VIDEO-BASED DETECTION AND TRACKING MODEL FOR TRAFFIC SURVEILLANCE

Abdul Razak Al Kherret, PH.D Student
Alkherret@yahoo.com
Faculty of Engineering, Cairo University, Egypt

Al Sayed Al Sobky, Assistant Professor
assobky@hotmail.com
Faculty of Engineering, Ain Shams University, Egypt

Ragab M. Mousa*, Professor and ASCE Associate Member
E-mail: ragabmousa@yahoo.com
Faculty of Engineering, Cairo University, Egypt
(Currently: Expert, Ministry of Transport and Communications, Sultanate of Oman)
(P.O. Box 684, Postal Code 100, Muscat, Sultanate of Oman, Tel: 00968 93819248)

Paper No. 15-1465
Submitted to:
The 94th Transportation Research Board Annual Meeting January 11-15, 2015
Washington, D.C.

Submission Date: October 23, 2014

Word Count	
Abstract	250
Text	5239
Tables (3 x 250)	750
Figures (6 x 250)	1500
Total	7,489

* Corresponding Author

ABSTRACT

The continuous increase of traffic congestion in urban areas demands a high reliable traffic management system for monitoring traffic flows and providing key input parameters for predicting traffic conditions. Video sequences of road scenes are increasingly used in several contexts with an emphasis on automation, notably for tracking moving objects in a static background. This paper presents a multiple-vehicle surveillance model developed using Matlab Software for detecting and tracking moving vehicles, and collecting traffic data for different lengths of region of interest (ROI), ranging between 5 and 30 m. The model was validated using simulated video scenes, designed in VISSIM with known traffic data. Measurements from model were compared with actual measurements reported by VISSIM and results confirmed exact match of vehicle counts. Statistical t-tests of mean speed differences confirmed the model validity at 5% significance level, especially with ROI length of 10 and 15 m. Validation of headway measurements was also confirmed for optimum ROI lengths. The model processes one second in video clips of frame rate 20 frames/sec in 0.96 sec. This is appropriate for real-time applications to yield traffic parameters including vehicle speed, headway, count, incident detection, queue detection, etc. However, the model was validated assuming no lane changes and no overlap of vehicles, and, hence, model validity is limited to these assumptions. It is recommended that this model be validated using real-world videos containing noises such as light variation, shadows, vibrations due to wind, skewed views, lane changes, and/or trucks that obscure full view of vehicles.

***Keywords*:** Matlab**,** Image Processing, Traffic Surveillance, Vehicle Detection, Vehicle Tracking, Speed.

INTRODUCTION

The daily life of people encounters more problems as the population continuously increases in urban area, and road traffic becomes more congested because of high demand and less level of road capacity and infrastructure. Since the effects of these problems are significant in daily life, it is important to seek efficient solutions to reduce congestion and provide secure transportation system.

In recent years, much research has been performed for developing real time traffic monitoring systems for managing the traffic flow of roadways, prevention of accidents, providing secure transportation, etc. Within these works, one aim is to realize different applications such as estimation of vehicle speeds on the roadways, determination of traffic intensity and if necessary, to direct the vehicles to less dense roads, manage the timing of traffic lights automatically, etc. *(1-4)*.

The common method to obtain information on traffic flow is by utilizing buried induction loops. Although this existing technique is not affected by weather and light conditions, it suffers from high installation and maintenance costs *(5,6)*.

In order to overcome this limitation, vehicle tracking using image processing techniques has been adopted for traffic monitoring system to yield the traffic parameters including queue detection, incident detection, lane changes, vehicle classification, vehicle counting and vehicle speed *(5-9)*.

A more reliable traffic flow modeling and an improved understanding of drivers' behavior can be attained since the vehicle tracking system can provide more individual vehicle data such as spacing, velocity and acceleration *(5)*.

The literature is abundant with researches that dealt with detection and tracking of moving objects in video sequence, and numerous mathematical models have resulted out of these studies. Parekh et al. *(10)* reported that a general subdivision of object detection techniques can be made of three main categories, namely, (a) Background Subtraction, (b) Temporal Differencing, and (c) Optical Flow. Similarly, the object tracking can be divided into three main categories, Point tracking, Kernel tracking, and Silhouette tracking.

In object detection techniques, Background Modeling (Background Subtraction) is used to detect moving objects in an image by taking the variations between the current image and the reference background image in a pixel-by-pixel fashion. The background subtraction method uses a simple algorithm. However, it is very sensitive to changes in the external environment. Similarly, Temporal Differencing method is used to calculate the absolute differences between two consecutive images to extract moving regions and obtain a threshold function to determine changes. The temporal differencing has a strong adaptability for a variety of dynamic environments, but its method of calculation is generally difficult to achieve complete outline of moving object. The Optical Flow method uses the optical flow distribution characteristics of moving objects over time in an image sequence. Flow computation methods cannot be applied to video streams in real time because they are very complex and very sensitive to noise *(11, 12)*.

In object tracking manners, the Point tracking method is based on monitoring and comparing the positions of different detected points from one frame to another. Kernel tracking method tracks objects by calculating the motion of an object shape and its appearance in successive frames. The Silhouette tracking method uses information inside the silhouette's region in the form of edged maps to track the object using shape matching approach *(13, 14)*.

As previously mentioned, background subtraction methods are very sensitive to changes in the scene. Also, this method requires a training period absent of foreground objects and is too slow to be practical. Stauffer and Grimson *(15)* modeled each pixel in an image sequence as a mixture of Gaussians and used an on-line approximation to update the model. Then, the Gaussian distributions of the adaptive mixture model are evaluated to determine which are most likely to result from a background process. Finally, each pixel is classified based on whether the Gaussian distribution which represents it most effectively is considered part of the background model. Kaewtrakulpong and Bowden *(16)* improved the previous adaptive background mixture model by updating equations and utilized and applying different equations at different phases to make the system learn faster, more accurate, and adapt effectively to changing environments. However, Matlab Software *(17)* adopted the previous two studies and presented a system object to detect foreground using Gaussian Mixture Models (GMMs).

Nowadays, detection and tracking moving objects are becoming more essential to traffic engineers since available systems such as video image processing (VIP) are successfully used in traffic data collection and traffic surveillance. According to Martin et al. *(18)*, Klein et al. *(19)* , and Klein *(20)*, all detector technologies and particular devices have limitations, specializations, and individual capabilities. Among these technologies, only microwave radar, active infrared, and VIP systems are capable of supporting multiple lane and multiple detection zone applications. In comparison to all other technologies, VIP system is considered the best in terms of installation, maintenance and portable improvement. Moreover, this technology allows users to check visually the results by watching videos previously recorded.

A VIP system typically consists of one or more cameras, a microprocessor-based computer for digitizing and analyzing the imagery, and software for analyzing the imagery of traffic stream to determine changes between successive frames and converting them into traffic flow data (Leduc, *(21)* and Mimbela and Klein *(22)*.

Several techniques of vehicle tracking system have been investigated for traffic monitoring. The review of the literature revealed the lack of studies dealing with the impact of vehicle detection zone (ROI) on the accuracy of detections and measured speeds of individual vehicles. In fact, most of reviewed studies focused only on the vehicle speed as the prime traffic data. *(23-27)*.

It is, therefore, the objective of this research to collect traffic data of detected vehicles and assess the impact of the size of ROI on the accuracy of such collected data. This paper presents the development of a multiple-vehicle surveillance model based on the features of Matlab programming language, especially the image processing toolbox. A **V**ideo-**B**ased **D**etection **A**nd **T**racking **M**odel called "VB-DATM" was developed in the course of this research. The paper also summarizes the main findings of the model applications, features, and limitations.

DEVELOPMENT OF THE PROPOSED MODEL

This section presents the efforts devoted in developing the proposed model. The proposed model is developed using Matlab Software and consists of three sequential modules, namely, Detection, Tracking, and Traffic Data Collection as outlined below. It should be mentioned that other software such as Visual Basic and C Sharp can be used; however, Matlab is used in this work because it has an image processing toolbox.

- **Detection Module:** in this module, the moving vehicle is detected as it enters the ROI and exact time and location are registered while vehicle is traveling in the ROI. This is repeated in each frame until the vehicle leaves ROI. These steps are repeated for all vehicles in the video clip.
- **Tracking Module:** in this module, data of detected vehicles recorded in frames are checked to segregate all frames belonging to each vehicle and storing them in a separate sheet. This data is simply spatial and temporal data of vehicle trajectory while traveling in the ROI. Each data sheet belongs to only one vehicle.

- **Traffic Data Collection Module:** With the availability of spatial and temporal data of each vehicle, traffic data such as flow, speed, headway, and possibly density can be computed in this module.

The methodology executed in each module is discussed in some detail in the following sections.

Vehicle Detection Methodology

Vehicle detection is the first step prior to performing more sophisticated tasks such as tracking *(17)*. In this paper, an interactive code was written to detect and track moving vehicles in video sequence using foreground detection based on Gaussian Mixture Models. The code for detection consists of three main parts; initialization, external functions, and stream processing loop. These are elaborated in the following subsections.

Initialization

The initialization is made at the beginning of this module to:

1. Read the video from *.avi file,
2. Read and store the height and length of video frames and frame rate,
3. Display the first frame of the video clip,
4. Read the limits of the region of interest (ROI) and lane separators as interactively defined by the user on the screen using the mouse.
5. Assign variables to store the coordinates of these limits and the length of the region in pixel.
6. Read the length of the ROI in meter. This step is performed to create a Conversion Factor which is used to convert dimensions in pixels to meters and vice versa.
7. Read the expected maximum speed (km/h). The Conversion Factor and the maximum speed are used to calculate the maximum distance that a vehicle can travel in single frame (step/frame).

External Functions

To facilitate the vehicle detection and tracking process, three main functions are written as follows:

1. Checking whether the center of the bounding box around the detected vehicle is inside the polygon of ROI (or detection area) of certain lane.
2. Calculating the travel distance between two given frames. The distance traveled by a vehicle (ΔD) during two frames is calculated using Eq. 1:

$$\Delta D = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \tag{1}$$

Where (x_1, y_1) are the coordinates of the vehicle center in the first frame, and (x_2, y_2) are the coordinates in the second frame.

3. Calculating the vehicle travel distance inside the ROI. If the position of a detected vehicle is currently at Point P (xp, yp) and the start line (BC) of ROI is connecting the two points B (x_B , y_B) and C (x_C , y_C), then the slope (m) of Line BC is given by Eq. 2:

$$m = \frac{y_B - y_C}{x_B - x_C} \tag{2}$$

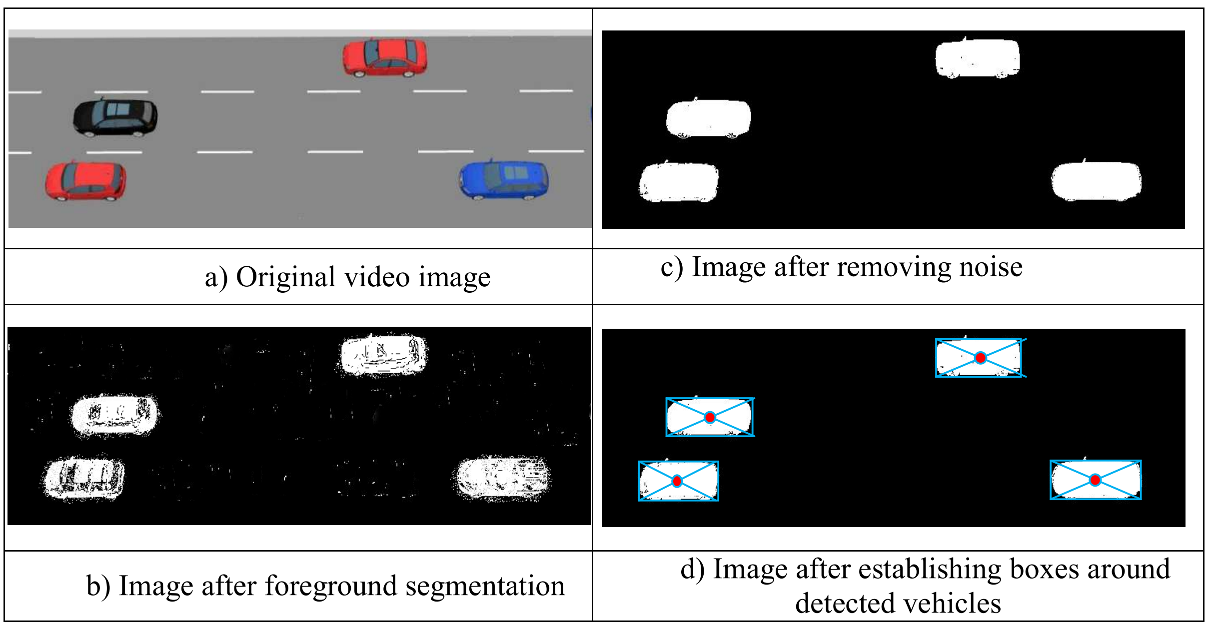
The travel distance (TD) can be calculated as the perpendicular distance from the vehicle position (P) to the start line of ROI as shown in Eq. 3 below.

$$TD = \left| \frac{m\left(x_p - x_B\right) + y_B - y_p}{\sqrt{m^2 + 1}} \right| \tag{3}$$

Stream Processing Loop

The main purpose of this loop is to detect vehicles in the input video. The processing loop uses variables created in the initialization stage and external functions mentioned above. The loop includes three stages. Stage 1 reads the input video frame and detects the foreground. Foreground detection means converting the colored frame, as shown in Figure 1-a, to a binary image (black and white image) with the same dimensions based on a ***conversation threshold***. Black areas refer to the fixed background of the scene whereas white areas (blobs) refer to the moving objects, as shown in Figure 1-b. The foreground segmentation process does not produce perfect moving objects and often includes undesirable noise due to the use of improper *conversation threshold*. In Stage 2, the noise is removed from the foreground image using special morphological filters and another proper binary image is produced as shown in Figure 1-c.

Stage 3 begins by establishing the bounding box around every blob in the foreground image and then determining the center coordinates of each box as shown in Fig. 1-d. The external functions can be used to determine the travel lane and travel distance inside ROI.



a) Original video image

c) Image after removing noise

b) Image after foreground segmentation

d) Image after establishing boxes around detected vehicles

FIGURE 1 Steps executed in the vehicle detection process

Finally, the frame number, *X*-center, *Y*-center, travel distance, and lane number are stored in “Frame-by-Frame” array. For simplicity, this frame is referred to as ***FBF*** Array. Effective frames are only those frames of vehicles inside the ROI. The “***FBF***” Array is the major output of the vehicle detection process. Table 1 shows an example of this output.

TABLE 1 An Example of Detection Data Stored in "*FBF*" Array

No.	Frame No.	X-Center	Y-Center	Lane No.	TD
..	..	…	…	.	……
91	62	801.5921	273.6767	2	25.4358
92	63	737.9049	270.7861	2	88.8748
93	64	661.2954	273.7859	2	164.8223
94	64	777.1307	197.1886	1	42.2238
95	65	593.6962	273.1075	2	232.5244
96	65	701.0692	197.3018	1	117.8907
97	66	513.2786	273.9481	2	312.1738
98	66	625.7424	197.2828	1	193.5577
99	67	553.7843	197.2428	1	265.2422
100	68	481.0581	197.4910	1	336.9268
101	69	825.4619	349.5140	3	8.6478
102	70	753.7151	349.9878	3	80.3323
103	71	674.3733	349.4387	3	158.9861
104	72	601.5645	351.7578	3	231.8533
105	73	521.0824	350.6503	3	311.4092
106	87	802.1127	270.1045	2	24.1596
107	88	737.7146	269.0123	2	88.7813
..	..	…	…	.	……

The first column in this table gives a global serial number assigned to each detection in "FBF" Array and repeated frame numbers mean records of multiple vehicles detected in the same frame.

Vehicle Tracking Methodology

In its simplest form, the vehicle tracking aims to keep track of already detected vehicles and isolate data of each vehicle. At the end of this process frames of the same vehicle are stored in a separate sheet or array called "***VEH***" Array. The number of sheets in this array is equal to the number of vehicles processed in the video, i.e., each page (or sheet) is dedicated to only one vehicle. The flowchart in Figure 2 illustrates the main steps executed during the tracking process. While in tracking process, a frame in question in "FBF" Array belongs to certain vehicle in VEH Array if the following checks with last frame of that vehicle (REF) are satisfied:

- Frame numbers of frame in question and REF are not equal,
- The distance between the two frame centers is less than the preset "Max-Step",
- The travel lane is the same for both frames, and
- The travel distance in ROI of frame in question is less than that of "REF" frame.

An example of the "VEH" Array resulting for one vehicle is shown in Table 2, which indicates that the vehicle was traveling in lane No. 2 and entered the ROI in Frame No. 62 and left it in Frame No. 66. The last column shows travel distance of this vehicle (in pixels), measured from the start of ROI.

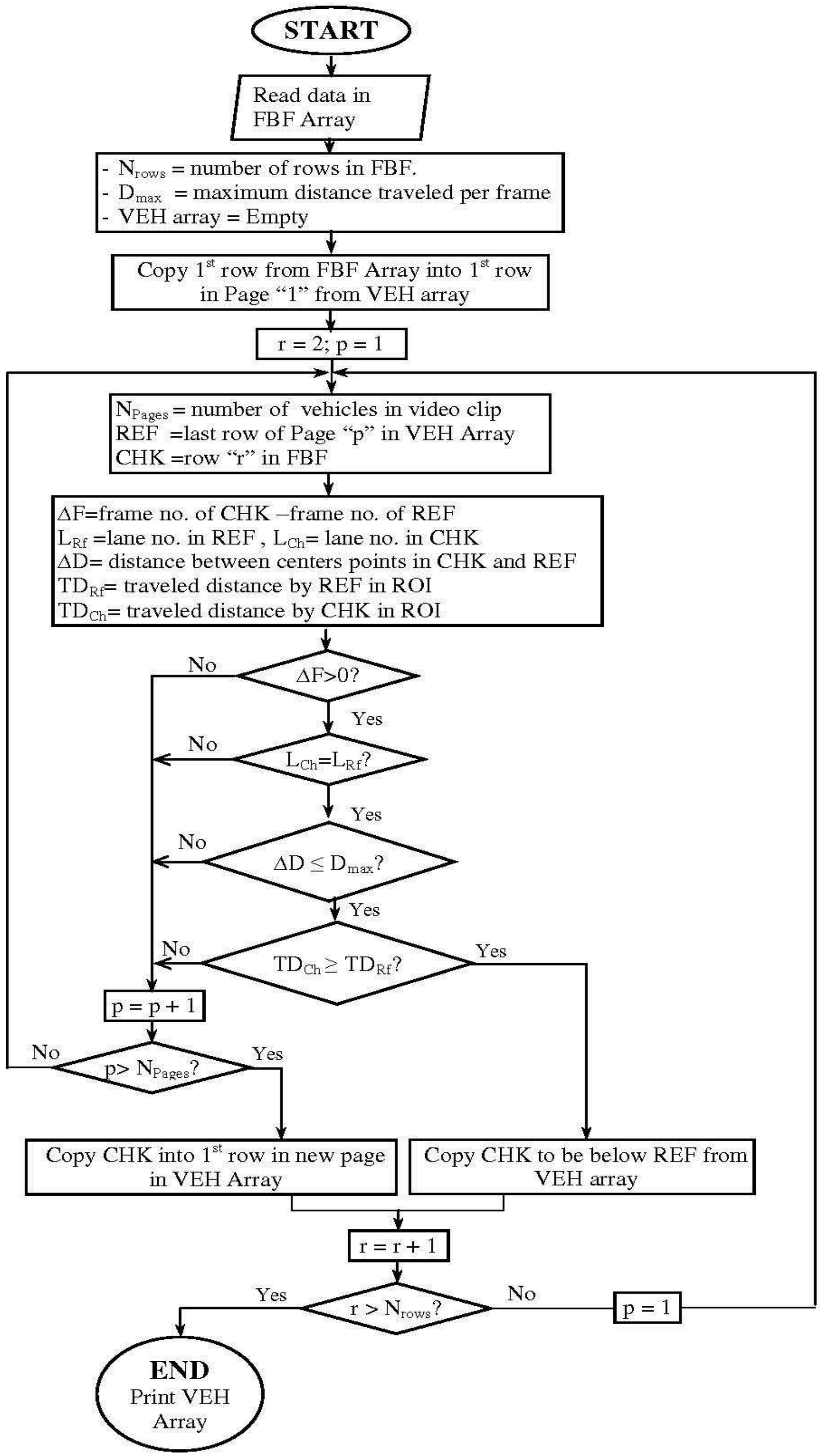


FIGURE 2 Flowchart depicting the logic of the proposed model in vehicle tracking

TABLE 2 Example of One Vehicle Tracking Data Stored in "*VEH*" Array

No.	Frame No.	X-Center (pixels)	Y-Center (pixels)	Lane No.	Distance (pixels)
91	62	801.5921	273.6767	2	25.4358
92	63	737.9049	270.7861	2	88.8748
93	64	661.2954	273.7859	2	164.8223
95	65	593.6962	273.1075	2	232.5244
97	66	513.2786	273.9481	2	312.1738

Traffic Data Collection Module

This module utilizes the proposed model in acquiring traffic data from the detected and tracked vehicles. Types of data that can be collected by the proposed model are discussed in the following sections.

Calculation of Spot Speed in the ROI

Estimating a vehicle speed depends on the data stored in "***VEH***" Array for such vehicle. The coordinates of the center point are used to measure the travel distance inside ROI in pixels, then the ***conversion factor*** **is used to** convert this distance into meters. Similarly, differences in the frames numbers determine the travel time inside ROI in terms of frames. The ***frame rate*** of video is then used to convert the travel time into seconds. For example, using Table 2, the total travel distance (*TD*) is computed as 288.3 pixels. Similarly, the total travel time (*TT*) of this vehicle is 4 frames (Leaving frame – Entry frame). Using the distance ***conversion factor*** of 0.022 (m/pixel) and frame rate of 10 frame/sec, the vehicle speed can be calculated as:

$$V = \frac{288.3136 \times 0.022}{(4/10)} = 15.9\, m/sec \qquad \text{or } 57.1 \text{ km/h}$$

Calculation of Headways in the ROI

The headway between two successive vehicles traveling in the same lane can be determined since time of the entry frame to ROI is recorded for each vehicle. Referring to Table 1, the first vehicle entered ROI in Lane no. 2 in Frame no. 62 and the next vehicle entered the same lane in Frame 87. The headway can then be calculated using the video frame rate 25/10 or 2.5 sec.

Calculations of Other Traffic Data

Other traffic data can be obtained. As such, vehicle count can be taken the same as the number of detections or headways in each lane. This traffic count can be used to calculate the traffic flow rate (Q) in vph. Average traffic speed (V) can also be computed from speeds of individual vehicles measured in the ROI. The gaps between successive vehicles can also be computed by multiplying the headway between each two vehicles by the speed of the lag vehicle. The average traffic density (K) can be computed as Q/V in veh/km.

VALIDATION OF THE DEVELOPED MODEL

This section presents the validation of the proposed model using video clips generated in VISSIM Software. The reasons for using simulation for validation include; (a) difficulty in obtaining permits in Cairo City to record video films with the required specifications and clarity, and (b) VISSIM can report the exact time and speed of each vehicle when it crosses certain data collection station. This feature eliminates the possibility of human errors in case data is extracted manually from played video clips.

Preparation of Video Clips for use in Validation

Mimbela and Klein *(22)* reported that higher mounting camera position allows for better angle and wider view, covering all lanes on the road. On the contrary, lower mounting heights would not provide effective images as some vehicles may hide part of other vehicles from the scene. As a result, video image processing recognizes overlapped vehicles as single objects. Due to this fact, camera location was selected to provide sufficient field of vision and prevent overlapping of vehicles. The camera height was set to 30 m with 1 m lateral offset from edge of road, and 65° pitch angle.

A 6-lane divided highway with 3.5 m lane width was created in VISSIM and traffic was simulated for 330 sec. Figure 3 shows the layout of this road section and the directions (Far or Near) with respect to the camera location. Lanes in each direction are labeled shoulder lane, middle lane, and median lane.

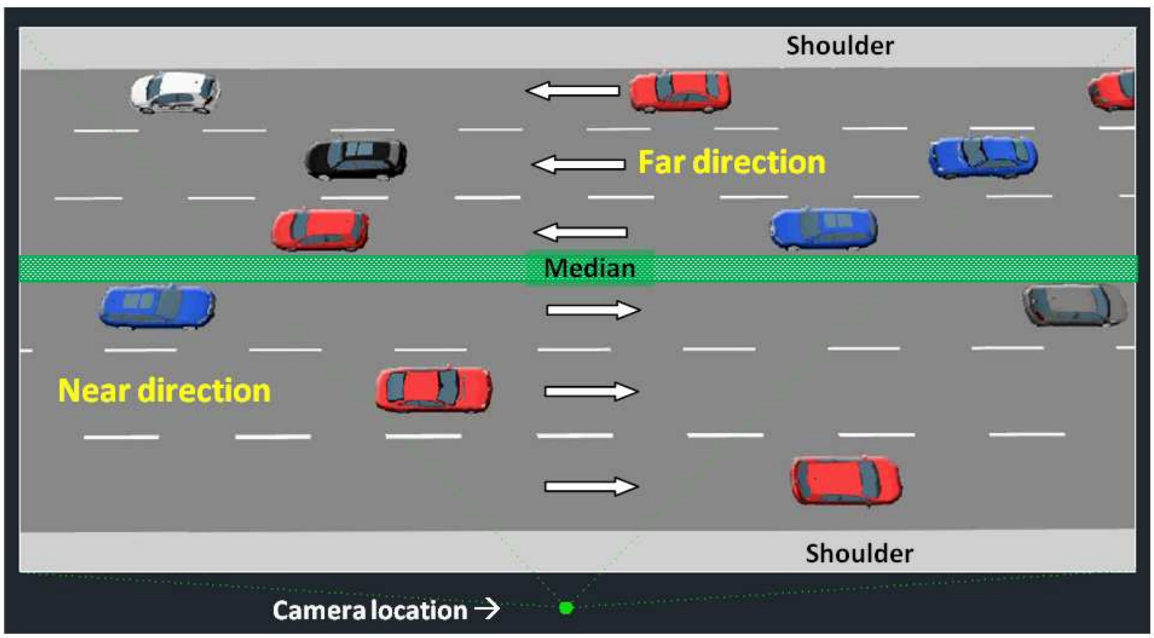


FIGURE 3 Layout of road section considered in VISSIM for producing video clips

Before the start of simulation, a section of 30 m long was considered for data collection. In particular, 13 lines were drawn across the 6 lanes at 2.5 m intervals. The total distance between the first and last lines is 30 m. The intersection points of these lines with each lane give 13 stations for data collection along each lane. This makes a total of 78 data collection stations (or points) in the 6 lanes. The Numbering and arrangement of these stations across the 6 lanes and lines is shown in Figure 4.

At the end of simulation, VISSIM creates two files. The first file is AVI format and contains the video clip of simulated traffic motion and the second file is in text format and contains speed and passing time of each vehicle at data collection points.

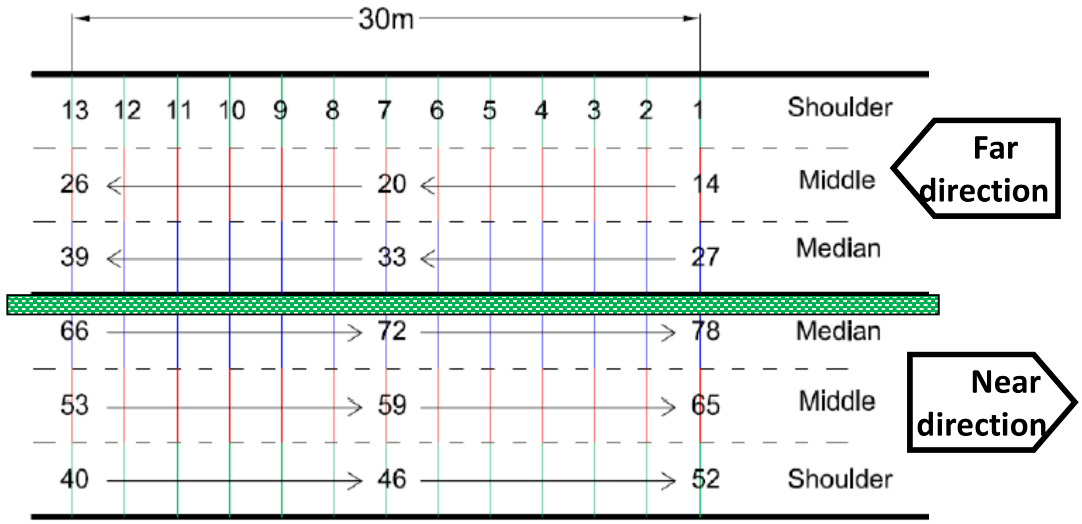


FIGURE 4 Locations of the 78 data collection points across the 6 lanes

Running the Developed Model for Data Collection

In Matlab environment, the developed model was utilized to analyze the video clip (AVI file) for different lengths of ROI, namely, 5, 10, 15, 20, 25 and 30 m. For each ROI length, the model was set up to detect, track, and collect traffic data for that ROI length. Model data are then compared with actual values provided by VISSAM. As such, when the 5 m ROI is used, the model considers the ROI to be the length between the first and third lines marked in the video clip. For comparing results of shoulder lane of Far direction, for example, traffic data obtained from the model are compared with the data obtained from VISSIM at Collection Points no. 1, 2, and 3. Similarly, when the 30 m ROI is considered, the model data for the same lane are compared with VISSIM data collected at Points no. 1, 7, and 13, and so on.

Comparative Analysis of Model Results

The developed model was run 6 times to detect and track moving vehicles in the video clip for 6 different lengths of region of interest (ROI). Results obtained from the model are compared with their counterparts in VISSIM to check model accuracy as presented in the following sections.

Comparison of Traffic Counts

The model succeeded to detect and track all vehicles simulated in the video clip for all lanes and all ROI lengths. These counts were the same as actual counts reported by VISSIM and, which were also easy to count manually in played video clip. As expected, the number of vehicles counted in each lane is identical for all ROI lengths.

Comparison of Vehicles Speed

This comparison was established to check the model accuracy in estimating vehicle speeds for all ROI lengths. This check was made in three steps. First, the check was made by plotting the speed obtained from model versus the actual speed obtained from VISSIM. As an example, Figure 5 depicts this comparison for the shoulder lane of the Near Direction and ROI of 10 m. The figure clearly shows that model speeds are very close to actual speeds (Refer to the 45° line in figure). This accuracy is applicable for all lanes and all ROI lengths.

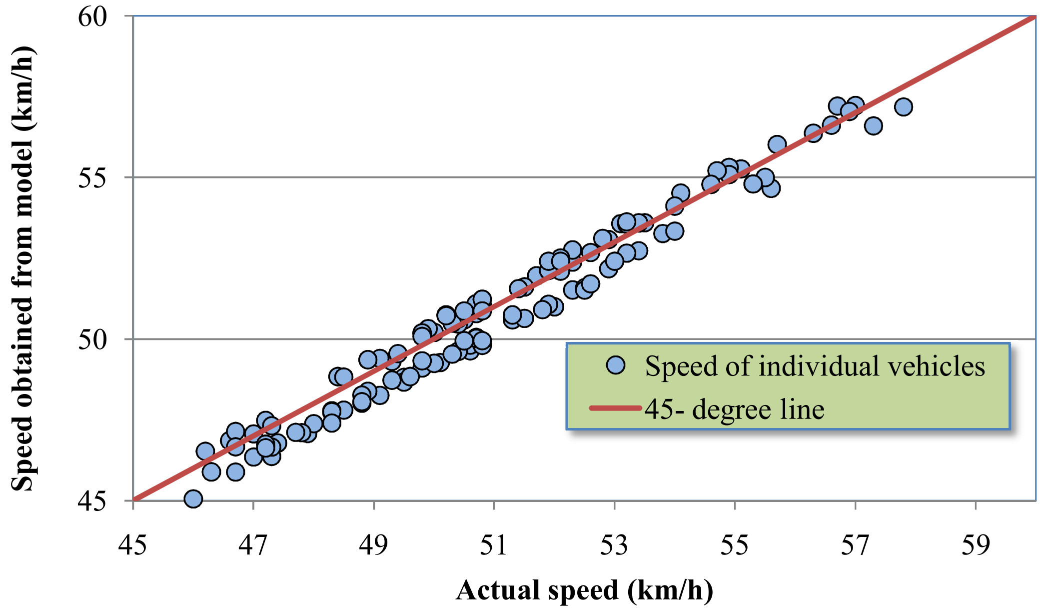


FIGURE 5 Comparison of model speed accuracy

The second step of comparison was made by computing the mean and standard deviation of the speeds obtained from model and VISSIM. The accuracy of the model estimates was expressed as the ratio of the mean speed of model and mean speed of actual values. Computed speed ratios for all traffic lanes and ROI lengths range between 96.7% and 99.5%, with higher ranges for Near direction lanes and ROI of 10 and 15. Similar ratios for standard deviation were computed, and were all greater than 1.0. This means that the developed model produces relatively higher variation in individual speeds, and this variation is higher for Far direction lanes than for Near direction lanes. The minimum variation is reported with ROI lengths of 10 and 15 m, where standard deviation ratios are close to 1.0.

The third step in comparison to conclude the model validation was to perform t-tests on the two data sets for each ROI case. Before conducting this test, however, the Kolmogorov-Smirnov test was performed on the data sets to check whether observations of each data set are normally distributed. The test results confirmed that all data are normally distributed. Results of the t-tests are shown in Table 3 for different lanes and ROI lengths.

TABLE 3 T-Test Results for Comparison of Model Speeds (*P-value*)

ROI Length	Near-Direction			Far-Direction		
	Shoulder Lane	Shoulder Lane	Shoulder Lane	Shoulder Lane	Middle Lane	Median Lane
5 m	0.049	0.108	0.004	0.008	0.005	0.002
10 m	0.534	0.499	0.470	0.170	0.136	0.159
15 m	0.431	0.351	0.324	0.163	0.131	0.363
20 m	0.021	0.063	0.017	0.032	0.007	0.036
25 m	0.086	0.083	0.299	0.058	0.002	0.033
30 m	0.117	0.011	0.018	0.000	0.001	0.000

Based on the results of the t-tests, it can be concluded that the model produces speeds that are not significantly different from actual speeds for ROI lengths of 10 and 15 m, and this is valid for all 6 lanes. It is interesting to note that although the model produces speeds having average speed greater than 96.5% of actual mean speed for ROI of 5, 20, 25, and 30 m, the difference between the two means was significant at the 5% level.

Comparison of Measured Headways

Measuring the individual headways (or gaps) between successive vehicles is essential because they indicate how vehicles are dispersed and whether they follow too close from each other. It can also help detect the presence of queues on the road. Beside applications in safety issues, the number of headways can be directly used to compute the traffic flow rate (Q). With the average speed (V) and flow rate (Q) available, one can readily calculate the average traffic density (K) as the subdivision of Q and V (or K = Q/V).

The model accuracy was also checked for optimum lengths of ROI, namely, 10 and 15 m. Figure 6 depicts a comparison of the model accuracy in measuring headways. The t-test was also run on measured headways and results confirm the model validity. All test results confirm that the differences between mean headway produced by the model and mean actual headway are not significant at the 5% significance level. Furthermore, the t-test for matched pairs also confirmed the same finding.

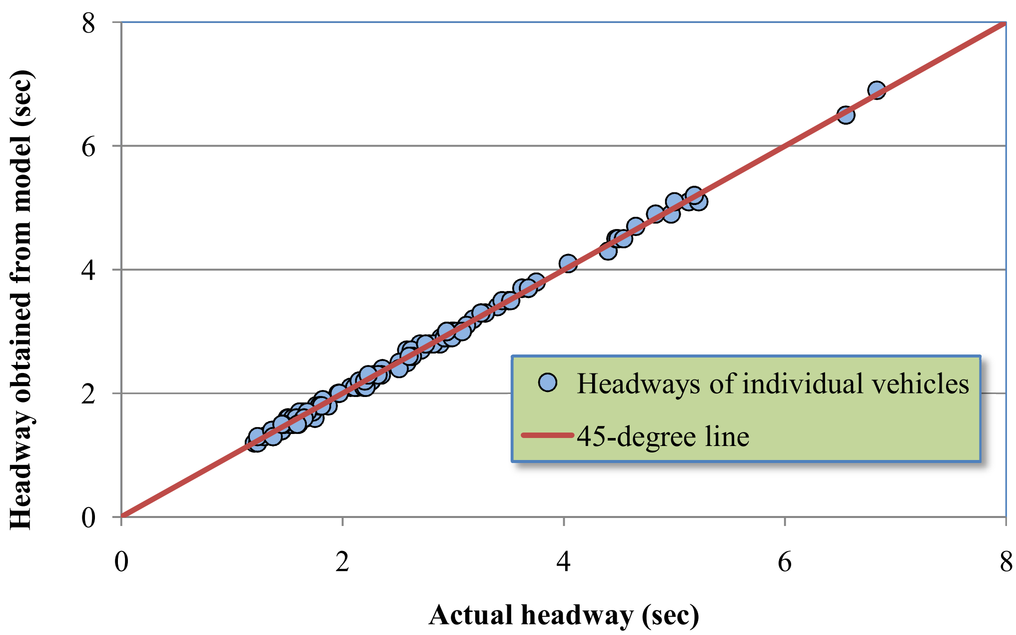


FIGURE 6 Comparison of model headway accuracy

MODEL PROCESSING TIME

The execution time of processing 20 fps video clips of resolution 1280×720 pixels was about 1.48 sec/frame or 29.6 sec/sec for processing the 20 frames. The time taken to detect and identify the foreground comprises about 92% of total execution time. Attempts were made to minimize the total execution time while maintaining almost the same accuracy. These included processing every other frame, which reduces the execution time by half (14.8 sec/sec). Attempts also included reducing image resolution to 640 x 360 pixels, which further reduced the execution time by a factor of 0.206 (3.04 sec/sec). In order to match real-time processing, the image was cropped to the ROI zone only. Cropping image means processing the ROI only and neglecting the rest of the image. After cropping the image, the execution time was reduced to 0.96 sec/sec when all 20 frames are processed. Since this execution time is less than 1.0 sec/sec, the model is capable of processing real-time applications. T-tests were performed to compare values of original data set with their counterparts after reducing execution time (i.e., after reducing resolution & cropping image). Results confirmed that there are no significant differences between the means of two data sets at the 5% significance level.

CONCLUSIONS AND RECOMMENDATIONS

This paper presents a multiple-vehicle surveillance model, written in Matlab programming language, for vehicle detection, tracking, and collection of traffic data. The model was validated by processing video clips created in VISSIM. Based on the analysis presented in this paper, it can be concluded that the proposed model is a valuable tool for collecting essential traffic data such as speed, flow, and headways, which can save time and resources. The model produces its best results with optimum ROI lengths, namely, 10 and 15 m. The model processes each second of video clip having 20 fps in 0.96 second. This rate qualifies the developed model for real-time applications such as incident detection, detection of queues, intelligent transportation system, etc. However, the model was validated using ideal conditions in video clips of simulated traffic streams. Accordingly, the model applications might be limited to such assumptions. It is recommended that model be validated using real-life video clips containing noises such as lane changes, light variation, shadows, vibration due to wind, skewed views, and/or trucks that obscure full view of vehicles.

ACKNOWLEDGMENT

This research is part of a Ph.D. research conducted in the Faculty of Engineering, Cairo University, Egypt. Simulation modeling was performed using licensed VISSIM Software in the Transportation Laboratory, Faculty of Engineering, Ain Shams University.

REFERENCES

1. Cathey, F.W.; Dailey, D.J. A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, Las Vegas, NV, USA, 6-8 June 2005; pp.777-782.
2. Grammatikopoulos, L.; Karras, G.; Petsa, E. Automatic estimation of vehicle speed from uncalibrated video sequences. In *Proceedings of the Int. Symposium on Modern Technologies, Education and Professional Practice in Geodesy and Related Fields*, Sofia, Bulgaria, 3-4 November 2005; pp. 332-338.
3. Melo, J.; Naftel, A.; Bernardino, A.; Victor J.S. Viewpoint independent detection of vehicle trajectories and lane geometry from uncalibrated traffic surveillance cameras. In *Proceedings of the International Conference on Image Analysis and Recognition*, Porto, Portugal, 29 September-1 October 2004; pp. 454-462.
4. Morris, B.; Trivedi, M. Improved vehicle classification in long traffic video by cooperating tracker and classifier modules. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, Sydney, Australia, 22-24 November 2006.
5. B. Coifman, D. Beymer, P. McLauchlan and J. Malik, "A real-time computer vision system for vehicle tracking and traffic surveillance," *Transportation Research Part C*, vol.6, no. 4, pp. 777-782,1998.

6. D. Bauer, A. N. Belbachir and N. Donath, et al., “Embedded Vehicle Speed Estimation System Using an Asynchronous Temporal Contrast Vision Sensor,” *EURASIP J. on Embedded Systems*, vol. 2007, Article ID 82174, 12 pages, 2007.
7. J. C. Tai, S. T. Tseng, C. P. Lin and K. T. Song, ”Real-time image tracking for automatic traffic monitoring and enforcement applications,” *Image Vis. Comput.*, vol. 22, pp. 485, Jun 2004.
8. D. J. Dailey, F. W. Cathey and S. Pumrin, “An algorithm to estimate mean traffic speed using uncalibrated cameras,” *IEEE Trans. On Intelligent Transportation Systems*, vol. 1, no. 2, pp. 98-107, June 2000.
9. Sheikh, K. and Abu-Bakar, S. Three-dimensional pose estimation from two-dimensional monocular camera images for vehicle classification. *Proc. of 6th WSEAS International Conference on Circuits, Systems, Electronics, Control and Signal Processing*, Dec 2007, pp. 356-361.
10. Parekh, H.S., Thakore, D. G., and Jaliya, U. K. A Survey on Object Detection and Tracking Methods. *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE),* Vol. 2, Issue 2, February 2014.
11. Athanesious, J., and Suresh, P. Systematic Survey on Object Tracking Methods in Video. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),* Volume 1, Issue 8 October 2012, 242-247.
12. Kastrinaki, V., Zervakis, M., and Kalaitzakis, K. A survey of video processing techniques for traffic applications. *Image and Vision Computing* 21 (2003), pp. 359–381.
13. Yilmaz, A., Javed, O., and Shah, M. Object tracking: A survey. *ACM Computing Surveys*, Vol. 38, No. 4, Article, pp. 266-280, December 2006.
14. Hu, W., Tan, T., Wang, L., and Maybank, S. A Survey on Visual Surveillance of Object Motion and Behaviors. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews,* Vol. 34, No. 3, August 2004.
15. Stauffer, C. and Grimson, W. L. Adaptive Background Mixture Models for Real-Time Tracking. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference,* Vol. 2 (06 August 1999), pp. 2246-252.
16. Kaewtrakulpong, P. and Bowden, R. An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection. *In Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems,* AVBS01, Video Based Surveillance Systems: Computer Vision and Distributed Processing (September 2001)
17. *MATLAB User’s Guide* (2012). MathWorks. www.mathworks.com.
18. Martin, P., Feng Y., and Wang X. Detector Technology Evaluation. Publication UT-03.30. Utah Department of Transportation, Salt Lake City, UT, (2003).
19. Klein, L. A., Mills, M. K., and Gibson, D. P. *Traffic Detector Handbook. Third Edition* - Volume I. Turner-Fairbank Highway Research Center, McLean, VA, (2006).
20. Klein, L. *Sensor Technology and Data Requirements for ITS. Artech House*, Boston, (2001)
21. Leduc, G. Road Traffic Data: *Collection Methods and Applications*. Institute for Prospective Technological Studies, (2008).
22. Mimbela, L. and Klein, L. A Summary of Vehicle Detection and Surveillance Technologies Used in Intelligent Transportation Systems. *FHWA Intelligent Transportation Systems Program Office*, (2007).
23. Rad, A. G., Dehghani, A., and Karim, M. R. Vehicle speed detection in video image sequences using CVS method. *International Journal of the Physical Sciences,* Vol. 5(17), pp. 2555-2563.

24. Rahim, H. A., Sheikh, U. U., Ahmad, R. B., and Zain, A. S. Vehicle Velocity Estimation for Traffic Surveillance System. *World Academy of Science, Engineering and Technology,* Vol:4 2010-09-22. Pp. 686 - 689.
25. Grammatikopoulos, L., Karras, G., Petsa, E. Automatic Estimation of Vehicle Speed from Uncalibrated Video Sequences. *International Symposium on Modern Technologies, Education and Professional Practice in Geodesy and Related Fields* Sofia, 03 – 04 November, 2005, Pp. 332-338.
26. Temiz, M. S, Kulur, S., Dogan, S. Real Time Speed Estimation from Monocular Video. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* Volume XXXIX-B3, 2012. XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia.
27. Dailey, D. J. and Li, L. Algorithm for Estimating Mean Traffic Speed with Uncalibrated Cameras. In *Transportation Research Record: Journal of the Transportation Research Board*, *No. 1719*, *Transportation Data, Statistics, and Information Technology*. Transportation Research Board of the National Academies, Washington, D.C., 2000, Pp. 27-32 .