

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/317312246>

# Speed Estimation On Moving Vehicle Based On Digital Image Processing

Article in *International Journal of Computing Science and Applied Mathematics* · March 2017

DOI: 10.12962/j24775401.v3i1.2117

---

CITATIONS

20

---

READS

4,477

2 authors, including:



**Budi Setiyono**

Institut Teknologi Sepuluh Nopember

42 PUBLICATIONS 97 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



papers [View project](#)

# Speed Estimation On Moving Vehicle Based On Digital Image Processing

Danang Wahyu Wicaksono and Budi Setiyono

**Abstract**—Along with the development of information and communication technology, the world urban people now recognize a new term called Smart City. One of Smart City components is smart transportation, known as Intelligent Transportation System (ITS) in which there is transportation management on the highway. Installation of CCTV (Closed Circuit Television) on the streets are now widely performed. It can be used to monitor conditions and detect problems such as traffic jam and vehicle speed limit violation. This research focuses on vehicle speed estimation using image processing from video data and Euclidean distance method with many different camera angles. The first step, video data is extracted into frames and applied pre-processing to extracted frames to minimize shadow effect. Then, using Gaussian Mixture Model (GMM) to extract foreground image. In the next step, the obtained foreground is filtered using median filter, shadow removing, and morphology operation. The detected vehicle object will be tracked to determine the location in each frame to estimate the speed based on its distance between frames. From the obtained results, this system is capable on estimating the speed of moving vehicle with the lowest accuracy is 87.01% and the highest accuracy is 99.38%.

**Index Terms**—Digital image processing, moving vehicle, speed estimation.

## I. INTRODUCTION

ONE of smart city components is smart transportation, known as Intelligent Transportation System (ITS) in which there is transportation management on the highway. Smart traffic monitoring system is uncompleted without the existence of system that is capable to detect any traffic problems automatically, such as traffic rules violation and traffic jam. Traffic jam can be detected from speed that is slow and followed by high volume of vehicles, so it is important to detect and estimate speed for knowing road profile based on the vehicle speed. Besides, there are many researches for classifying and counting vehicle object from video image [1] in which adding speed estimation process will give a contribution to build a smart system for traffic monitoring that is able to do detecting, counting, classifying, and estimating speed for vehicle object from video data.

In this paper, several steps are used to build a system that capable to estimate vehicle speed from video data captured from highway at fixed or static position with different camera angles to know its influence for speed estimation result. Using the advantage of background subtraction with Gaussian Mixture Model (GMM) to extract and detect moving object from video [2], contrast and brightness adjustment to minimize shadow

effect, median filtering to remove noise, and morphological operation to reconstruct broken foreground image after noise removing process for better object detection. We use Euclidean distance to calculate the distance of moving object from 2 consecutive frames by its centroid point while object is being tracked along in ROI (Region of Interest) then the distance is divided by time between frames to get estimated speed. For simulation, we use JAVA Programming with OpenCV library.

## II. RELATED WORK

There are many researches with the same topic for estimating vehicle speed based on image processing. Arash Golami R. with the paper "Vehicle Speed Detection in Video Image Sequences using CVS method" uses CVS as background model and Euclidean distance and get the highest accuracy 90% [3]. Jinhui Lan with the paper "Vehicle Speed Measurement based on Gray Constraint Optical Flow" uses optical flow and three frames differencing to measure vehicle speed and get the highest accuracy 99% [4], but as we know optical flow has high complexity algorithm and more sensitive to noise. Hardy Santosa and Agus Harjoko with the paper "Vehicle Counting and Vehicle Speed Measurement Based On Video Processing" use Euclidean distance to measure vehicle speed and get the highest accuracy 90.5% [5]. Asif Khan with the paper "Speed Estimation of Vehicle in Intelligent Traffic Surveillance System Using Video Image Processing" also uses Euclidean distance to estimate vehicle speed and get the highest accuracy 98.12% [6]. But those papers did not do research about the influence of camera angles and ROI selection for the best result for estimating vehicle speed from video data. This paper adds discussions about the influence of camera angles and ROI selection for estimating vehicle speed based on image processing. Camera angle is an important aspect for determining the best position for the camera that will be placed on the traffic.

## III. SYSTEM ARCHITECTURE

The system architecture for estimating vehicle speed from video data consists of 9 processes that are discussed further later in this paper. Each process will do particular work which the result will be used by next process until estimated speed is calculated. Block diagram for this system is given by Fig. 1.

### A. Video Input

Video data is captured from static camera at top position of highway, using geometric equation to calculate perpendicular

Manuscript received February 2, 2017; accepted February 20, 2017.

The authors are with the Department of Mathematics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia. Email: danangww@yahoo.com, budi@matematika.its.ac.id

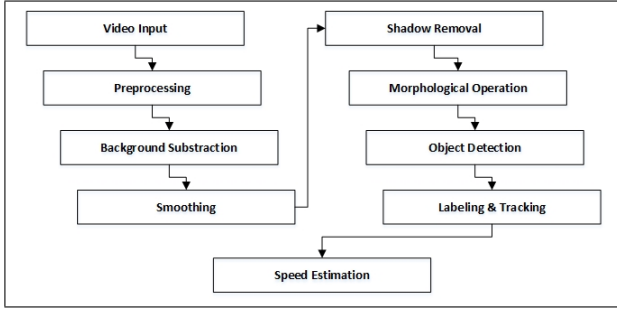


Fig. 1. Block Diagram for System Architecture

view ( $P$ ) and distance between object and camera ( $D$ ) based on camera height ( $H$ ), field of view angle of the camera ( $T_c$ ), and angle of the camera ( $T_v$ ). Perpendicular view is used to calibrate estimated speed to standard unit, km/h. Video capturing layout is given by Fig. 2. From Fig. 2, we can get

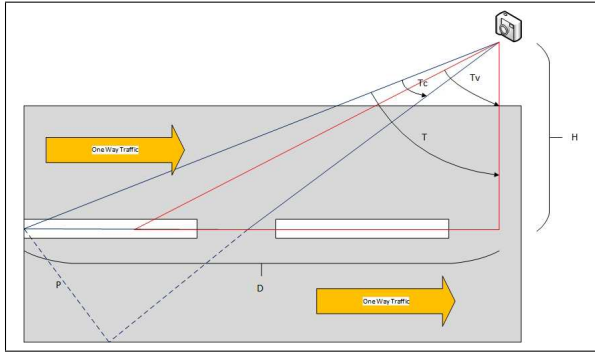


Fig. 2. Video Capturing Layout

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

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

where  $f$  is focal length of the camera and  $v$  is vertical dimension of 35 mm image format which can be found from camera specifications. Then, we can get

$$D = H \tan(T) \quad (3)$$

and

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

### B. Preprocessing

Video data is extracted to frames and each frame is applied preprocessing before frame is being processed by background subtraction. Preprocessing is used to minimize bold shadow that can be detected as solid object which can decrease object detection accuracy. This paper uses brightness and contrast adjustment for preprocessing step which has general equation given by

$$g(i, j) = \alpha f(i, j) + \beta \quad (5)$$

where  $f(i, j)$  is pixel intensity in coordinate  $(i, j)$ ,  $\alpha$  is gain parameter with default 1, and  $\beta$  is bias parameter with default

0. To minimize bold shadow which can be detected as solid object,  $\alpha$  should be decreased and  $\beta$  should be increased with a certain change of  $\alpha$  and  $\beta$  that the result can be seen from histogram of image result from background subtraction process.

### C. Background Subtraction

Background subtraction is used to extract foreground image based on background model for each frame. There are many background subtraction methods which have their own advantages and disadvantages. This paper uses Gaussian Mixture Model known as GMM method to do background subtraction process because it is a good algorithm to use for the classification of static postures and non-temporal pattern recognition, more robust than other models, and it can handle multi-modal situations [6]. To implement GMM method, we use BackgroundSubtractor class from OpenCV library. BackgroundSubtractor in OpenCV will produce foreground image with 3 gray levels, 255 for moving object, 127 for object shadow, and 0 for background.

### D. Smoothing

Smoothing is used to filter or reduce noise produced from background subtraction process. The noise generated from background subtraction is typically in salt and pepper type in which using median filter is good to reduce the noise. The illustration how median filter works is given by Fig. 3.

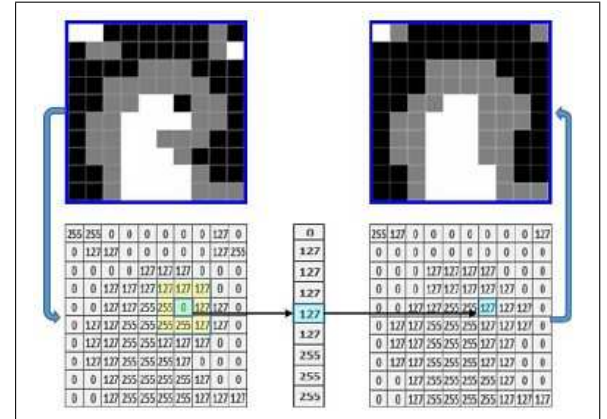


Fig. 3. Median Filter

### E. Shadow Removal

Background subtraction using GMM can also be used to detect shadow, so to increase object detection accuracy, the shadow should be removed. This process will map foreground image produced from GMM to binary image which contain only pixel value 1 and 0 for identify moving object and background respectively. Equation (6) is used to this shadow removal and mapping process and illustration is given by Fig. 4.

$$I_{binary}(i, j) = \begin{cases} 0 & \text{if } I_{gray}(i, j) = 0 \text{ or } I_{gray}(i, j) = 127 \\ 1 & \text{if } I_{gray}(i, j) = 255 \end{cases} \quad (6)$$

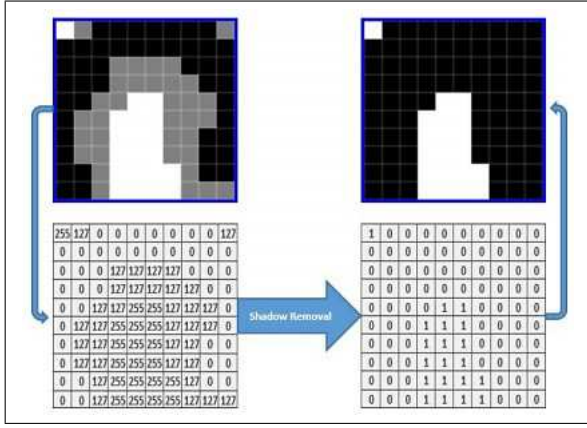


Fig. 4. Shadow Removal Process

### F. Morphology Operation

In this paper, morphology operation is used to reconstruct image after shadow removal process. Shadow removal process usually creates a gap inside object which if that gap is too large, it can make object will be separated that can affect object detection accuracy. There are two fundamental processes in image morphology, erosion and dilation which erosion will do shrinking and thinning operation to objects and dilation will grow or thicken objects in binary image. From these two fundamental processes, there are opening and closing operation that is combination of erosion and dilation. Opening is the process that do erosion firstly and then followed by dilation, generally smooths the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions. Closing is the process that do dilation firstly and then followed by erosion, generally smooths sections of contours but, as opposed to opening, it fuses narrow breaks and long thin gulfs, eliminate small holes, and fills gaps in the contours [7]. To reconstruct image which contain gaps, we will use this closing operation. Illustration of how closing operation works to an object on binary image is given by Fig. 5.

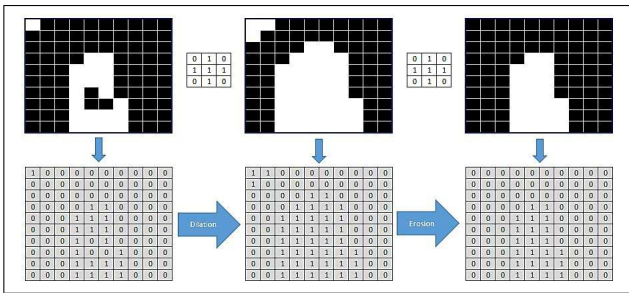


Fig. 5. Closing Operation in Image Morphology

### G. Object Detection

In this paper, object detection is done by finding object contours on the binary images. This process uses findContours function from OpenCV to detect moving objects. For each detected object, bounding box is given to identify and also to determine centroid point for the object. This centroid point is

the most important thing in this paper for speed estimation by using Euclidean distance to know the distance traveled by the object from two consecutive frames by its center point. Let  $B(i, j)$  is a bounding box which starts at coordinate  $(i, j)$ , then centroid point  $C(i, j)$  which is coordinate  $(C_i, C_j)$  can be determined by

$$C(i, j) = (C_i, C_j) \quad (7)$$

with

$$C_i = B_i + \frac{B_{width}}{2}, C_j = B_j + \frac{B_{height}}{2} \quad (8)$$

where  $C_i$  is centroid coordinate at row  $i$ ,  $C_j$  is centroid coordinate at column  $j$ ,  $B_i$  is bounding box coordinate at row  $i$ ,  $B_j$  is bounding box coordinate at column  $j$ ,  $B_{width}$  is the width of bounding box, and  $B_{height}$  is the height of bounding box. Illustration for object detection and centroid determination is given by Fig. 6.

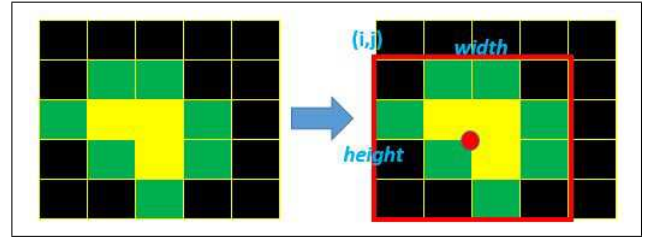


Fig. 6. Object Detection based on Contour Finding (left) and Centroid based on Bounding Box (right). Green pixels represent border of the object, red point represents centroid point, and red rectangle represents bounding box

### H. Labeling and Tracking

Labeling is a process that gives a label for the detected object to know the same object at next frame. While the object is in ROI, the object will be tracked and system will calculate the distance using Euclidean distance and then calculated the estimated speed between two consecutive frames. This speed calculation is always done while object is in ROI and being tracked, so what happen is there are more than one speed calculations and save them to a speed list for the object and when the object is out of ROI, system will calculate average speed of the object based on its speed list, so the result for estimated speed in this paper is the average speed. Speed estimation process will be discussed further in the next section after this section.

### I. Speed Estimation

Speed estimation is done by calculate the distance traveled by the object for two consecutive frames and then divided by time between two frames. To calculate the distance traveled by the object for two consecutive frames using Euclidean distance, let  $C_t(a, b)$  and  $C_{t+1}(c, d)$  is centroid point of the object in frame  $t$  and  $t + 1$  respectively. The distance  $d$  calculated by Euclidean distance is given below

$$d = \sqrt{(a - c)^2 + (b - d)^2} \quad (9)$$

To calculate speed  $V$  between two frames, we use

$$V = k \frac{d}{t} \quad (10)$$

where  $k$  is calibration coefficient based on perpendicular view  $P$  and image height  $I_{height}$  and  $t$  is time between two frames which is given by

$$k = \frac{P}{I_{height}}, t = \frac{1}{fps} \quad (11)$$

Speed  $V$  from Equation 10 above is still in unit meter per second or  $\frac{m}{s}$ , so that we have to convert it to standard unit of speed, kilometer per hour or  $\frac{km}{h}$  by multiplying 3.6 which is given by Equation 12.

$$V = 3.6k \frac{d}{t} \quad (12)$$

To illustrate how the vehicle speed is calculated while in ROI, we give an example in Fig. 7 and Fig. 8 that shows object position in frame  $t$  and  $t+1$  respectively. Then, we calculate the distance using Euclidean distance given by Fig. 9.

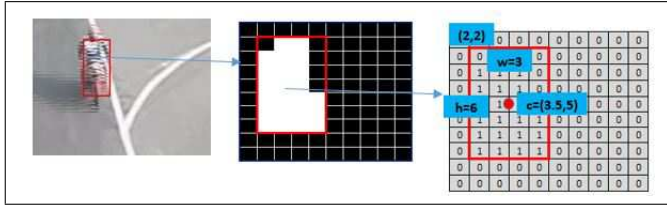


Fig. 7. Object at frame  $t$ . Left : a real object in video data. Middle : result of object detection in image processing. Right : bounding box and centroid of the object

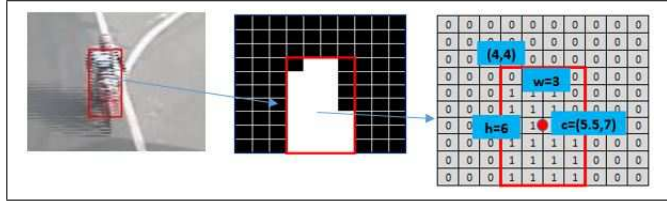


Fig. 8. Object at frame  $t+1$ . Left : a real object in video data. Middle : result of object detection in image processing. Right : bounding box and centroid of the object

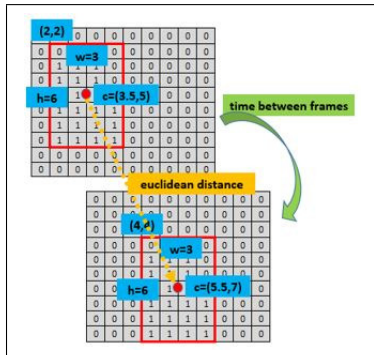


Fig. 9. Illustration for Vehicle Speed Estimation from given example at Fig. 7 and Fig. 8.

With given known parameters there are image size  $320 \times 240$ ,  $fps = 30$ ,  $T_v = 60$ ,  $T_c = 41.10$ , and  $H = 7.6$ , then we can calculate  $d = \sqrt{(5.5 - 3.5)^2 + (7 - 5)^2} \approx 2.83$  pixels,  $t = \frac{1}{30} = 0.033s$ ,  $D = 7.6 \tan(80.55) = 45.661$  meters,

$P = 2\sqrt{7.6^2 + 45.661^2} \tan(\frac{41.10}{2}) = 34.705$  meters, and  $k = 34.705/240 = 0.145$ , so the estimated speed for given example is  $V = (3.6)(0.145)(\frac{2.83}{0.033}) \approx 44.765$  km/h.

## IV. RESULTS AND DISCUSSIONS

### A. Experiment Data

Video data for experimental need in this paper is derived from acquisition process on highway according to the video capturing layout at Fig. 2. There are two aspects of data, the vehicle speed and the camera angle that is used for video acquisition. For video acquisition process, we use motorcycle which run relative constantly 15, 20, 30, 40, 50, and 60 km/h and for each speed we use 3 different camera angles there are 45, 50, and 60 degrees so there are 18 data which to be executed and to be validation data for the system in this paper. Validation process in this paper is derived from relative error calculation between estimated average speed and real average speed of vehicle.

### B. Experiment Result

As described before, video input data is extracted to frames and then each frame is processed firstly by preprocessing. Preprocessing is used to minimize solid shadow that can be detected as object in background subtraction process. We use contrast and brightness adjustment to do preprocessing. The result for this process is given by Fig. 10 which show that with preprocessing, we can get better result for object detection, the word "better" in this paper means that shadow is not detected as object. After preprocessing and background subtraction

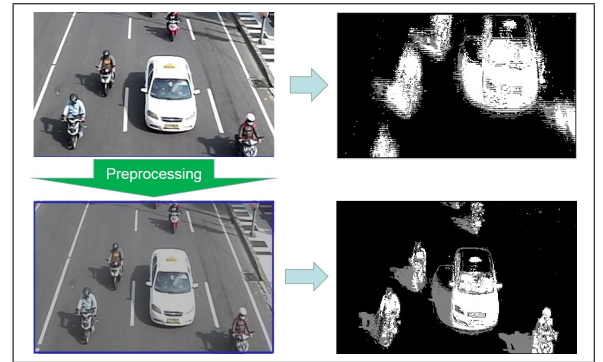


Fig. 10. Preprocessing Result on Background Subtraction GMM. Top : without preprocessing. Bottom : with preprocessing.

using GMM process, then foreground image generated from GMM is processed by smoothing to remove noise. Result for smoothing process is given by Fig. 11. The next step is shadow removal which map the image into binary image (only contain 1 and 0) using Equation 6. Result for shadow removal process is given by Fig. 12. The image result from shadow removal process can contain many gaps inside the object which can split one object into separate objects, so that it is needed to reconstruct the image using morphological closing operation. Result for morphological closing operation is given by Fig. 13. After this process, we can do object detection process using connected component and estimate speed for the object



which is given by Fig. 14 and Fig. 15 respectively. To integrate all processes together, we create an GUI interface in Java Programming to ease us to do experiment. GUI interface for the system in this paper is given by Fig. 16.

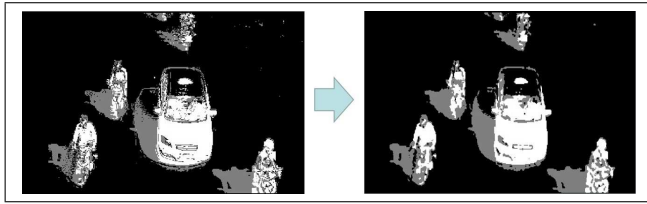


Fig. 11. Smoothing Result. Left : before smoothing. Right : after smoothing

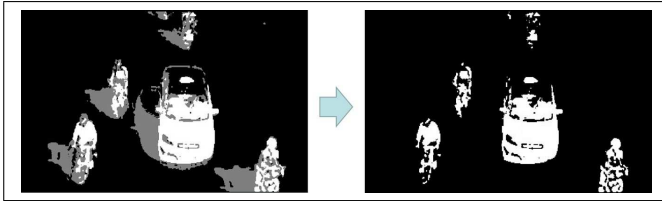


Fig. 12. Shadow Removal Result. Left : before shadow removal. Right : after shadow removal

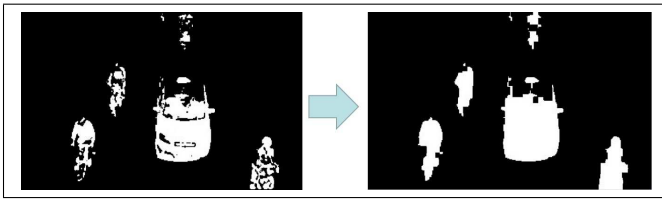


Fig. 13. Morphology Closing Result. Left : before operation. Right : after operation

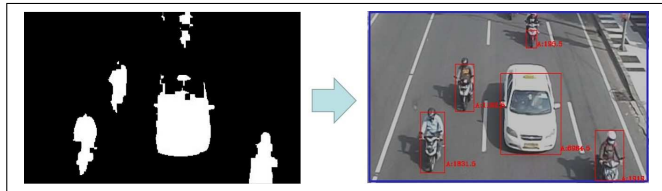


Fig. 14. Object Detection Result. Left : before object detection. Right : after object detection

Region of Interest (ROI) selection for this camera position and using Euclidean distance to calculate the distance traveled by the object is important to be checked further, because as we can see in real view, if our eyes were the camera above highway, the object that come toward us will come slightly faster as well as the object come nearer from camera. So with this perspective, we can draw three regions of video captured, there are slow, medium, and fast region, marked by color red, green, and yellow respectively, that is given by Fig. 17. To test the best ROI region to be selected in experiment section based on camera angle, we use three data from one same speed with three different camera angle to determine which is the best region for each camera angle. Result for testing ROI selection is given by chart in Fig. 18. The result that can be derived

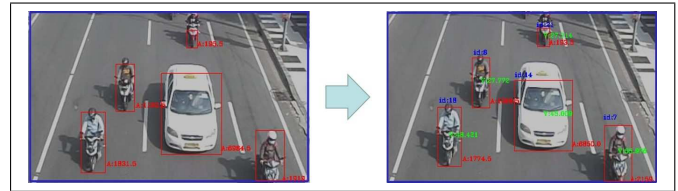


Fig. 15. Speed Estimation Result. Left : before speed estimation. Right : after speed estimation

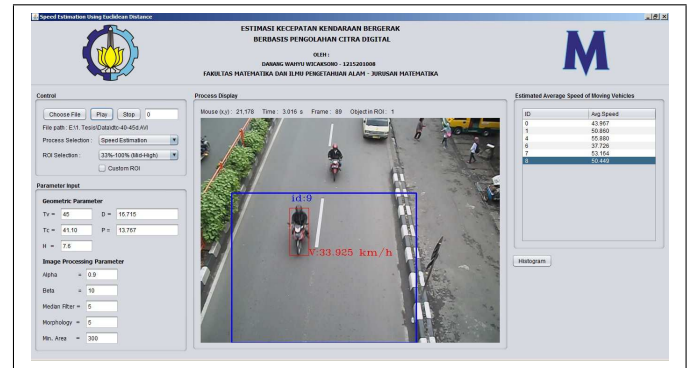


Fig. 16. System Interface

from Fig. 18, the best ROI region for 60 degree is full region, for 50 degree is medium to fast region, and for 45 degree is same as 50 degree's. Then this result will be a reference to do experiment using complete video data in the next.

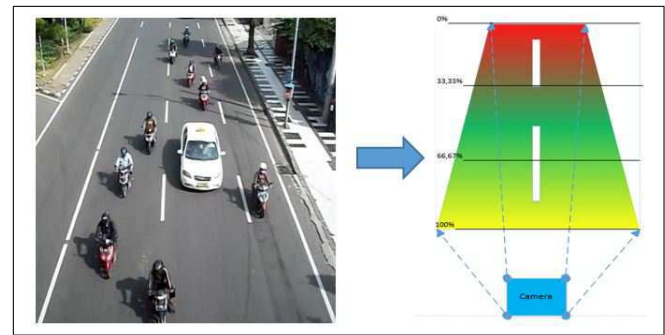


Fig. 17. ROI Regions

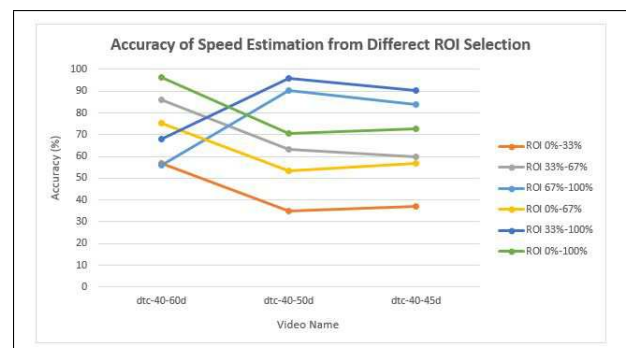


Fig. 18. ROI Selection for Speed Estimation Result

### C. Data Tables

The detail about video data that is used in this paper is described at Table 1. Each video data has detail information about  $fps$ ,  $T_v$ ,  $T_c$ ,  $H$ , and known speed of the vehicle. For example, video data's name "dtc-15-60d" means that the video is taken place at DTC which vehicle speed is 15 km/h and camera angle 60 degree. Notation  $V_e$  is estimated speed which is result from the system and  $Err_r$  is relative error from real speed and estimated speed in percentage (%). Then, we can

TABLE I  
VIDEO DATA

No	Video Name	$fps$	$T_v$	$T_c$	$H$	$V_e$	$Err_r$
1.	dtc-15-60d.avi	30	60	41.10	7.6	16.335	8.9
2.	dtc-20-60d.avi	30	60	41.10	7.6	21.424	7.12
3.	dtc-30-60d.avi	30	60	41.10	7.6	31.884	6.28
4.	dtc-40-60d.avi	30	60	41.10	7.6	38.520	3.7
5.	dtc-50-60d.avi	30	60	41.10	7.6	48.632	2.74
6.	dtc-60-60d.avi	30	60	41.10	7.6	53.487	10.86
7.	dtc-15-50d.avi	30	50	41.10	7.6	15.874	5.83
8.	dtc-20-50d.avi	30	50	41.10	7.6	21.154	5.77
9.	dtc-30-50d.avi	30	50	41.10	7.6	31.201	4.0
10.	dtc-40-50d.avi	30	50	41.10	7.6	38.348	4.13
11.	dtc-50-50d.avi	30	50	41.10	7.6	48.446	3.11
12.	dtc-60-50d.avi	30	50	41.10	7.6	53.807	10.32
13.	dtc-15-45d.avi	30	45	41.10	7.6	15.346	2.3
14.	dtc-20-45d.avi	30	45	41.10	7.6	20.124	0.62
15.	dtc-30-45d.avi	30	45	41.10	7.6	27.934	6.88
16.	dtc-40-45d.avi	30	45	41.10	7.6	36.922	7.7
17.	dtc-50-45d.avi	30	45	41.10	7.6	46.192	7.62
18.	dtc-60-45d.avi	30	45	41.10	7.6	52.201	12.99

calculate accuracy  $Acc$  in percentage from relative error by using Equation 13 and the result for accuracy is given by Fig. 19.

$$Acc = 100 - Err_r \quad (13)$$

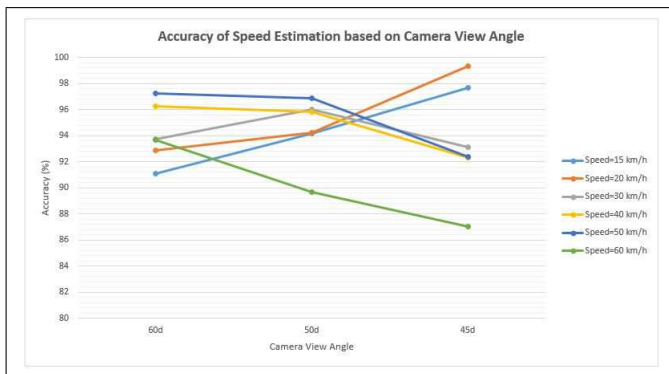


Fig. 19. Accuracy of Speed Estimation based on Camera View Angle

### D. Evaluation

- The accuracy result for speed estimation is satisfied with minimum 87.01% and maximum 99.38%.
- If we categorize our own data to be three categories which are slow speed : 15 km/h and 20 km/h, medium speed : 30 km/h and 40 km/h, and high speed : 50 km/h and 60 km/h, we can see that camera angle 60 degree gives

high accuracy for high speed, and camera angle 50 degree gives high accuracy for medium speed, and camera angle 45 degree gives high accuracy for slow speed.

- The highest error for speed estimation in this system is speed estimation on high speed vehicle.
- There are still many problems on object detection method which is still be an open problem, for example : two vehicles are near each other will be detected as one object. Problem on object detection process will affect speed estimation accuracy because the centroid point is depend on object detection result.

### V. CONCLUSIONS

Speed estimation is one of many important parts of Intelligent Traffic System (ITS) which can be done by using image processing technique. It is more efficient and economical than conventional technique without using image processing, for example : using speed radar and manual inspection. Research in this paper is a complement for other researches which are discussed the same topic in which this paper shows that there is an influence between speed estimation, camera angle, and ROI selection while using Euclidean distance. So that, this paper can be a reference for further researches about speed estimation on moving vehicle using image processing and Euclidean distance. The system in this paper acts as prototype which can be more developed and be more complex for larger system to build a complete ITS system for smart city development.

### REFERENCES

- [1] B. C. Putra, "Moving vehicle classification with fuzzy logic based on image processing," Master's thesis, Sepuluh Nopember Institute of Technology, 2016.
- [2] B. Setiyono, D. R. Sulistyningrum, I. Mukhlash, and R. A. J. Firdaus, "A new approach algorithm for counting of vehicles moving based on image processing," *International Journal of Computer Science and Information Security*, vol. 14, no. 10, pp. 366–370, 2016.
- [3] A. G. Rad, A. Dehghani, and M. R. Karim, "Vehicle speed detection in video image sequences using cvs method," *International Journal of Physical Sciences*, vol. 5, no. 17, pp. 2555–2563, 2010.
- [4] J. Lan, J. Li, G. Hu, B. Ran, and L. Wang, "Vehicle speed measurement based on gray constraint optical flow algorithm," *Optik-International Journal for Light and Electron Optics*, vol. 125, no. 1, pp. 289–295, 2014.
- [5] H. Sundoro and A. Harjoko, "Vehicle counting and vehicle speed measurement based on video processing," *Journal of Theoretical and Applied Information Technology*, vol. 84, no. 2, pp. 233–241, 2016.
- [6] A. Khan, I. Ansari, M. S. Z. Sarker, and S. Rayamajhi, "Speed estimation of vehicle in intelligent traffic surveillance system using video image processing," *International Journal of Scientific & Engineering Research*, vol. 5, no. 12, pp. 1384–1390, 2014.
- [7] R. C. Gonzalez and R. E. Woods, *Image processing*. Prentice Hall, 2007.