect accuracy in vehicle speed detection. Therefore, in this study an analysis and experiment will be conducted regarding the effect of the camera angle at the time of data acquisition. In addition, we divide the ROI region and we observe the influence on the accuracy of the results, in order to obtain the proper vehicle speed estimation.

The uniqueness of this research is trying to analyze the influence of ROI positions with different points of view. It turns out that the position of ROI with different shooting angles (although the angle difference is small), also greatly affects the accuracy of the results, so the contribution of this research is how to set the camera angle, if it will be used for vehicle speed detection on certain roads, for example in Indonesia. If it is used on a toll road, the speed of the vehicle is high, we propose to use a 45 degree angle. If it is used on a road where the average speed is low (for example on a village road / congested road), the camera angle is 45 degrees.

In the rest of the paper, the structure is as follows. Section 2 presents related work. Section 3 describes the proposed method and explains the steps to solve the problem of speed estimation for multi vehicles. Furthermore, in the fourth section, the trial and analysis of the results obtained are explained. The final section of this paper contains conclusions and research that will be carried out in the future.

2. RELATED WORK

There have been many previous studies related to vehicle speed detection and estimation based on digital image processing. The research conducted by Tarun Kumar and Dharmendra Singh Kushwaha used the ROI of the area around the vehicle number plate [4]. Therefore, a red reversal operation is used so that the pixel intensity in ROI increases. The focus of that research is on the detection and removal of fake backgrounds. Other researchers, using an optical flow and motion vector approach for tracking on vehicles[5]–[10] In that study, the method for calculating vehicle speed estimates is a difference of three frames. Accuracy of speed estimation is increased using optical flow values and managed to obtain the highest accuracy, but the optical flow method used for speed detection is very sensitive to noise and requires a long computational time. In that study, experiments were also conducted by changing the intensity of lighting.

The velocity estimation study on other vehicles was also conducted by Budi Setiyono et al. Using the Gaussian Mixture Model (GMM) to separate the background and foreground on a single vehicle [11]. GMM is also widely used in other studies, especially for separating background and foreground because the results are quite accurate [12]–[16]. Other than GMM, Arash Gholami R., et al were using Combination of Saturation and Value (CVS) for foreground extraction and for detecting vehicles. Accuracy obtained in that study is +/- 7 km/h [17]. To improve accuracy in the vehicle detection process, shadow removal was used in that study. Research on shadow removal has been carried out by other researchers [18]–[24]. For vehicle speed calculation, it is based on the pixel transfer distance between frames using Euclidean Distance. In that research, the software was able to obtain vehicle speed estimates using the Euclidean distance method with a maximum success rate of 98.12% [25].

Furthermore, Hardy Santosa S and Agus Harjoko in their research used the Euclidean distance method to estimate vehicle speed and stated that the results of speed estimation from processing video data are still better than using speed gun with the highest accuracy rate of 90.5%. Other studies, Fuzzy

approaches have also been used for speed estimation [26], [27].

However, most of these studies focus to estimate the speed of a single vehicle. In addition, the influence of the camera's slope in video acquisition and the selection of the Region of Interest (ROI) has not yet been studied further. By choosing the camera tilt angle and determining the right ROI area, the accuracy of speed estimation can be increased, even for multi vehicles. This is because both factors are very influential in speed estimation.

Therefore, in this research, we focus on determining the relationship between angle of shooting on the camera and accuracy of the results. In addition, ROI will be divided into three parts, to obtain the best ROI for speed detection.

3. PROPOSED METHOD

We propose a method for the estimation of multi-vehicle speed. The steps in the method are shown in Fig. 1. Broadly speaking, the steps taken to solve the problem are divided into four stages, namely initialization, preprocessing, object detection and estimated vehicle speed.

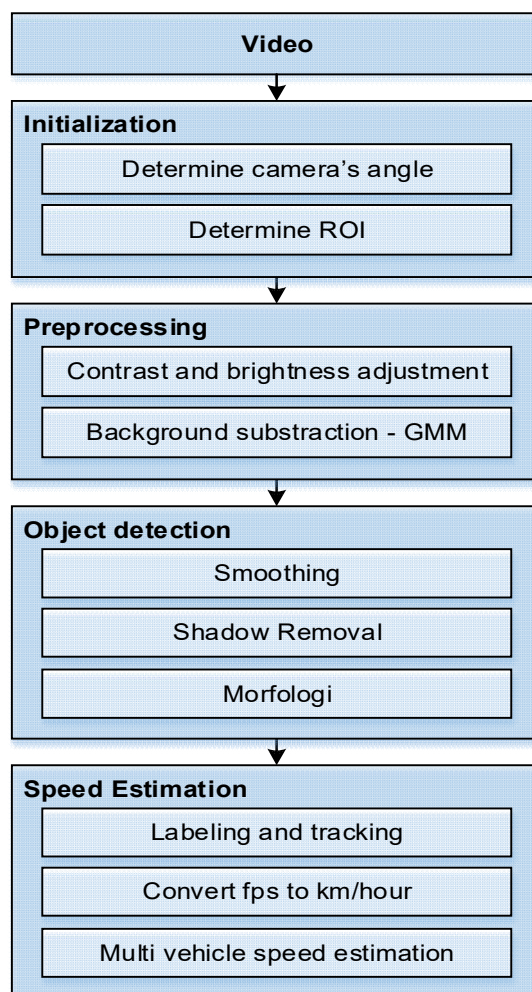


Figure 1 – Proposed method

Initialization stage consists of the determination of the camera angle and selection of ROI area, which is an important part of this study. Determining the right angle of the camera will increase accuracy in estimating multi-vehicle speeds.

Preprocessing includes contrast adjustments to facilitate the background and foreground separation. The next step is to detect and separate the vehicle objects, which include smoothing, shadow removal and morphology. In the final stage, the speed estimation in multi vehicles includes tracking, centroid distance calculation between frames with Euclidean Distance, and conversion to speed in km/h using frame rate (fps) considerations. The result is speed estimation for each vehicle that has been detected successfully. The steps in Figure 1 will be explained in more detail.

3.1 CAMERA'S ANGLE SETUP

Before the speed estimation process is carried out, at the time of image acquisition, first a geometric parameter is needed to calibrate the vehicle speed from pixel / second to km / hr. The representation of geometric parameters can be seen in Fig. 2.

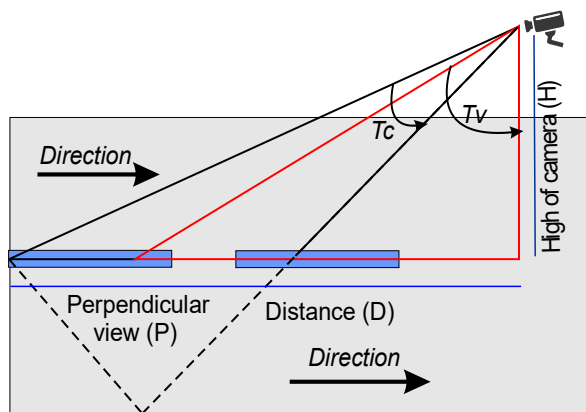


Figure 2 – Geometric parameters Camera angles for video acquisition

In Fig. 2, it appears that $\theta_3 = T_c$ which is the angle of Field Of View (FOV) coverage of the camera defined in Equations 1 and θ_1 is the camera's slope angle and includes the angle of coverage of the FOV, so based on the Image, θ_1 is obtained by Equation 2.

$$\theta_3 = T_c = 2 \arctan\left(\frac{v}{2f}\right), \quad (1)$$

$$\theta_1 = T_v + \frac{T_c}{2}, \quad (2)$$

where f is the camera's focal length, v is the vertical dimension of the 35mm image format, and T_v is the camera's tilt angle during video acquisition. The distance D is obtained from Equation 3

$$D = H \tan(\theta_1), \quad (3)$$

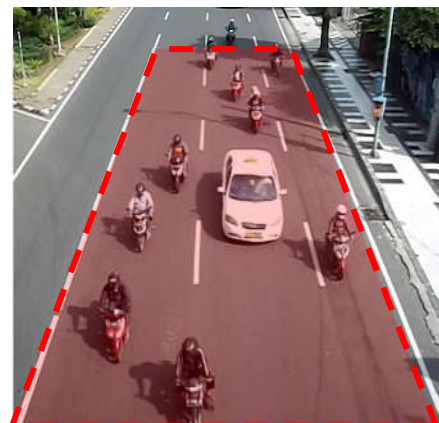
where H is the camera height from the road surface. Equations (5-7) will be used to calculate the perpendicular view P , shown in Equation 4.

$$P = 2 \tan\left(\frac{\theta_3}{2}\right) \sqrt{H^2 + D^2}. \quad (4)$$

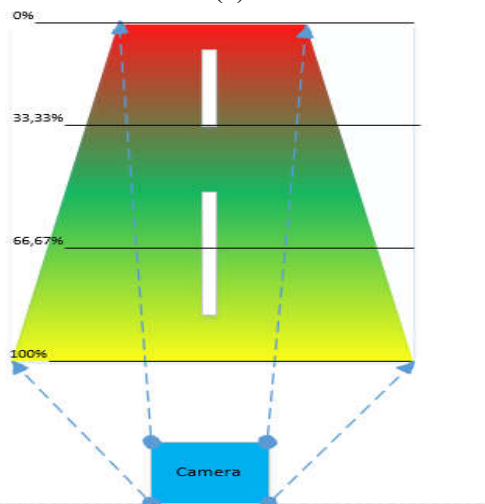
Once geometric parameters are obtained as input, namely T_c , T_v , and H , then the system will calculate the values of D and P automatically. Then the vehicle speed is calculated using the Euclidean distance which is explained in Section 2.7. where the speed will be calculated through the distance between centroid points then divided by the time between 2 frames and the result is multiplied by the calibration factor.

3.2 DETERMINE REGION OF INTEREST

The problem of choosing ROI for speed recording areas is one of the important stages, because it will affect the estimated speed produced.



(a)



(b)

Figure 3 – Visual Camera Perspective of Vehicle Speed (a) Region of Interest in Video (b) Speed Perspective in 3 areas

In this study, the perspective of the camera used is from the top with a static camera position, so that the resulting image will have a speed perspective like Fig. 3(a). This perspective will also be used to estimate speed based on centroid points between frames. We divided the ROI area into 3 speed profile areas, namely 0-33%, 33-66% and 66-100%, as seen in Fig. 3(b).

By dividing the area into 3 velocity profiles, we categorize the area into 6 possible selection of ROI regions, shown in Table 1.

Table 1. Speed Profile on various types of ROI

ROI Category area (%)	Speed Profile		
	Low	Medium	High
0 – 33,33	v		
33,33 – 66,67		v	
66,67 – 100			v
0 – 66,67	v	v	
33,33 – 100		v	v
0 – 100	v	v	v

3.3 PREPROCESSING

Preprocessing in this study is contrast and brightness enhancement. Broadly speaking, the process can be explained as follows: Suppose the pixel $f(i, j)$ is the intensity of the original pixel at the coordinates (i, j) and $g(i, j)$ is the pixel intensity of the result. If $\alpha > 0$ is a parameter of gain (contrast) and β is a parameter of bias (brightness), then contrast and brightness enhancement are defined in Equation 5.

$$g(i, j) = \alpha \cdot f(i, j) + \beta. \tag{5}$$

In the background subtraction process, to separate the background with the foreground, the process is as follows: for every frame that has been extracted, look for the background model that will be used to generate the foreground image based on the background model. The process of finding the background model and generating foreground is computed using the Gaussian Mixture Model (GMM). GMM is used in this process because this method is resistant to changes in the characteristics of the background model which may change at any time. This can happen because each pixel in each frame change will be evaluated through the weight parameter update, standard deviation, and mean.

Each pixel will be grouped by distribution, which is considered the most effective as a background model. The greater the standard deviation value, the stronger the smoothing that occurs in the image. Each pixel has its own model. The processed data is the intensity of the pixels obtained from the input frame. Every frame is extracted and the model for each pixel will be updated.

In the system to be built, there are several parameters that have been defined previously. This parameter is used for background subtraction processes with GMM [5]. These parameters include α (learning rate) with a value of 0.01, the number of Gaussian components, namely 3, T (threshold) with a value of 0.4. In addition, there is also an initialization of some GMM parameters, among others: ω_k which is the weight of each pixel in the k-Gaussian with a value of 1/3 where 3 is the sum of the Gaussian distribution; μ_k is the mean of each pixel in the k-Gaussian where each pixel of each Gaussian has a random value between 0 and 255; σ_k is the standard deviation of each pixel in the k-Gaussian.

3.4 OBJECT DETECTION

To obtain good results in object detection, in this study, we used three processes namely smoothing, shadow removal and morphology. This process is conducted so that each object captured by the camera can be easily separated from each other to be given an ID on each object.

Smoothing is a process to refine the image so that noise in the foreground image results in a reduced subtraction background process. The background subtraction process used in this study can simultaneously be used to detect shadows. However, to maximize the detection process, the identified shadows are not immediately removed when detected, but need a smoothing process so that the suspected part of the shadow becomes more accurate. Therefore, the shadow removal process in this system is carried out after the smoothing process.

As explained earlier, the image results from the smoothing process have 3 levels of gray, namely 0, 127, and 255. Shadow removal is the process by which the values are mapped to binary numbers namely 0 and 1 as described in Equation 6.

$$I_{bin}(x, y) = \begin{cases} 0, & I_{gray}(x,y)=0 \text{ or } I_{gray}(x,y)=127 \\ 1, & I_{gray}(x,y)=255 \end{cases}. \tag{6}$$

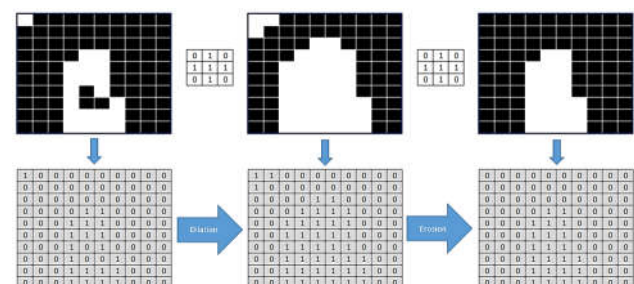


Figure 4 – Morphology Closing Process

Binary image shadow removal results sometimes produce objects that have holes in the center of the

object. Thus, it is necessary to fill the holes using morphology, namely closing operations with the aim of better object detection results. By performing the closing operation on the binary image from the shadow removal results, the small holes in the object will be closed. The results of the closing process on pixels can be seen in Fig. 4.

3.5 LABELLING AND TRACKING

The object detection process in this study uses the concept of connected components to detect contours on binary images. Contours detected on objects are based on the outer border or the outermost boundary with the border following method. The steps for object contour detection is presented in Figure 5.

1. The scanning process runs with a raster scan, which is from top to bottom (row), left to right (column).
2. Look for pixels $f(i, j)$ as pixel outer border candidates that meet the following conditions:

	j-1	j
i	0	1

and if the pixel is the outer most border, it satisfies the following conditions:

 - a. Pixel $f(i, 1), f(i, 2), \dots, f(i, j - 1)$ is pixel 0, or
 - b. pixel $f(i, h)$ parallel to the border point, and pixels $f(i, h + 1)$ are background. Label it as a border sign.
3. If the end border has been found, namely:

	j-1	j
i	1	0

Label it as a border sign.
4. Run raster scanning until all pixels are passed.

Figure 5 – The Steps in Object Contour Detection

The object of the vehicle will be labeled as an ID based on the order in which the vehicle appears on ROI. When the object is in ROI, the vehicle speed between frames will always be recorded. When the object leaves the ROI, the average speed will be calculated and then stored in memory. The stored data is displayed in the output data in the form of a table of results from the average estimated speed of each vehicle. To ensure that objects are counted and stored more than once, the tracking object process between frames needs to be conducted. Tracking is a process that aims to find the same object in the previous frame. Two objects are said to be the same

if they meet Inequality 7 and Inequality 8.

$$L_{objek1} \leq \frac{L_{objek2} + R_{objek2}}{2} \leq R_{objek1} \quad (7)$$

$$T_{objek2} \leq T_{objek1} \leq B_{objek2}, \quad (8)$$

where object1 is an object that is being processed, objects are objects in the previous frame. L is the left boundary of the object, R is the right boundary of the object, T is the upper limit of the object, and B is the lower limit of the object. If the two objects identified are the same, then the ID in the previous object will move to the object being processed and will calculate the speed based on the distance of the centroid point of the object between the two frames. The process will continue until the object leaves the ROI.

3.6 EUCLIDEAN DISTANCE FOR SPEED ESTIMATION

To compute the distance traveled in pixels, suppose the coordinates of an object are described in Equation 9:

$$c_t(a, b) \text{ and } c_{t+1}(c, d), \quad (9)$$

where c_t and c_{t+1} is the position of the centroid at frames t and $t + 1$ for one object, namely with coordinates (a, b) and coordinates (c, d) . The distance difference is measured by the Euclidean Distance method [4], namely:

$$d = \sqrt{(a - c)^2 + (b - d)^2}, \quad (10)$$

whereas, for speed, we obtained

$$V = k \frac{d}{t}, \quad (11)$$

where k is the calibration coefficient and t is the time between 2 consecutive frames, where

$$k = \frac{\text{perpendicular_view}}{\text{image_height}} \quad (12)$$

$$t = \frac{1}{fps}. \quad (13)$$

The illustration of speed estimation using Euclidean distance is given in the following explanation. Suppose there is an image from the object detection process at frames t and $t + 1$ Figure 7 is a more detailed illustration of calculating the speed of objects from the distance and time between the frames given in Fig. 6.

In Fig. 7, two centroid points are obtained on one object that its speed will be estimated, for example $c_t = (3.5, 5)$ and $c_{t+1} = (5.5, 7)$. Then the distance will be calculated using Euclidean distance using Equation 10 so that we obtained

$$d = \sqrt{(5.5 - 3.5)^2 + (7 - 5)^2} \approx 2,83 \text{ pixel.}$$

Speed v obtained from Equation (12), is in meters per second or m/s . For conversions from meters per second to kilometers per hour or km/h , the result v multiplied by 3.6 is obtained from Equation 16.

$$1 \text{ m} = \frac{1}{1000} \text{ km} \tag{14}$$

$$1 \text{ second} = \frac{1}{3600} \text{ hour.} \tag{15}$$

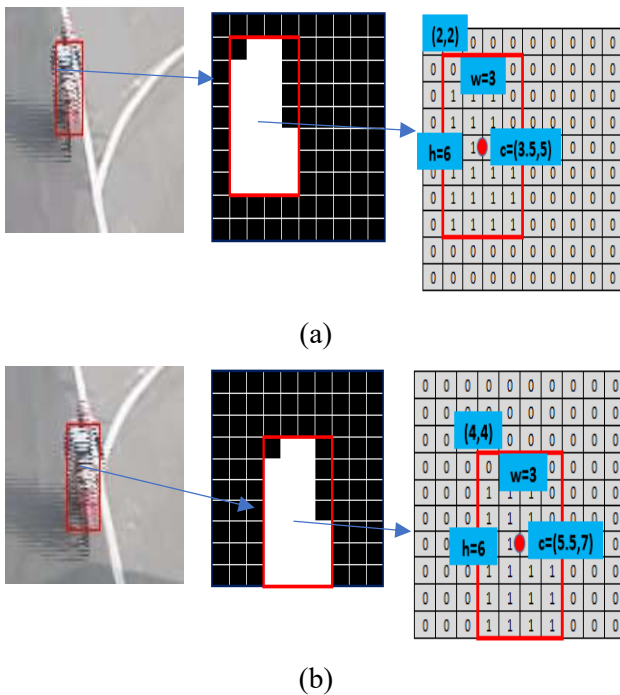


Figure 6 – Illustration of Estimated Speed (a) Image at frame t (b) Image at $t + 1$ frame

From the Equation 14 dan 15, it is obtained

$$1 \frac{m}{s} = \left(\frac{1}{1000} \right) \frac{km}{\left(\frac{1}{3600} \right) jam} = 3,6 \frac{km}{jam} \tag{16}$$

So the velocity formula V in kilometers per hour or km / h from Equations 11 and 16 is

$$V = 3,6 k \frac{d}{t} \tag{17}$$

Equation 17 is the velocity formula used in this study where the speed is in standard units, namely kilometers per hour.

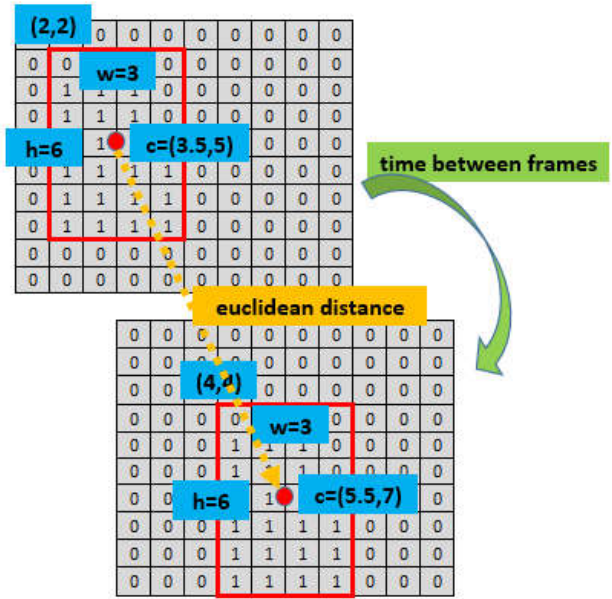


Figure 7 – Object speed of frame t and frame $t + 1$ from Euclidean distance and time between frames

3.7 ALGORITHM DESIGN

Algorithms for detecting multi-vehicle speeds in parallel are determined by giving IDs to each detected vehicle object. The process is explained by the algorithm in the form of the pseudocode as shown in Figure 8. The first time the listObj is initialized with 0, then the vehicle will be detected as a new object and labeled ID, entered in the listObj, but the speed estimation process has not been computed. In the next frame, if the vehicle detected is the same as the vehicle in the previous frame, which is already on the listObj, the object updating process will include an update ID on the new vehicle object, update the bounding box location, calculate the distance between centroid points on the current frame and the previous frame, and estimated speed. Then stored on listSpeed for the vehicle object. If the vehicle is not the same or does not match the vehicle objects that are on the list, it can be said that the detected vehicle is a new vehicle, and will be given an ID and entered into the list. This process is done as many objects detected in ROI. When the vehicle exits the ROI, the average speed will be calculated based on the instantaneous speed that has been stored in the listSpeed, and the results are displayed in the table of speed estimation results according to the vehicle ID.

Algorithm : Vehicle Speed Estimation

```

1. initialize listObj and ID
2. listObj ← {}; ID ← 0
5. For Each obj in Objects do
6.   If listObj is empty, then
7.     let boundingBox is rectangle
       coordinate of object and
       centroid calculated from it.
8.     listSpeed ← {}
9.     obj ← ID
10.    obj ← boundingBox
11.    obj ← centroid
12.    obj ← listSpeed
13.    listObj ← obj
14.    If listObj is not empty,
then
15.      found ← false
16.      For Each lobj in listObj do
17.        if obj equals lobj, then
18.          let cobj, clobj is centroid
           of obj and lobj
           respectively, v is
           velocity, d is distance,
           k is calibration factor,
           and t is time between 2
           frames.
19.          d ← Euclidean_distance
             (cobj, clobj)
20.          v ← 3.6 k  $\frac{d}{t}$ 
21.          listSpeed ← v
22.          lobj ← boundingBox
23.          lobj ← centroid
24.          lobj ← listSpeed
25.          found ← true
26.          If lobj is out of ROI,
then
           let va is average
           velocity and res is
           result repository
27.          va ←  $\frac{\sum v \text{ listSpeed } v}{\text{size}(\text{listSpeed})}$ 
28.          res ← va
29.          End If
30.          End If
31.          End For
32.          End If
33.          End For
34.          If found is false, then
35.            ID ← ID + 1
36.            listSpeed ← {}
37.            obj ← ID
38.            obj ← boundingBox
39.            obj ← centroid
40.            obj ← listSpeed
41.            listObj ← obj
42.          End If
43.        End if
44.      End For

```

Figure 8 – Multi vehicle detection algorithm and speed estimation

Figure 9 is the result of vehicle detection and the provision of bounding boxes in multi vehicles.

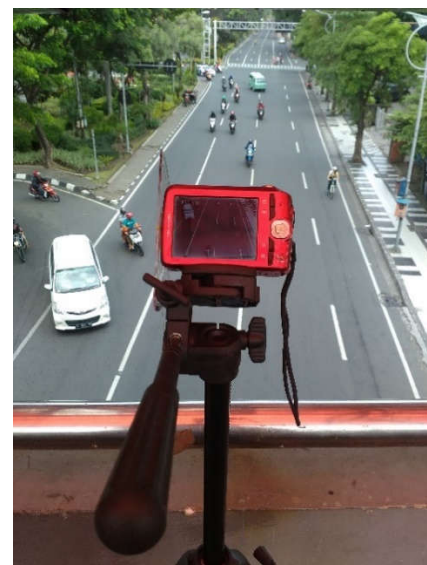


Figure 9 – Bounding Box dan ID on multi Vehicle detection

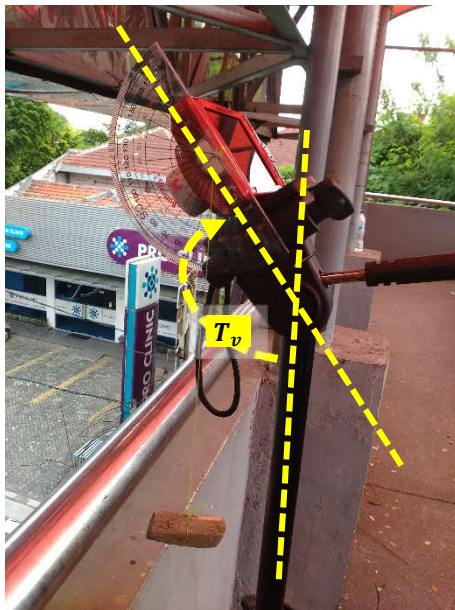
4. EXPERIMENTAL RESULT

For video data acquisition, the camera's tilt angle is designed to be modified as shown in Fig. 10. The trial was conducted in several processes, namely (i) trials related to the Region of Interest (ROI) (ii) trial preprocessing (iii) test object detection and (iv) trial speed estimation.

The data used in the speed estimation trial is a video of vehicles on the highway of known speed and acquisitions are conducted with different geometric parameters. Acquisition of using the Kodak Easyshare C1530 camera with 32 mm focal length and 24 mm vertical dimension so that based on Equation 5, obtained $T_C=41,1^\circ$. The scenario in this study uses a vehicle with a known speed to be the actual speed using speed gun as a reference (ground truth). Simultaneously video recording is carried out with predetermined parameters. Next, a manual comparison of the system speed and speed of the speed gun will be done manually.



(a)



(b)

Figure 10 – Camera Position during Video Acquisition. (a) Rear View. (b) Side View

The vehicle speed used for the experiment is 15, 20, 30, 40, 50, and 60 km/hour, and is relatively constant starting from the vehicle entering the record area until the vehicle exits the record area. While video acquisition is also conducted with several different camera tilt angles of 60, 50, and 45 degrees with the video frame rate is 30 fps, $T_C = 41,1^\circ$ and 7.6 m camera height from the ground. Trial data are presented in Table 2.

Table 2. Data Experiment

No	Video Name	T_v	Speed (km/h)
1	dtc-15-60d.avi	60	15
2	dtc-20-60d.avi	60	20
3	dtc-30-60d.avi	60	30
4	dtc-40-60d.avi	60	40
5	dtc-50-60d.avi	60	50
6	dtc-60-60d.avi	60	60
7	dtc-15-50d.avi	50	15
8	dtc-20-50d.avi	50	20
9	dtc-30-50d.avi	50	30
10	dtc-40-50d.avi	50	40
11	dtc-50-50d.avi	50	50
12	dtc-60-50d.avi	50	60
13	dtc-15-45d.avi	45	15
14	dtc-20-45d.avi	45	20
15	dtc-30-45d.avi	45	30
16	dtc-40-45d.avi	45	40
17	dtc-50-45d.avi	45	50
18	dtc-60-45d.avi	45	60

While the selection of the best ROI area is computed by comparing the different ROI regions according to the camera tilt angle in the trial video. This is conducted with the aim of getting the best ROI area in order to produce a more accurate speed estimate.

Table 3. Testing of ROI Type

Tye of ROI Area	Accuracy(%) - Camera's Angle			Avg
	45°	50°	60°	
0–33,33	37,005	34,8275	56,72	42,851
33,33–66,67	59,945	63,0875	85,86	69,631
66,67–100	83,885	90,5225	55,765	76,724
0–66,67	56,6725	53,555	75,245	61,824
33,33–100	90,465	95,87	67,985	84,773
0–100	72,71	70,3775	96,3	79,796

In the form of a graph, Table 3 above is presented in Figure 11. The graph clearly shows that the type of ROI profile for regions 33.33 - 100 has the highest accuracy.

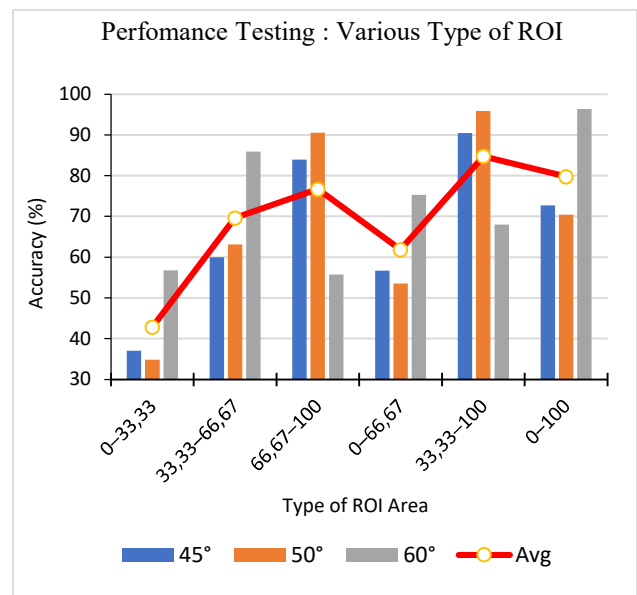


Figure 11 – ROI Performance type testing

From the picture, it appears that the area between 33.33 - 100 percents of ROI has the highest average accuracy.

The results of the trial related to preprocessing for background processing are shown in Fig. 12. Fig. 12(a) is one of the acquisition frames, while image 12(b) of background subtraction results uses GMM from Fig. 12(a). The smoothing process used is the median filter. Fig. 13 (a) shows the smoothing results performed on the image resulting from the background subtraction process.

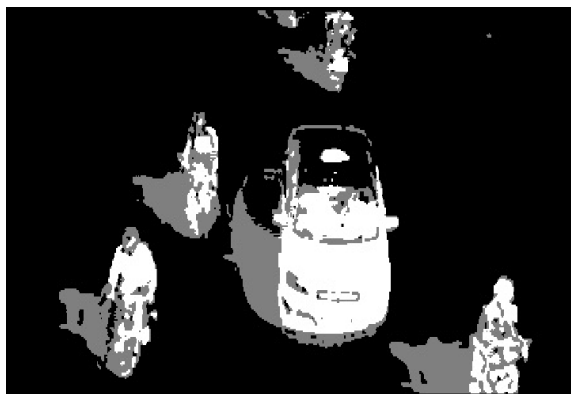


(a)



(b)

Figure 12 – Result of Background Subtraction
(a) An acquisition frame (b) background subtraction result



(a)



(b)



(c)

Figure 13 – (a) Smoothing Results (b) Shadow Removal Results (c) Morphology Results

After smoothing, a shadow removal process will be performed on each video frame. The results of this process are shown in Fig. 13 (b). The last stage of the preprocessing process is morphology. Fig. 13 (c) shows morphological results after shadow removal. In the picture it can be seen that the hollow area will be closed.

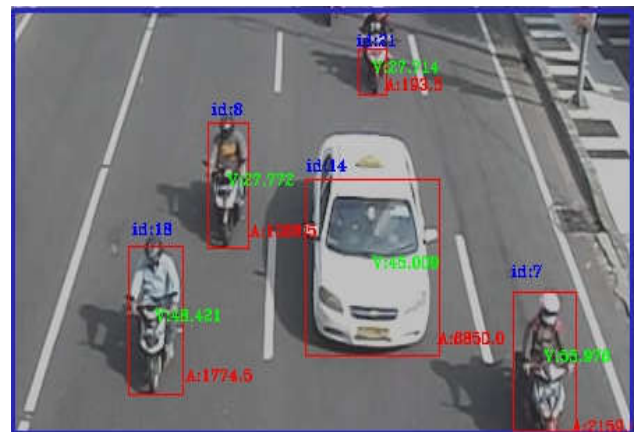


Figure 14 – Bounding Box and Labelling

Furthermore, every detected vehicle object will be labeled and tracked. Tracking is used to ensure that vehicles entering and exiting the ROI area are the same vehicle objects.

Table 4. Experiment for various Camera Angles

Speed (km/h)	Accuracy(%) - Camera's angle		
	45°	50°	60°
15	97,7	94,17	91,1
20	99,38	94,23	92,88
30	93,12	96	93,72
40	92,305	95,87	96,3
50	92,384	96,892	97,264
60	87,01	89,68	89,145
Avg	93,64983	94,47367	93,4015

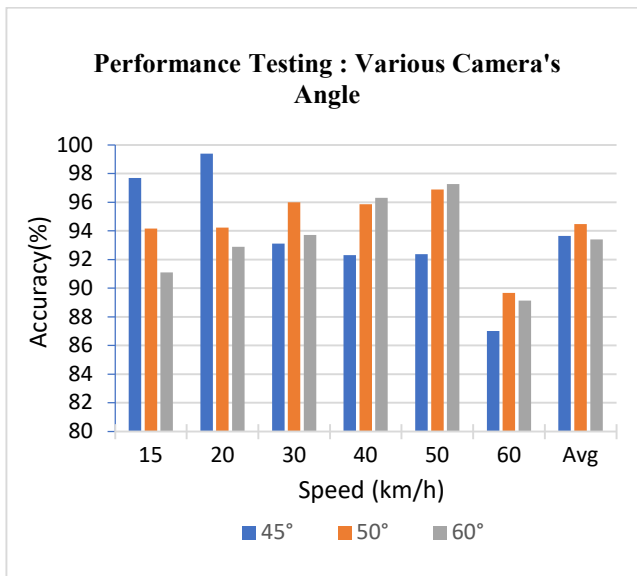


Figure 15 – Trial results use the best parameters

Thus, each vehicle object can be assigned an ID. The labeling and bounding box results will be visualized using the original video frames so that it will look like Figure 14.

With the steps that have been discussed above, the final step is to test all test data videos and use the best parameters, which include the camera angle, ROI area, preprocessing and speed estimation. The results of the experiment are shown in Table 4 and presented in graphical form in Figure 15. The highest accuracy obtained from this trial is 99.38% obtained at the slope of the camera angle of 45°.

Some of the things related to the results of our experimental analysis are as follows: (i). If there are two object vehicles that coincide and have the same speed, then the method will assume the two vehicles become one vehicle, so it only calculates the speed of one vehicle. This is due to the shadow of one object concerning another object. Therefore, a more reliable shadow removal method is needed. (ii). In this study, the vehicle object detection method can only work in bright weather conditions and high lighting, so that for night conditions, a lot of noises and lighting are less needed by other methods. Rain is an example of noise.

4. CONCLUSION AND FUTURE WORK

This study discussed the estimation of vehicle speed at different camera tilt angles with the lowest accuracy of 87.01% and the highest was 99.38% and resulted in a consistent level of accuracy for

estimating consistent multi-object vehicle speeds.

From the results obtained, by classifying the test data into three types, namely low, medium, and high speed that the camera angle can affect the accuracy of the results of speed estimation. The 45 degree camera tilt angle provides the highest accuracy for low speed, a 50 degree angle provides the highest accuracy for medium speed, and a 50 degree angle provides the highest accuracy for high speed so that this can be taken into consideration by the parties concerned.

The proposed object detection system has some disadvantages in certain conditions, for example vehicle objects that have shadows, environmental conditions that have a lot of noises, such as rain and at night. Thus, in the next study, researchers will develop algorithms and methods to deal with these problems. Furthermore, another study that will be conducted is to make the system become adaptive to changing environmental conditions, because of the effects of lighting, noise and different objects.

5. ACKNOWLEDGMENT

This work was supported by PDUPT with the title of *Sistem Deteksi Kecepatan Kendaraan Berbasis Pengolahan Citra untuk Membantu Meningkatkan Manajemen Transportasi, Kontrak Penelitian Nomor 867/PKS/ITS/2018 Tanggal 1 Februari 2018*, from the Ministry of Research, Technology and Higher Education of the Republic of Indonesia.

6. REFERENCES

- [1] V. Albino, U. Berardi, and R. M. Dangelico, "Smart cities: definition, deminsion, and performance," *J. Urban Technol.*, vol. 22, pp. 3–21, 2015.
- [2] M. Mirboland and K. Smarsly, "A semantic model of intelligent transportation systems," no. 2011, pp. 1–10, 2015.
- [3] M. A. Khan, S. Abbas, Z. Hasan, and A. Fatima, "Intelligent Transportation System (ITS) for Smart-Cities using Mamdani Fuzzy Inference System," no. January, 2018.
- [4] T. Kumar and D. S. Kushwaha, "An Efficient Approach for Detection and Speed Estimation of Moving Vehicles," *Procedia Comput. Sci.*, vol. 89, pp. 726–731, 2016.

- [5] S. Indu, M. Gupta, and P. A. Bhattacharyya, "Vehicle Tracking and Speed Estimation using Optical Flow Method," *Int. J. Eng. Sci. Technol.*, vol. 3, no. 1, pp. 429–434, 2011.
- [6] J. Lan, J. Li, G. Hu, B. Ran, and L. Wang, "Vehicle speed measurement based on gray constraint optical flow algorithm," *Opt. - Int. J. Light Electron Opt.*, vol. 125, no. 1, pp. 289–295, Jan. 2014.
- [7] R. Ke, S. Kim, Z. Li, and Y. Wang, "Motion-vector clustering for traffic speed detection from UAV video," *2015 IEEE 1st Int. Smart Cities Conf. ISC2 2015*, no. October, 2015.
- [8] S. S. S. Ranjit, S. A. Anas, S. K. Subramaniam, K. C. Lim, A. F. I. Fayeez, and A. R. Amirah, "Real-Time Vehicle Speed Detection Algorithm using Motion Vector Technique," *Proc. Int. Adv. Electr. Electron. 2012*, pp. 67–71, 2012.
- [9] P. P. S. S and C. Manisha, "Vehicle Detection and Tracking using the Optical Flow and Background Subtraction."
- [10] D. Woo and A. N. Hong, "An Integrated Solution to Motion Tracking of Moving Vehicles Using Optical Flow," vol. 1, no. 1, 2014.
- [11] B. Setiyono, D.R. Sulistyningrum, Soetrisno, F. Fajriyah, and D. W. Wicaksono, "Vehicle speed detection based on Gaussian mixture model using sequential of images," vol. 890, no. 1, pp. 0–6, 2017.
- [12] T. Huang, H. Peng, and K. Zhang, "Model Selection for Gaussian Mixture Models," *arXiv:1301.3558v1*, pp. 1–27, 2013.
- [13] D. a Reynolds, "Gaussian Mixture Models," *Encycl. Biometric Recognit.*, vol. 31, no. 2, pp. 1047–64, 2008.
- [14] Y. Li, Z. Li, H. Tian, and Y. Wang, "Vehicle Detecting and Shadow Removing Based on Edged Mixture Gaussian Model," *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 800–805, Jan. 2011.
- [15] M. K. Sarker, C. Weihua, and M. K. Song, "Detection and Recognition of Illegally Parked Vehicles Based on an Adaptive Gaussian Mixture Model and a Seed Fill Algorithm," vol. 13, no. 3, pp. 197–204, 2015.
- [16] D. H. H. Santosh, P. Venkatesh, P. Poornesh, L. N. Rao, and N. A. Kumar, "Tracking Multiple Moving Objects Using Gaussian Mixture Model," no. 2, pp. 114–119, 2013.
- [17] A. Gholami Rad, A. Dehghani, and M. Rehan Karim, "Vehicle speed detection in video image sequences using CVS method," *Int. J. Phys. Sci.*, vol. 5, no. 17, pp. 2555–2563, 2010.
- [18] W. Wu, J. Shao, and W. Guo, "Moving-object Detection Based on Shadow Removal and Prospect," *Int. Conf. Artif. Intell. Soft Comput.*, vol. 12, no. 2, pp. 369–374, 2012.
- [19] B. Setiyono, D. R. Sulistyningrum, and H. Al-habib, "Improvement of sub region matching illumination transfer in hybrid shadow removal method for moving vehicle video," *Int. J. Eng. Technol.*, vol. 7, no. 4, pp. 2515–2520, 2018.
- [20] A. Prati, I. Miki, M. M. Trivedi, and R. Cucchiara, "Detecting Moving Shadows :," no. 858, pp. 1–15.
- [21] S. Murali, V. K. Govindan, and S. Kalady, "A Survey on Shadow Removal Techniques for Single Image," *Int. J. Image, Graph. Signal Process.*, vol. 8, no. 12, pp. 38–46, 2016.
- [22] H. Prajapati, S. Longowal, B. Singh, and S. Longowal, "An Approach for Shadow Removal in Moving Object," *Int. J. Comput. Sci. Inf. Secur.*, vol. 14, no. 8, pp. 781–790, 2016.
- [23] Y. W. Shou, C. T. Lin, C. T. Yang, and T. K. Shen, "An efficient and robust moving shadow removal algorithm and its applications in ITS," *EURASIP J. Adv. Signal Process.*, vol. 2010, 2010.
- [24] A.G. Mangala and R. Balasubramani, "A Review On Vehicle Speed Detection Using Image Processing," *Int. J. Curr. Eng. Sci. Res.*, vol. 4, no. 11, pp. 23–28, 2017.
- [25] K. Kobayashi, K. C. Cheok, and K. Watanabe, "Estimation of absolute vehicle speed using fuzzy logic rule-based Kalman filter," no. March, pp. 3086–3090, 2005.
- [26] H. Pazhoumand-Dar, "Object Speed Estimation by using Fuzzy Set," *J. World Acad. ...*, vol. 4, no. 4, pp. 241–244, 2010.
- [27] J. Pouramini and A. Saeedi, "Fuzzy Model Identification for Intelligent Control of a Vehicle Speed Limit," *J. Math. Comput. Sci.*, vol. 2, no. 2, pp. 337–347, 2011.



Budi Setiyono, received the B.Sc. degree in 1997 in mathematics from the Institut Teknologi Sepuluh Nopember (ID), the M.Sc. degree in 2002, in Informatics from the Bandung Institute of Technology (ID) and PhD degree in 2012 from the

Institut Teknologi Sepuluh Nopember (ID). He is currently an Assistant Professor in the Institut Teknologi Sepuluh Nopember (ID).

His research interests are in Image and video processing, Databases and web technology.



Dwi Ratna Sulistyaningrum, received the B.Sc. degree in mathematics and the M.Sc. in Informatics, both from the Institut Teknologi Bandung (ID) and PhD degree in 2013 from the Institut Teknologi Sepuluh Nopember (ID). She is currently an Assistant

Professor in the Institut Teknologi Sepuluh Nopember (ID).

Her research interests are in Image and video processing and cryptography.



Soetrisno, received the B.Sc. degree in mathematics from the Institut Teknologi Sepuluh Nopember (ID), the M.Sc. degree in Computer Science from the Universitas Indonesia (ID). He is currently an Assistant Professor in the Institut Teknologi Sepuluh

Nopember (ID).

His research interests are in image processing and simulation modeling.



Danang Wahyu Wicaksono, received the B.Sc. and M.Sc. degree in mathematics from the Institut Teknologi Sepuluh Nopember (ID). He is currently a senior freelance programmer.

His research interests are in image and video processing.

ing.