PAPER • **OPEN ACCESS**

The Use of Background Subtraction Method in Model Design of Traffic Light Timer Control System

To cite this article: S Purwiyanti *et al* 2019 *J. Phys.: Conf. Ser.* **1376** 012020

The Use of Background Subtraction Method in Model Design of Traffic Light Timer Control System

S Purwiyanti, RP Farisi, F A Setyawan, S R Sulistyanti, Y Yuniati, U Murdika

Dept. Electrical Engineering, Universitas Lampung, Bandar Lampung, Indonesia

E-mail: sri.purwiyanti@eng.unila.ac.id

Abstract. Traffic jam caused many problems, one of which, causing time wasting and increasing fuel consumption of user's vehicle. One of the solutions offered to solve traffic jam is a control system named traffic light timer. It is a traffic management system in each spot of the road done by minimizing the waiting time of the rider during the jam. This research built traffic light timer controller model based on image processing by using Background Subtraction method. The object captured on the road will be processed using Background Subtraction method, the results then show the situation of the road as the parameters of quiet, medium, and heavy roads. The images are then forwarded to the traffic light timer control system to be processed using the hardware and software used in the research. The results show that the Traffic light timer control system is able to classify the heaviness of vehicles queue, to control the timer, and the lights of the traffic light automatically based on the queue of the vehicles with time processing around 2 seconds.

1. Introduction

One of problem in modern cities is a traffic jam. In fact, traffic jam causes several problems, like to increase fuel consumption and time wasting. To overcome the problem, one of solution is to build a system that could control a timer of the traffic light, so it can minimizing the waiting time of the rider during the jam.

Some publications have been report researches about smart system of timer traffic jam by using different method [1-7]. However the system was not optimal yet. One of method used is image processing method [8] by using edge detection method. However, the result shows that the system needs delay time about 4 s. In this research, we used image processing with different method, i.e. Background Subtraction method, to build the controlling system of traffic light timer.

2. Methods

The Image processing is a general term used to manipulate and modify images using various methods [9]. Generally, image processing is a process of improving image quality so that it is more easily interpreted by humans or computers [10].

The color image consists of 3 matrix layers, i.e. R-layer, G-layer and B-layer. The initial process in image processing is to change the color image into a grayscale, it is used to simplify the image model. The 3 layers are converted to 1 layer of grayscale matrix and the result is grayscale image. In this image there are no more colors anymore except a degree of grayness [2]. Then the Background Subtraction process is carried out, that is the process for finding objects in the image by comparing the existing images with a background model. The purpose of Background Subtraction is to know or distinguish the background and objects in an image [11].

In this research, capturing image has been done by using one camera installed on each of road. In this model only two road segments is used because it has presented a system for the type of traffic light at a crossroads. Diagram block of the system is seen in Figure 1. The image of the object will be captured by the camera. Raspberry Pi 3 will process the information and then adjust the length of time the green light based on the number of cars caught by the camera. Then, display will show the length of time that is generated. The control program is created by using Library Open CV in Python Programming Languages.

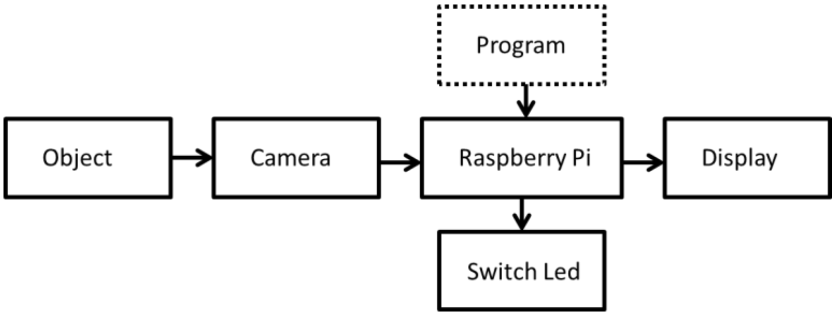

Figure 1. Diagram Block of the system

After the image is taken by using camera automatically, then the image is repaired in order to produce the desired image quality, for example by increasing brightness. The next step is to determine Background. The background model used is the image of the road in empty condition. Next is an example of the process of calculating the background image to be grayscale using a 10x5 pixel image sample with a matrix as shown in Figure 2.

$$\begin{bmatrix} 50 & 21 & 5 & 0 & 0 & 4 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 & 2 & 1 & 0 & 2 & 2 & 0 \\ 2 & 5 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 2 \\ 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 78 & 80 & 61 & 61 & 25 & 77 & 50 & 72 & 65 & 76 \end{bmatrix}$$

(a)

$$\begin{bmatrix} 54 & 24 & 6 & 0 & 1 & 5 & 2 & 1 & 3 & 1 \\ 2 & 2 & 0 & 4 & 2 & 2 & 1 & 3 & 3 & 1 \\ 4 & 7 & 0 & 0 & 1 & 0 & 0 & 2 & 2 & 3 \\ 2 & 1 & 3 & 0 & 1 & 1 & 2 & 2 & 2 & 4 \\ 86 & 87 & 64 & 28 & 80 & 78 & 54 & 77 & 69 & 80 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 39 & 15 & 8 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 9 & 9 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 62 & 69 & 57 & 21 & 61 & 53 & 29 & 47 & 36 & 45 \end{bmatrix}$$

(c)

Figure 2. Matrix composition for a) Red, b) Green, and c) Blue Backgrounds

To change the image into grayscale, the following calculation is done:

$$Y(0,0) = (0,299*50) + (0,587*54) + (0,114*39) = 51,094 \sim 51 \quad (1)$$
$$Y(1,0) = (0,299*21) + (0,587*25) + (0,114*15) = 21,664 \sim 21 \quad (2)$$
$$Y(2,0) = (0,299*5) + (0,587*6) + (0,114*8) = 7,664 \sim 8 \quad (3)$$
$$Y(3,0) = (0,299*0) + (0,587*0) + (0,114*4) = 0,456 \sim 0 \quad (4)$$

The process is carried out on all pixels. After calculating the conversion from RGB to grayscale, the grayscale image is obtained in the background. Then the analysis is done to get the best threshold value to be use. In this study, the selected threshold value is T = 50. Pixels whose intensity values are below 50 are then changed into black (intensity value = 0), while pixels whose intensity values above 50 are converted to white (intensity value = 1). The initial image, the image in grayscale, and the binary image are shown in Figure 3.

Then, the foreground extraction process is carried out from the background. The process of determining the foreground is to reduce the current image with the background image. In simple terms this is done with the following equation:

$$R_{((r,c))} = I_{((r,c))} - B_{((r,c))} \tag{5}$$

The value of R is then compared to the predetermined threshold value, if it is greater than the threshold value then the pixels in I (r, c) can be considered different from the pixels in B (r, c) [7]. If the result of the reduction is 1, then the pixel 1 will be formed in the foreground, which it is mean white color. Calculation is done for all pixel values in the matrix.

(a) (b) (c)

Figure 3. Background images: a) original, b) grayscale, and c) after binaryzation

3. Results and Discussions

After the program is successfully formed, an experiment is carried out using a traffic light system model for three road conditions, i.e. quiet, medium, and heavy.

To determine the quiet parameter, the number of pixels in the white image is calculated (value = 1) in the foreground result of the image. In this process, images with 3 vehicles are used [Figure 4(a)], which in this case is the upper limit of the estimated number of vehicles in a quiet road condition. The process found that WHITE PIXEL = 81,899 pixels of TOTAL PIXEL = 921,600 pixels. So, the empty parameter can be determined ≤ 90,000 pixels.

(a) (b) (c)

Figure 4. Foreground images under: a) quiet, b) moderate, and c) heavy road conditions.

Experiments to find out the parameters have been done by using 6 miniature vehicles [Figure 4(b)], which in this case is the upper limit of the estimated number of vehicles in a state of moderate road. Based on the number of white pixels, it can be determined that the empty parameter is 90,000 pixels - 140,000 pixels. The same procedure has been done to determine heavy parameters, which is by using 9 miniature vehicles [Figure 4 (c)]. Thus it can be determined that the heavy parameter is > 140,000 pixels. Hardware model are shown in Figure 5.

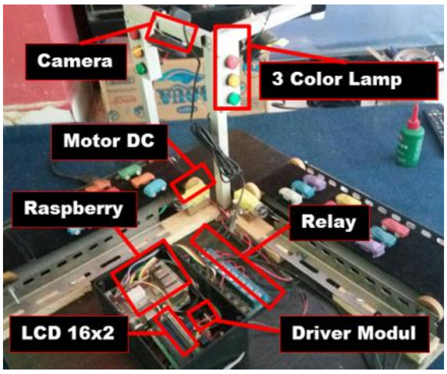

Figure 5. Hardware model

The next experiment is to test how long the light traffic lights up according to the number of vehicles in the queue. If the number of white pixels is less than 90000 pixels, (quiet condition) the results of the program are counter down from 10 to 0, so that it is displayed on the LCD for 10 seconds. The counter calculates the duration of the green light, which is 10 s. The green light only lights up for 10 s because the maximum number of waiting vehicles is only 3. The more the number of vehicles waiting in line, the longer the green light will light up [Figure 6].

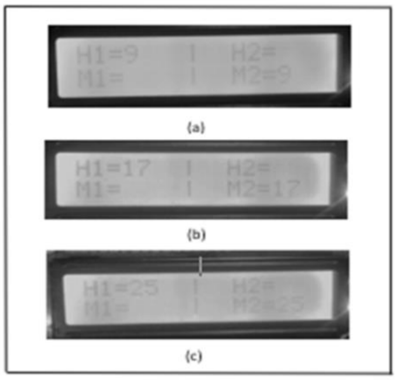

Figure 6. Traffic light timer display under a) quiet, b) moderate, and c) heavy road conditions

The next experiment is to find out the time needed to process the data, from the program running until the model produces output. The time calculation has been done using a stopwatch. The experimental results are shown in Table 1. It can be seen that the number of vehicles affects the time taken by the processor to process the image into an output. Even so, this process only requires a maximum of 2 seconds to process under heavy road condition.

Table 1. Process time depends on road condition

No.	Road Condition	Process Time
1.	Quiet	00:01.78
2.	Quiet	00:01.81
3.	Quiet	00:01.85
4.	Moderate	00:01:88
5.	Moderate	00:01:88
6.	Moderate	00:01:93
7.	Heavy	00:01:97
8.	Heavy	00:01:99
9.	Heavy	00:02:01

4. Conclusions

Background subtraction model has been used to build traffic light timer control system based on image processing method. The system able to classify the heaviness of vehicles queue, to control traffic light timer automatically based on the queue of the vehicles with time processing around 2 seconds. This result may contribute in order to build a smart traffic light system.

Acknowledgments

We thanks to Lampung University for funding support through Dipa FT Unila Program.

References

[1] Sudarono 2011 *Pengendalian Blok Sistem Traffic Light dengan Pengaturan Rush Hour yang dapat Diatur Secara Terjadwal* Universitas Muhammadiyah Surakarta
[2] Khoswanto, H. Lim, and R. Liong 2005 *Traffic Light Controller Menggunakan Media RF* Universitas Kristen Petra
[3] Manto 2011 *Perangkat Pengatur Timer Lampu Lalu Lintas Berdasarkan Antrian Kendaraan* Universitas Indonesia
[4] A. Bhalearo 2015 IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) **10** 53
[5] N. Bhasi and Chithra 2017 International Research Journal of Engineering and Technology (IRJET) **4** 321
[6] L. Singh, S. Tripathi, H. Arora 2009 International Journal of Recent Trends in Engineering **2** 2
[7] V. Parthasarathi, M. Surya, B. Akshay, K. M. Siva and S. K. Vasudevan 2015 Indian Journal of Science and Technology **8**(16) 1
[8] Sonia, et al 2015 *Smart Traffic Light using Image Processing and Fuzzy Logic Method* Bachelor Telkom University
[9] N. Efford 2000 *Digital Image Processing: A Practical Introduction Using Java* University of Michigan: Addison-Wesley
[10] Poerwadarminta W.J.S 1990 *Kamus Besar Bahasa Indonesia* Balai Pustaka, Jakarta p.1158.
[11] M. Nixon, S. Aguado, and Alberto S 2002 *Feature Extraction and Image Processing* Linacre House, Jordan Hill, Oxford