# Video Size Comparison for Embedded Vehicle Speed Detection & Travel Time Estimation System by Using Raspberry Pi

I.Iszaidy[1], A.Alias[2], R.Ngadiran[3], R.B.Ahmad[4], M.I.Jais[5], D.Shuhaizar[6]
[1,2,3,4,5,6]affiliations Embedded,Network and Advance Computing (ENAC)
School of Computer and Communication Engineering
Universiti Malaysia Perlis
Pauh Putra,Perlis,Malaysia
iszaidyismail@gmail.com[1], aqilahalias18@gmail.com[2],
ruzelita@unimap.edu.my[3], badlishah@unimap.edu.my[4],
ilamanjais86@gmail.com[5],shuhaizar@gmail.com[6]

***Abstract*—As traffic continues to grow up, the issue regarding the road accident also growing quickly. The accident occurred due to the high speed of vehicles on the road. This paper proposed a vehicle speed detection and travel time estimation system using Raspberry Pi to estimate the speed of passing vehicles through this system. The system is designed to detect the moving vehicles and calculate its velocity. The system used OpenCV as an image processing software to detect and track the moving vehicles. Several types of capturing size of the video are used in this system to check and measure the performance of the embedded board.**

***Keywords*—travel time;vehicle speed;embedded linux;raspberry pi;**

## I. Introduction

Nowadays, the number of transport in the roadway rapidly grows due to the advanced technology that keeps on manufactured cars with the best quality. As the number of vehicles continues to grow on our roadway systems, accident always occurred because of the high speed moving vehicles. Due to this issue, providing reliable speed detection and travel time estimation system to road users has emerged as one of the most critical challenges for traffic information systems. However, measuring the speed of the vehicles is a very complex and difficult task, as the resulting accuracy varies with many variables of time-varying nature, including the day-to-day traffic demands, responses of individual drivers to daily commuting congestion, conditions of the road facility, weather, incidents, and reliability of available detectors[1].

Advanced technology offers a various technique for collecting data to merge with the existing traffic information system. However, different devices provide different result and accuracy. Some of the techniques are based on software and some of it are based on hardware. The early technology that's being used to estimate travel time are toll-tag matching[2], license plate number tracking[3]–[5] been used to tackle this issue. As the technology keeps growing, travel time could be estimated or predicted by using Global Positioning System (GPS)[6]–[8], Bluetooth[9]–[11], wireless sensor network (WSN)[12], radio frequency (RF)[13], image processing[14], [15] and so on. All of these devices have its own characteristics which make them suitable to be used in certain road situations.

The development of a traffic monitoring system, especially on the highway, which contains a huge number of vehicles and faster speed vehicles needs a few numbers of powerful travel time estimation system. Most of the existing systems are based on personal computers(PC) due to image processing which need a powerful processing board. However, the portability of PC is limited by its weight, size and the high power consumption. Thus results in that the using PC as travel time estimation system is confined in a few fields, and it is inconvenient to install especially on the highway. The way to get rid of the limit of PC is using embedded board[**?**].

The system composed of hardware(embedded board) and software(image processing). Image processing applications require the detection module for detecting the moving objects in the observed scene before applying any further technique for object recognition and activity identification[16].Background subtraction is one of the simplest methods to be used and tested as image processing for object detection. Background subtraction is a widely used technique for object detection, especially from static camera. The technique used the difference between the current frame and the previous frame, often called as the background image.The background image contains of the scene which have no moving objects and must be updated regularly for adapting the chance of luminance conditions and geometry settings.

Traffic monitoring systems which used background subtraction in their system is presented in [17]–[20].Cheung et al. compare several background subtraction algorithms for detecting moving vehicles and pedestrians in urban traffic video sequences.Several algorithm been used to find out which algorithm able to produce good result with much lower computational complexity[17].In [18], Mandellos et al. presented a robust system with a reconstruction algoritm for detecting

and extracting vehicles in the traffic surveilance scene.The proposed system able to locate moving objects in a complex road scene by implementing the advanced background subtraction technique.The advanced background subtraction able to detect vehicles in various conditions including unstable lighthing, different view-angles and congestion.Komagal et al. used background subtraction for moving object detection in video streams[19]. The proposed technique has been compared with Frame difference, Approximate Median and Mixture of Gaussian method. The good performance techiques are used for real time tracking.In 2012, Unzueta et al. presented a robust vision system for vehicle tracking and classification devised for traffic flow surveilance[20].In [20], they proposed a robust adaptive multicue background subtraction strategy which able to detect foreground pixels of moving and stopped vehicles even in challenging situations or noisy in images.

All above techniques try to estimate the background model efficiently from the temporal sequences of the frames. The proposed system must be robust against changes in illumination. Second, it should avoid detecting non-stationary background objects such as swinging leaves, rain, snow, and shadow cast by moving objects. Finally, its internal background model should react quickly to changes in background, such as starting and stopping of vehicles. Due to the low resources of embedded platform, the background subtraction technique is a suitable technique to reduce lag and slow performance of embedded platform.

The work carried out in this paper focuses on the comparison between capturing size of the video by the proposed system. The remainder of this paper is organized as follows.Section II discussed the system hardware and the Section III continue discussing about the system's software. While in Section IV introduced the result of the system focussed on CPU and memory performances. Finally, conclusion and future work development are shared in Section V.

## II. System Hardware

The Figure 1 shows the component that has been used in order to make this system work. Raspberry pi works as the main processing board and it is used to process the current frame. Raspberry Pi is furnished with two USB2.0 ports. These are joined with the LAN9512 combo center point/Ethernet chip IC3, which is itself, a USB Gadget associated with the single upstream USB port on BCM2835. The USB Ports empower the connection of peripherals, for example, consoles, mouse, webcams that furnish the Raspberry Pi with extra usefulness. The camera is connected via USB port and used to grab the current frame before it's been processed by the Raspberry Pi. After the result been calculated by Raspberry Pi, the speed of the vehicle is displayed on the monitor. The vehicle speed result can be used to calculate the travel time and user can plan their trip without exceeding their vehicle speed.

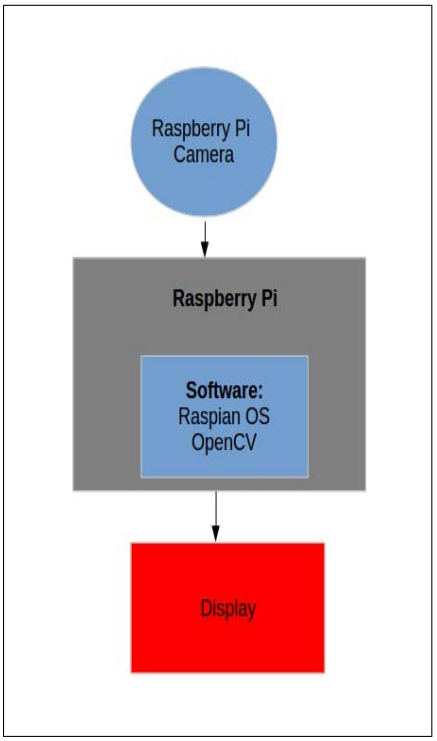


Fig. 1. Overall system design

## III. System Software

On the software part, Raspian OS is used as the operating system for Raspberry pi and OpenCV-Python binding is installed on it. OpenCV (Open Source Computer Vision Library) is an open source PC vision and machine learning programming library. OpenCV was developed to give a commonplace system to PC vision applications and to animate the usage of machine perception in the business items.The OpenCV library contains more than 2500 optimized codes and algorithms such as face detect, tracking movements, video capturing, extraction of 3D model of objects, produce 3D point clouds from stereo camera, Hough transform, etc[21].

The program is run by extracting the vehicle speed information from the a video file. The video file consists of multiple cars traveling on a roadway. The position of the motion in the video is on the x -axis. The speed of the cars is obtained through the video. The cars detected in the video frame is captured and crop into images. The images are displayed in JPEG format with the calculated speed display on it.

Figure 2 shows the overall flowchart for this system. The system start by initializing itself and then read the current frame from the video for vehicle detection purposed. When the car is detected in the frame, the system starts to get the initial pixel coordinates of the car. Next, the system will track the car and get the next pixel coordinates on the others frames. Lastly, the system will calculate the speed of the vehicle and display it.

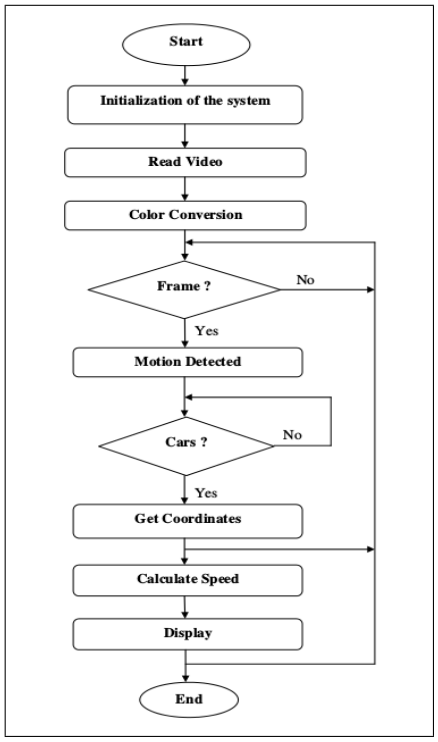


Fig. 2. Overall flowchart of the system

Fig. 3. The image of captured car by the system

## IV. RESULT

Figure 3 shows the image of the captured car from the video by the system. The number that been written on the image is the actual speed of the vehicle.

Figure 4 shows the CPU performance of Raspberry Pi 2 when three types of capturing size have been applied to it for vehicle speed measurement. From the result, three types of capturing size(320p, 540p, 720p) used almost same CPU usage even though the size of video capturing is different. While the result is different in Figure 5. The result of memory performances when different capturing size has been applied to Raspberry Pi 2 is shown in Figure 5. In this result, the 720p capturing size of the video used about 600 MB of memory usage, which is higher than 320p and 540 capturing sizes.

From both results, it can be concluded that Raspberry pi 2 is used about 25% of its CPU usage and 600 MB/1000MB of its memory usage to run this system successfully.

## V. CONCLUSION

Vehicle speed detection and travel time estimation system on the road is quite complex and needs a very high specification

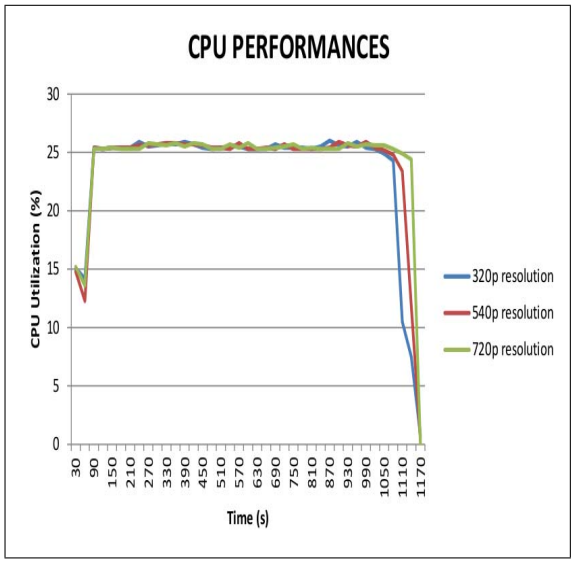


Fig. 4. CPU performance for different capturing size

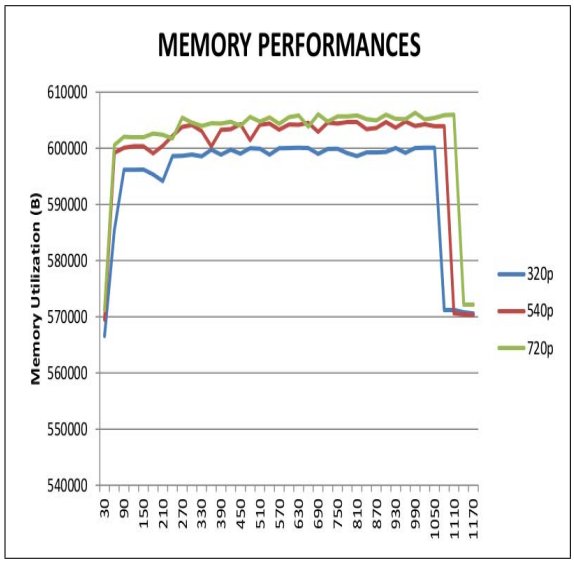


Fig. 5. Memory performances for different capturing size

of the computer. Hence the embedded technology is rapidly developing; it is possible to use Raspberry Pi as the processing board to measure and estimate the vehicle speed and travel time. Even though the image processing required a lot of resources, the OpenCV library is another alternative to handle complex image processing. For the future work, the threading technology, which able to handle I/O(Input output) heavy task can be used. The system needs something to grab the current frame quickly to update the latest result.

## REFERENCES

[1] N. Zou, "A Reliable Travel Time Prediction System With Sparsely Distributed Detectors," Ph.D. dissertation, University of Maryland, 2007.

[2] H. Yamazaki, N. Uno, and F. Kurauchi, "The effect of a new intercity expressway based on travel time reliability using electronic toll collection data," *IET Intelligent Transport Systems*, vol. 6, no. 3, p. 306, 2012.

[3] S. Bera and K. V. K. Rao, "Estimation of origin-destination matrix from traffic counts: The state of the art," *European Transport - Trasporti Europei*, vol. 49, no. 49, pp. 3–23, 2011.

[4] Y. Tanaka, "Travel-time data provision system using vehicle license number recognition devices," in *Proceedings of the Intelligent Vehicles '92 Symposium*, 1992.

[5] D. C. D. Chen, K. Z. K. Zhang, and T. L. T. Liao, "Practical travel time prediction algorithms based on neural network and data fusion for urban expressway," *Natural Computation (ICNC), 2010 Sixth International Conference on*, vol. 4, no. Icnc, pp. 1754–1758, 2010.

[6] M. Gao, T. Zhu, X. Wan, and Q. Wang, "Analysis of travel time patterns in urban using taxi GPS data," in *Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCom 2013*, 2013, pp. 512–517.

[7] K. Osamura, A. Yumoto, and O. Nakayama, "Vehicle speed estimation using video data and acceleration information of a drive recorder," *2013 13th International Conference on ITS Telecommunications (ITST)*, pp. 157–162, 2013.

[8] S. V. Kumar, L. Vanajakshi, and S. C. Subramanian, "A Model Based Approach to Predict Stream Travel Time using Public Transit as Probes," *IEEE Intelligent Vehicles Symposium (IV)*, no. IV, pp. 101–106, 2011.

[9] M. Blogg, C. Semler, M. Hingorani, and R. Troutbeck, "Travel Time and Origin-Destination Data Collection using Bluetooth MAC Address Readers," in *Australasian Transport Research Forum(ATRF) 2010 Proceedings*, no. October, 2010, pp. 1–15.

[10] P.-a. Laharotte, R. Billot, E. Come, L. Oukhellou, A. Nantes, and N.-e. E. Faouzi, "Spatiotemporal Analysis of Bluetooth Data : Application to a Large Urban Network," *IEEE Transaction On Intelligent Transportation Systems*, pp. 1–10, 2014.

[11] A. Bhaskar, M. Qu, and E. Chung, "Bluetooth Vehicle Trajectory by Fusing Bluetooth and Loops : Motorway Travel Time Statistics," *IEEE Transactions On Intelligent Transportation System*, vol. 16, no. 1, pp. 113–122, 2015.

[12] M. Saqib, S. D. Khan, and S. M. Basalamah, "Vehicle Speed Estimation using Wireless Sensor Network," *The First International Conference on Advanced Communication and Computation(INFOCOMP 2011)*, pp. 98–102, 2011.

[13] N. Kassem, A. E. Kosba, and M. Youssef, "RF-based vehicle detection and speed estimation," in *2012 IEEE 75th Vehicular Technology Conference*, 2012, pp. 1–5.

[14] C. Pornpanomchai and K. Kongkittisan, "Vehicle speed detection system," in *2009 IEEE International Conference on Signal and Image Processing Applications(ICSIPA09)*, 2009, pp. 135–139.

[15] Y. Goda, S. Chen, and L. Zhang, "New Vehicle Speed Measurement System with Image Processing," in *Proceedings of The 2nd International Conference on Intelligent Systems and Image Processing 2014*, no. 2, 2014, pp. 320–325.

[16] P. Spagnolo, T. D. Orazio, M. Leo, and a. Distante, "Moving object segmentation by background subtraction and temporal analysis," *Image and Vision Computing*, vol. 24, no. 5, pp. 411–423, 2006.

[17] S.-C. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," in *Proceedings of Video Communications and Image Processing, SPIE Electronic Imaging*, 2004, pp. 881–892.

[18] N. a. Mandellos, I. Keramitsoglou, and C. T. Kiranoudis, "A background subtraction algorithm for detecting and tracking vehicles," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1619–1631, 2011.

[19] E. Komagal and A. Vinodhini, "Real time Background Subtraction techniques for detection of moving objects in video surveillance system," *2012 International Conference on Computing, Communication and Applications (ICCCA)*, pp. 1–5, 2012.

[20] L. Unzueta, M. Nieto, A. Cortés, J. Barandiaran, O. Otaegui, and P. Sánchez, "Adaptive multicue background subtraction for robust vehicle counting and classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 527–540, 2012.

[21] L. Ujjainiya and M. K. Chakravarthi, "Raspberry Pi Based Cost Effective Vehicle Collision Avoidance System Using Image Processing," *ARPN Journal of Engineering and Applied Scieces*, vol. 10, no. 7, pp. 3001–3005, 2015.