DEVELOPMENT OF BACKGROUND SUBTRACTION ALGORITHM FOR A NON-INTRUSIVE TRAVEL TIME ESTIMATION

by

ISZAIDY ISMAIL
(1540211646)

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

School of Computer and Communication Engineering
UNIVERSITI MALAYSIA PERLIS

2019

ACKNOWLEDGMENT

"*You simply will not be the same person two months from now after consciously giving thanks each day for the abundance that exists in your life. And you will have set in motion an ancient spiritual law: the more you have and are grateful for, the more will be given you*."

– Sarah Ban Breathnach

Alhamdulillah, all praises and thanks to Allah S.W.T.

There are many individuals that have supported and guided me towards the successful completion of this PhD research work and therefore deserve my utmost thanks.

First of all, my sincere gratitude goes to my beloved parents, Ismail Hasim and Fatimah Md Isa, for giving me life in the first place and for always encouraging me to pursue success in this life. No words can describe my appreciation for my beloved wife, Syafinar Ramli, for her unconditional support and encouragement in my pursuit of a PhD.

Special thanks also goes to my helpful supervisors, Dr. Ruzelita Ngadiran and Prof. Ir. Ts. Dr. R Badlishah Ahmad. Without their supervision and support, I would not be where I am now. The guidance and advice you have given are indeed much appreciated, and I am deeply honoured to have both of you as my supervisors.

I would like to acknowledge the Ministry of Higher Education for funding via *Skim Latihan Akademik Bumiputra* (SLAB). In addition, I would like to thank Universiti Malaysia Perlis (UniMAP) for providing the facilities and equipment I have needed to produce and complete my thesis.

Lastly, I would like to express my gratitude to all my colleagues and friends, especially to my embedded research group members (Dr Ilman, Dr Herwansyah, Dr Amir Nazreen, Wafi and Badrulhisham) for the stimulating discussions, working together for deadlines and for all the fun we have had in last few years along this journey.

TABLE OF CONTENTS

	PAGE
DECLARATION OF THESIS	**i**
ACKNOWLEDGMENT	**ii**
TABLE OF CONTENTS	**iii**
LIST OF TABLES	**vi**
LIST OF FIGURES	**ix**
LIST OF ABBREVIATIONS	**xiii**
LIST OF SYMBOLS	**xv**
ABSTRAK	**xvi**
ABSTRACT	**xvii**
CHAPTER 1 : INTRODUCTION	**1**
1.1 Overview	1
1.2 Problem Statement	5
1.3 Research Objective	6
1.4 Research Motivation	7
1.5 Research Scope	7
1.6 Thesis Organization	8
CHAPTER 2 : LITERATURE REVIEW	**9**
2.1 Introduction	9
2.2 Travel Time	9
2.3 Motion Detection	17

2.3.1 Background Subtraction 18

2.3.2 Taxonomies of BGS 19

2.3.3 Relevant BGS Approaches 21

2.3.4 Preliminary Study of the BGS Algorithm 26

2.3.5 BGS Measurement Technique 29

2.4 Embedded Platform 34

2.4.1 Background Study on Embedded Vision System 35

2.4.2 Hardware Evaluation of Embedded System 40

2.5 Embedded Software 41

2.6 Summary 43

CHAPTER 3 : METHODOLOGY 44

3.1 Introduction 44

3.2 Software Design and Development 45

3.3 Background Subtraction 48

3.3.1 Proposed BGS 1 51

3.3.2 Proposed BGS 2 55

3.3.3 Embedded BGS 59

3.3.4 Post-processing and Pre-processing of BGS 61

3.4 Implementation of Travel Time Estimation 62

3.5 Hardware Description 65

3.5.1 Raspberry Pi Description 67

3.5.2 Hardware Performance Measuring Tools 69

3.6 Summary 70

CHAPTER 4 : RESULTS & DISCUSSION 71

4.1 Introduction 71

4.2 Dataset Videos 72

4.3 Quality Assessment of the BGS Algorithm 75

4.4 Learning Phase 76

4.4.1 Parameter Value Determination for Proposed BGS 1 76

4.4.2 Parameter Value Determination for Proposed BGS 2 84

4.5 Evaluation Phase 95

4.5.1 Synthetic Videos 96

4.5.2 Real Videos 104

4.6 Influence of Post-process & Pre-process 112

4.7 Influence of Resolution 114

4.8 Implementation on Embedded Platform 116

4.9 Travel Time Estimation 117

4.10 Summary 120

CHAPTER 5 : CONCLUSION 121

5.1 Conclusion 121

5.2 Future Works 123

5.2.1 Exploring other Embedded board 123

5.2.2 Enhancing Software Design 124

REFERENCES 125

APPENDIX A LIST OF PUBLICATION 139

LIST OF TABLES

		PAGE
Table 2.1	Advantage and disadvantage of travel time estimation device	17
Table 2.2	Summary of BGS algorithms	26
Table 2.3	Summary of BGS performances result	27
Table 2.4	Comparison performances of different BGS methods	28
Table 2.5	Description of TP, TN, FP and FN	30
Table 2.6	Embedded board and respective specifications	40
Table 3.1	Parameters for proposed BGS 2	56
Table 3.2	Technical features of the Raspberry Pi 2 Model B	69
Table 4.1	Combination number for synthetic data	72
Table 4.2	The average value for minimum background counter map determination	80
Table 4.3	The average value for threshold determination	82
Table 4.4	Summary of proposed BGS 1 parameters	83
Table 4.5	The average score of minimum background counter map for proposed BGS 2	87
Table 4.6	The average score for history threshold determination	90
Table 4.7	F-measure score for maximum background counter map	91
Table 4.8	SSIM score for maximum background counter map	91
Table 4.9	D-score result for maximum background counter map	92

Table 4.10	FSD score for maximum background counter map	92
Table 4.11	Average score for maximum background counter map	93
Table 4.12	Summary of proposed BGS 2 parameters	93
Table 4.13	Average and improvement quality between proposed BGS 1 and 2	94
Table 4.14:	Parameter setting for each BGS algorithm	95
Table 4.15	Segmentation result of datasest 112	96
Table 4.16	Segmentation result of dataset 212	99
Table 4.17	Segmentation result of dataset 312	99
Table 4.18	Segmentation result of dataset 412	101
Table 4.19	Segmentation result of dataset 512	103
Table 4.20	Average segmentaion result for all synthetic dataset	104
Table 4.21	Static quality for *baseline* dataset	105
Table 4.22	Overall segmentation quality for *baseline* dataset	105
Table 4.23	Static quality for *night video* dataset	107
Table 4.24	Overall segmentation quality for *night video* dataset	107
Table 4.25	Static quality for *low frame rate* dataset	109
Table 4.26	Overall segmentation quality of *low frame rate* dataset	109
Table 4.27	Static quality for *camera jitter* dataset	111
Table 4.28	Overall segmentation quality for *camera jitter* dataset	111
Table 4.29:	Average overall result of all event for real video	112
Table 4.30	Evaluation of static quality for median filter application	113

Table 4.31 Overall evaluation segmentation quality for median filter application 113

Table 4.32 Hardware performance on different resolution 115

Table 4.33 Overall segmentation evaluation of different resolution 115

Table 4.34 Performance comparison on *baseline* dataset 116

Table 4.35 Performance comparison on *low frame rate* dataset 117

Table 4.36 Hardware performance and travel time estimation result 119

Table 4.37 Hardware performance and travel time estimation result 120

LIST OF FIGURES

		PAGE
Figure 2.1	Available devices used to estimate travel time	12
Figure 2.2	Frame difference method	23
Figure 2.3	Flow of image processes on counting TP, TN, FP and FN	30
Figure 2.4	Diagram of SSIM system	32
Figure 3.1	Research workflow	45
Figure 3.2	Software design and development workflow	46
Figure 3.3	Software and hardware architecture	48
Figure 3.4	Flowchart of the basic BGS	50
Figure 3.5	Position of background counter map	52
Figure 3.6	Background counter map illustration	54
Figure 3.7	Simple flowchart of proposed BGS 2	57
Figure 3.8	Proposed BGS 2 flowchart	58
Figure 3.9	Parallel implementation on proposed BGS	60
Figure 3.10	Blob tracking flowchart	64
Figure 3.11	Developed system hardware	67
Figure 3.12	Raspberry Pi 2 interfaces and connectors	68
Figure 4.1	Street scene from synthetic dataset with its ground truth	73
Figure 4.2	F-measure value for minimum background counter map determination	78

Figure 4.3	SSIM value for minimum background counter map determination	78
Figure 4.4	D-score value for minimum background counter map determination	79
Figure 4.5	FSD value for minimum background counter map determination	79
Figure 4.6:	F-measure score for threshold determination	80
Figure 4.7:	SSIM score for threshold determination	81
Figure 4.8:	D-score result for threshold determination	81
Figure 4.9:	FSD score for threshold determination	82
Figure 4.10	Snapshot of 111 dataset at 930th frame with τ = 20 : (a) $f_{\text{px-min}}$ = 1; (b) $f_{\text{px-min}}$ = 5; (c) $f_{\text{px-min}}$ = 10	83
Figure 4.11	F-measure score for minimum background counter map determination	85
Figure 4.12	SSIM score for minimum background counter map determination	86
Figure 4.13	D-score result for minimum background counter map determination	86
Figure 4.14	FSD score for minimum background counter map determination	87
Figure 4.15	F-measure score for history threshold determination	88
Figure 4.16	SSIM score for history threshold determination	88
Figure 4.17	D-score for history threshold determination	89
Figure 4.18	FSD score for history threshold determination	89

Figure 4.19 Snapshot of output from proposed BGS 1 (left) and proposed BGS 2 (right) algorithms at 670th frame for: (a) Dataset 111; (b) Dataset 211; (c) Dataset 311; (d) Dataset 411; and (e) Dataset 511 94

Figure 4.20 Snapshot of dataset 112 at 195th frame: (a) Input image; (b) Ground truth; (c) Output by Frame Difference; (d) Output by ABGL; (e) Output by MOG2; (f) Output by PBAS; (g) Output by Codebook; (h) Output by proposed BGS 2; and (i) Output by proposed BGS 1. 97

Figure 4.21 Snapshot of dataset 212 at 300th frame: (a) Input image; (b) Ground truth; (c) Output by Frame Difference; (d) Output by ABGL; (e) Output by MOG2; (f) Output by PBAS; (g) Output by Codebook; (h) Output by proposed BGS 2; and (i) Output by proposed BGS 1. 98

Figure 4.22 Snapshot of dataset 312 at 555th frame: (a) Input image; (b) Ground truth; (c) Output by Frame Difference; (d) Output by ABGL; (e) Output by MOG2; (f) Output by PBAS; (g) Output by Codebook; (h) Output by proposed BGS 2; and (i) Output by proposed BGS 1. 100

Figure 4.23 Snapshot of dataset 412 at 1170th frame: (a) Input image; (b) Ground truth; (c) Output by Frame Difference; (d) Output by ABGL; (e) Output by MOG2; (f) Output by PBAS; (g) Output by Codebook; (h) Output by proposed BGS 2; and (i) Output by proposed BGS 1. 102

Figure 4.24 Snapshot of dataset 512 at 1185th frame: (a) Input image; (b) Ground truth; (c) Output by Frame Difference; (d) Output by ABGL; (e) Output by MOG2; (f) Output by PBAS; (g) Output by Codebook; (h) Output by proposed BGS 2; and (i) Output by proposed BGS 1. 103

Figure 4.25	Snapshot of *baseline* dataset at 790th frame: (a) Input image; (b) Ground truth; (c) Output by Frame Difference; (d) Output by ABGL; (e) Output by MOG2; (f) Output by PBAS; (g) Output by Codebook; (h) Output by proposed BGS 2; and (i) Output by proposed BGS 1.	106
Figure 4.26	Snapshot of *night video* dataset at 440th frame: (a) Input image; (b) Ground truth; (c) Output by Frame Difference; (d) Output by ABGL; (e) Output by MOG2; (f) Output by PBAS; (g) Output by Codebook; (h) Output by proposed BGS 2; and (i) Output by proposed BGS 1.	108
Figure 4.27	Snapshot of *low frame rate* video at 900th frame: (a) Input image; (b) Ground truth; (c) Output by Frame Difference; (d) Output by ABGL; (e) Output by MOG2; (f) Output by PBAS; (g) Output by Codebook; (h) Output by proposed BGS 2; and (i) Output by proposed BGS 1.	109
Figure 4.28	Snapshot of *camera jitter* dataset at 1000th frame: (a) Input image; (b) Ground truth; (c) Output by Frame Difference; (d) Output by ABGL; (e) Output by MOG2; (f) Output by PBAS; (g) Output by Codebook; (h) Output by proposed BGS 2; and (i) Output by proposed BGS 1.	110
Figure 4.29	Parameter setting for proposed BGS 2	118
Figure 4.30	Parameters setting for blob tracking	118
Figure 4.31	Output by travel time estimation at 63rd frame; (a) input video, (b) segmented output and (c) output with bounding box	119
Figure 4.32	Output by travel time estimation at 38th frame; (a) input video, (b) segmented output and (c) output with bounding box	119

LIST OF ABBREVIATIONS

3D	Three-dimensions
ABGL	Adaptive Background Learning
ARM	Advanced RISC Machine
ATMS	Advance Traffic Information System
AVDS	Automatic Vehicle Detection System
BGS	Background Subtraction
BMC	Background Modelling Challenge
BSD	Berkeley Software Distribution
CCS	Central Computer Software
CCTV	Closed-circuit Television
CPU	Central Processing Unit
DSP	Digital Signal Processing
ETC	Electronic Toll Collection
FOSS	Free Open Source Software
FOV	Field of View
FPGA	Field-programmable Gate Array
FPS	Frames-per-second
GMM	Gaussians Mixed Model
GPS	Global Positioning System
HD	High Definition
ITIS	Integrated Transportation Information System
ITS	Intelligent Transport System
KDE	Kernel Density Estimation
LIDAR	Light Detection and Ranging
LCD	Liquid Crystal Display
MOG	Mixture Of Gaussians
OD	Origin Destination
OS	Operating System
PACDS	Parallel Architecture Core Digital Signal
PBAS	Pixel Based Adaptive Segmenter
PC	Personal Computer
RADAR	Radio Detection and Ranging
RAM	Random Access Memory

RF	Radio Frequency
RISC	Reduced Instruction Set Computer
SiVIC	Simulator of Vehicle, Infrastructure, and Sensors
SoC	System On Chip
USB	Universal Serial Bus
VGA	Video Graphics Array
VMS	Variable Message Sign
WSN	Wireless Sensor Network

LIST OF SYMBOLS

X_t	Foreground(motion) mask
$I_t(x,y)$	Intensity value of pixel(x,y) at time t
f_{px}	Background Counter Map
τ	Threshold
$\tau_{history}$	History Threshold
m	Row
n	Coloum
x	Pixel position on x-axis
y	Pixel position on y-axis
N	Frame sequences
t	Frame number
d	Intensity distance
s	Second (time)
S	Speed
D	Distance

Pembangunan Algoritma Penolakan Latar Belakang Untuk Anggaran Masa Perjalanan Tanpa Gangguan Luar

ABSTRAK

Pada masa kini, sistem terbenam telah menjadi sebahagian dalam kehidupan seharian kita. Kemajuan dalam pengimejan komputer dan pembelajaran mesin, termasuklah dengan peningkatan kuasa pengkomputeran, memungkinkan sistem ini melaksanakan tugas-tugas lain seperti pemprosesan imej dan analisis video; dan mewujudkan sistem pengimejan terbenam. Walau bagaimanapun, bebanan komputer adalah cabaran terbesar dalam algoritma penglihatan komputer kerana kerumitan dan susahnya kepada pemprosesan seni bina komputer terbenam. Oleh itu, prestasi pelaksanaan algorithma pemprosesan perlu dianalisis. Algoritma penolakan latar belakang yang asas didapati mempunyai kompleksiti rendah tetapi ia boleh menjejaskan kualiti segmentasi dari penolakan latar belakang. Dalam tesis ini, teknik berasaskan pemprosesan pengimejan komputer pintar yang baru dicadangkan untuk mengesan gerakan (kenderaan) untuk sistem pengiraan masa perjalanan menggunakan sistem terbenam. Bersesuaian dengan ini, penolakan latar belakang yang ringan, cepat dan cekap diperlukan untuk sistem ini untuk mengesan objek dari rantaian imej. Kaedah yang baru dicadangkan dengan memperkenalkan peta kaunter latar belakang untuk membuat keputusan tentang penentuan latar belakang. Dua versi algoritma yang berbeza, iaitu cadangan *BGS* 1 dan 2, telah dibangunkan untuk tujuan berlainan (latar belakang yang bising dan latar belakang yang pelbagai) dan dinilai menggunakan kaedah penilaian kualiti and kuantitatif berdasarkan parameter yang berkaitan. Selain itu, algoritma juga dinilai berdasarkan prestasi pemprosesan pusat (*CPU*) dan memori dan juga prestasi masa perlaksanaan sebelum keputusan diambil untuk digunakan dan diterapkan pada sistem anggaran masa perjalanan berasaskan sistem terbenam. Dua algoritma tersebut dinilai menggunakan PC untuk menentukan nilai parameter terbaik untuk aplikasi algoritma. Penemuan membuktikan bahawa kaedah yang dicadangkan dapat memberi keputsan yang baik dari segi kualiti segmentasi. Prestasi cadangan *BGS* 2 adalah lebih baik untuk setiap ukuran (*F-measure*- peningkatan sebanyak 31.67%, *SSIM*- peningkatan sebanyak 11.84%, *D-score*- 9.78% kurang ralat dan *FSD* - peningkatan sebanyak 11.02%) berbanding cadangan *BGS* 1. Berdasarkan nilai ini, kedua-dua algoritma ini digunakan pada *Raspberry Pi* dan ia menggunakan 25%-26% *CPU* and 4.5%-4.7% memori. Kedua-dua algoritma masih mengekalkan skor kualiti segmentasi yang sama seperti dijalankan di atas *PC* dan memberikan skor kualiti segmentasi yang sama berbanding *PBAS* and *MOG2* dengan menghadkan bagi set data yang disediakan. Kedua-duanya melaksanakan proses dengan cepat seperti algoritma penolakan biasa. Cadangan *BGS* ini telah digunapakai untuk anggaran masa perjalanan dan keputusan ia dianalisis. Anggaran masa perjalanan berfungsi dengan baik dan ia menggunakan 41%-56% daripada *CPU* dengan 4.5%-5.5% daripada memori.

Development of Background Subtraction Algorithm for a Non-intrusive Travel Time Estimation

ABSTRACT

In today's world, embedded systems have become an integral part of our daily lives. The advancements in computer vision and machine learning, in conjunction with the abundance of computing power, have made it possible for these systems to perform other tasks such as image processing and video analytics; giving rise to embedded vision systems. However, the computer vision algorithms cannot always address the computational costs required due to the complexity and weight of the embedded computer processing architecture. Therefore, the execution performance of the processing algorithms should be investigated further. The basic background subtraction algorithm is found to have low complexity, but it may affect the segmentation quality of the foreground result. In this thesis, a new intelligent computer vision processing-based technique is proposed to detect motion (vehicle) in an embedded travel time estimation. Initially, a light, fast, and efficient background subtraction is needed for this system to extract the subject from frame sequences. A new frame difference-based method was proposed with a background counter map to conduct background determination decisions. Two different versions of the algorithm, which are proposed BGS 1 and 2, were developed for different purpose (noisy and multimodal background) and evaluated using a quantitative segmentation quality assessment method based on relevant parameters. Moreover, the algorithms were also analyze the CPU and memory performance and execution time before the final result was used and applied on the embedded-based travel time estimation. Two algorithms were evaluated using a PC to determine the best parameter value for the algorithm application. The findings proved that the proposed methods performed excellently in terms of segmentation quality. Proposed BGS 2 performed better for every measurement (F-measured—31.67% improvement, SSIM—11.84% improvement, 9.78 % less error, and FSD—11.02 % improvement) compared to proposed BGS 1. Based on this value, both algorithm is implement on the Raspberry Pi and it used 25%-26% of CPU and 4.5%-4.7% of memory. But both algorithm maintained the segmentation quality score as on PC and provided a similar segmentation quality score compare to PBAS and MOG2 with limitation to the provided dataset. Both also execute fast as the frame difference algorithm. Based on these results, proposed BGS was implemented for travel time estimation and analyzed. Travel time estimation worked well by consumed 41%-56% of CPU with 4.5% - 5.5% of memory.

CHAPTER 1 : INTRODUCTION

1.1 Overview

Nowadays, advanced technology such as modern transportation systems is needed in the areas of traffic controls system and detection, communication, and information provision. In order to determine the transportation needs and appropriate solution for an area, it is important to have understanding of the underlying characteristic of travel. For transportation systems, innovative system are required to take into account the whole phenomenon of the transport domain (Zhou, Lu, & Zhang, 2012).

The origins and destination of traffic are among the most important of travel characteristics. An origin-destination (OD) study is a review of travel information representing the number of trips between origin and destinations within a specific time interval (Bauer et al., 2018; X. Li et al., 2017). This information is particularly useful for helping and assisting traffic planning authorities to understand travel pattern in the study area. By understanding the traffic movements across a wide area, the collected data from the analysis can be used in future transportation planning and evaluation. Government transportation agencies often conduct OD surveys to identify and understand the travel patterns in a certain region. These surveys use the data to identify the popularity of origin and destination pairs. From the results, decision-makers can determine the region that needs new facilities, or locations that need investment in public transport infrastructure. OD data are regularly used as input into local systems (Blogg, Semler, Hingorani, & Troutbeck, 2010).

Travel time is one of the vital information in an OD study (Fan et al., 2018; Lu, Li, Zheng, & Wang, 2018). Advanced travellers make decisions based on real-time measurements to help plan their journeys. Particularly, providing travel time information at particular subset of trips (from origin to destination) to road system users will allow them to create more informed decisions about their choices, such as the start time of their trips, journey routes, and transportation mode. The information can be presented to travellers via variable message signs, on-board navigation devices, cellular phones, Internet websites, and more. In addition, road system authorities are in much need for historical records of daily patterns of travel times and traffic speeds to assist in the decision-making process for investments in transportation infrastructure for the whole road network (Bar-Gera, 2007). Thus, travel time information is turning into a very important data for various applications in the transportation system.

In this context, video surveillance systems provide the basic functionality needed to transform the security paradigm from investigation to estimation. The video surveillance system can be used to collect important traffic information data for both individual travellers and road system authorities. Travel time information is more useful when travellers need to choose several paths from their origin to their destination. This information would be more beneficial if future travel time were estimated as early as possible. Therefore, researchers have developed several methods and techniques to measure and estimate travel times based on different devices. Travel time estimation is depending on traffic flow, traffic incident, vehicle speed, and weather conditions and it is a complex task (L. Huang & Barth, 2008; Tak, Kim, & Yeo, 2014).

Enormous change has arisen in the world of embedded systems pushed by the advancement in integrated circuit technology and the availability of open source resources. This has given an advance to new challenges in developing an advanced embedded system. This scenario is further evidenced from the appearance of sophisticated new products such as smartphones and the continual increase in the amount of resources that can be packed into a small form factor, which require significant high-end skills and knowledge. More people are gearing up to acquire more skills and knowledge to keep abreast of the technologies to build advanced embedded systems that serve numerous purposes (Che Rosli, 2010).

Recent technological advances have enabled the development of a new generation of smart cameras, which represent a quantum leap in sophistication. While today's digital cameras capture images, smart cameras capture high-level descriptions of a scene and analyse what they see. These devices could support a wide variety of applications including human and animal detection, surveillance, motion analysis, and facial identification. Efficiency in size, cost, weight, interchangeability, and consistency are the primary factors that determine whether an embedded board can be selected as the hardware platform for a system. Utilising a Linux-based embedded board allows us to handle the availability of open source resources such as kernels, libraries, and drivers in developing and implementing this system (Ahmed et al., 2008).

Different travel time's estimation have been established and operated over the years. Hereby has been shown that the image sensor is one of the best choices to be applied and used for a travel time estimation due to its non-intrusiveness and capability to capture global and specific vehicle behaviour data. The image sensor can provide a wealth of traffic information such as incident detection and so on. As the embedded board becomes cheaper and digital image sensor are gradually able to produce images with excellent quality, video-based systems can turn out to be a lower cost alternative for non-intrusive travel time and speed measurement (Luvizon, Nassu, & Minetto, 2014).

The hardware components of the embedded vision system consist of a image sensor and an embedded board (Guo, Zhou, Zhou, Kimura, & Goto, 2018; Mhalla, Chateau, Gazzah, Essoukri, & Amara, 2018; Perri, Frustaci, Spagnolo, & Corsonello, 2018). An embedded vision system is a combination of an image sensor with a surveillance system in an embedded board to create the travel time measuring tool. The software design considers the start of the process from image grabbing up to vehicle tracking. Through this process, the average travel time of the vehicles that pass through the road can be measured. This data will be very useful for road users to choose the right and fastest trip.

Identifying moving objects such as vehicles using an embedded vision system is a fundamental and critical task, especially when it comes to traffic monitoring and analysis. Background subtraction algorithms are commonly used to detect and track moving objects. First, each frame that the image sensor grabs is compared against the background frame. The differences between the two frames are considered to be the moving object. Then, the differences are further processed for object localization and tracking purposes (Cheung & Kamath, 2004). Although many background subtraction algorithms have been used as the first step in many computer visions processing, the problem of identifying objects in a complex environment with limited computer resources such as the embedded board is still far from being completely solved.

1.2 Problem Statement

Following the introduction section, the goal of this work is to create a robust and persistent background subtraction method for travel time estimation. Object segmentation is the initial and crucial step for successful estimation. Therefore, this step must be robust against noise and environmental changes.

State of art camera based travel time estimation usually based on video data of the visual domain. A persistent monitoring of the scene however often desired. When aiming for a persistent twenty-four hours measuring, a fixed fusion based on image characteristic might be challenging, since the monitored situation can be change dramatically by the visual range of light, such as the illuminating problem, camera jitter, low frame rate video, and overlapping problem (Kalsotra & Arora, 2017; Sofwan et al., 2018). For example, on a rainy day, the environment will become dark, so vehicles may be tough to track and measure the travel time. This problem create a hard process to be solve.

Various background subtraction algorithms were proposed in recent years for detecting moving objects. As it is well known, the background subtraction (BGS) separates foreground (object) in the observed scene from the background (static object that remains quiescent for a long time). In order for a BGS to be useful in embedded system, it must be both accurate and computationally efficient because it has limited computational resources. The recent background subtraction algorithms are accurate because it used complex model which required a high computational resources (Shen et al., 2016).

Few projects estimate travel time based on embedded boards. Most existing systems have been based on high-specification central processing unit (CPU) with a large processing board (Sofwan et al., 2018). The use of PCs is restricted to a few fields, since PCs obviously limit portability due to high power consumption, size, and weight. Actually portable solutions based on embedded board are presented in (Abaya, Basa, Sy, Abad, & Dadios, 2014; Baby & Ahamed, 2014; Soetedjo, Ashari, Mahmudi, & Nakhoda, 2014; Ujjainiya & Chakravarthi, 2015). Travel time estimation should consider the embedded board because it is small in size and is an efficient low-cost hardware platform, which is able to acquire and conduct video processing without any external processing unit (Cocorullo et al., 2015).

1.3 Research Objective

The objectives of this research are:

i. To design a new background subtraction for an embedded platform.

ii. To evaluate a quantitative image segmentation quality based on relevant parameters.

iii. To analyse the CPU, memory performance and execution time in comparison to others BGS algorithm in an embedded-based environment to determine the feasibility of the system.

1.4 Research Motivation

The primary motivation for this work is to design a new BGS algorithm with good segmentation quality and implement it on an embedded board, which has limited computer resources. Profitability of size, weight, and cost are the primary roots leading to the selection of the embedded board as the hardware platform for the system. Analysing the performance of the background subtraction algorithm on the embedded board allows us to take advantage of the availability of an image-processing algorithm for developing and implementing the system. Additionally, the others motivation of this work is to develop, implement, and analyse a travel time estimation composed of a low-cost embedded board and proposed BGS algorithm.

1.5 Research Scope

This research focuses on the development, implementation, and analysis of a background subtraction algorithm for a non-intrusive travel time estimation using an embedded board, Raspberry Pi. The overall system development includes software development and implementation with hardware integration. An embedded board with a GNU/Linux-based Operating System (OS), which utilises open source technology kernels, libraries and compilers are used to provide an alternative low-cost solution for the hardware of the system.

On the software development, designing the algorithm and enabling the background subtraction algorithm to work with travel time estimation on embedded boards are involved. To yield the best performance of the proposed BGS, it has been compared with other BGS algorithm by using multiple synthetic videos and real videos database. Then, the proposed BGS is implemented on travel time estimation, which demonstrates the feasibility of proposed BGS to operate in real-time scenario on embedded board.

1.6 Thesis Organization

This thesis composed of five (5) chapters and chapter one (1) presents an overview and problem statements on the issues, motivating aspects, objectives, scope and thesis layout. While chapter 2 presents the literature review on travel time, BGS algorithm and its application on embedded platform. Chapter 3 describes the hardware components of the travel time estimation such as Raspberry Pi board. The software development process, which involves software design and proposed BGS algorithm, also are explained in chapter 3. Chapter 4 discusses the performance of the system in terms of CPU, memory utilization and segmentation quality such as F-measure, SSIM, D-score and FSD. In addition, chapter 5 concludes the thesis by summarizing the important ideas and discusses possible direction for future employment and contributions.

CHAPTER 2 : LITERATURE REVIEW

2.1 Introduction

This chapter contains five (5) sections, which starts with a critical review about travel time estimation based on different devices that can be used as tools for travel time measurement. The existing traffic monitoring systems in Malaysian highways are outlined together with the first section. Then, the discussion continues by focusing on the motion detection which contains introduction of background subtraction, available background subtraction algorithm and the evaluation method for measuring the background subtraction algorithm performance. Next, the discussion is followed with an explanation of available embedded platforms that are currently used as the core platforms for image processing. And lastly, the embedded softwares that relates with the proposed background subtraction algorithm are described as well.

2.2 Travel Time

In Malaysia, in some part, the expressway are monitored using an Integrated Transportation Information System (ITIS). Advanced Traffic Management System (ATMS) collects all the traffic information for ITIS via several devices such as Automatic Detection System (AVDS). AVDS technology relies on digital-based image processing and is capable of detecting incidents happening around the coverage area. It also collects traffic data like average speed, traffic volume, vehicle count, and occupancy of each lane (Masbah & Abidin, 2016).

AVDS works based on video image processing for traffic detection and it is control via an integrated Central Computer Software (CCS) at control centre. Two units of cameras are installed at strategic locations in every area to improve detection accuracy and reduce false alarm. An Intelligent Transport System (ITS) software system is used for integration with the AVDS, CCTV, and Variable Message Sign (VMS) (Dahl, Sjåfjell, & Skogen, 2014). The software provides centralised management, surveillance, and monitoring along the PLUS expressway. Using the central computer system also, traffic data collection, travel time, CCTV selection and control, VMS Display, alarm and event, incident and action plans, can be managed easily (Masbah & Abidin, 2016).

Travel time is the total time takes to get from A (origin) to point B (destination) per kilometre, i.e. it concludes all the time spent stopped on a route or it is the elapsed time it takes for a vehicle to traverse a given segment of a street (Pionke & Beal, 1997). Considering travel time information as the most important performance measure for a traffic information system. The prediction of travel time can help travellers or road users make reliable travel decisions, especially regarding route choice and time of departure (Yildirimoglu & Geroliminis, 2013). Spot speed measurement systems measure traffic stream speeds over a short roadway segment at fixed location along roadway. These spot speed measurements are used to compute spatial travel times over an entire trip using space-mean-speed estimated (Dion & Rakha, 2006).

Vehicle speed is calculated by comparing two successive serial frames, where a matching technique is used to determine the distance of the moving vehicle between consecutive frames (Hagargund, Udayshankar, & Rashmi, 2013). The displacement of the vehicle and its speed is important. These two parameters are related to travel time, as the travel time is the time taken for the vehicle to go through the distance at a certain speed. Vehicle speed, v is described in Equation (2.1) (Yan, Yancong, & Zengqiang, 2010).

$$v = \frac{\Delta d}{\Delta t} = \frac{(d_{i+1} - d_i)}{(t_{i+1} - t_i)} \tag{2.1}$$

Where, t_i and t_{i+1} represent the time elapsed between two sequential frame; d_i and d_{i+1} represent the location of detected vehicle from two image.

Travel time study is usually used to record congestion and to compute the actual effect of highway improvements. Most travel time estimation used a probe vehicles to quantify the travel time. This technique are theoretically much uncomplicated, however its implementation tends to be costly due to the labour intensive. Routinely two experts are involved in the process: one will control the vehicle, and another one document the location, the distance and time the vehicle passes between agreed checkpoints.

Then, few different devices come out and can be used to measure and estimate the vehicle speed and travel time. Figure 2.1 describes the devices used to measure travel time. In evaluating the level of service of a road network, a paper from Yamazaki, Uno, & Kurauchi (2012) proposed a method to estimate travel time using electronic toll collection (ETC) devices. When vehicles moved between the interchanges, the travel time is evaluated based on the average travel time between from ETC data for each 15 minutes interval.

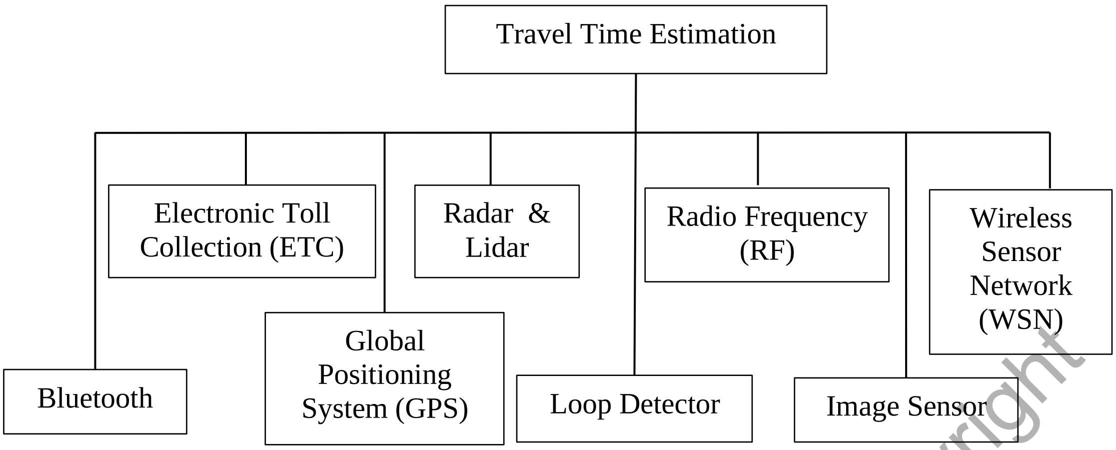

Figure 2.1 Available devices used to estimate travel time

There has also been a growing curiosity in evolving an unidentified probe vehicle monitoring system to estimate travel times on highways and arterials based on wireless signals available from latest technologies such as Bluetooth (Bachmann, 2011; A. Bhaskar, Qu, & Chung, 2015; Blogg et al., 2010; Laharotte et al., 2014). As an example, a paper from Blogg et al. (2010) proposed a unique way to use Bluetooth media access control (MAC) address data for collecting speed and travel time data. In this approach, multiple MAC readers are placed at strategic location to detect wirelessly the MAC devices and capture the time taken between reader's points.

In addition, Radio Detection and Ranging (Radar) and Light Detection and Rangin (Lidar) devices are also one of the popular methods for detecting vehicle speed. Radar works by sending a radio signal to the moving vehicle after which a receiver will pick up the reflected signal. Then, the receiver determines the speed of the moving vehicle by measuring the different frequency between the sending signal and received signal

(Hagargund, Udayshankar, & Rashmi, 2013). Meanwhile, Lidar uses the same technique as Radar, but Lidar measures the time taken for a light pulse to move out from the Lidar gun to the vehicle and bounce back to the receiver. Both devices, although able to determine the vehicle speed accurately, are costly (Lin, Li, & Chang, 2008).

Another study proposed the use of another tool, radio frequency (RF), for a vehicle detection and speed estimation system. The system used wireless signal strength in an RF zone to estimate the street states and car speed. Even though the signal strength data can be acquired by RF transmitters and receivers mounted on the sideways of the route, this proposed system demands to use common wireless networks such as Wi-Fi or cellular networks to discover the traffic characteristics and calculate approximately the vehicle speed. When vehicle with mobile devices carried by used user entering the area of interest create an increase in the variance of stream lying at the entrance. When vehicle is exiting the area of the interest, the increase of in the variance of the stream is lying at exit. The time between these two events can be taken to estimate the vehicle speed (Kassem, Kosba, & Youssef, 2012).

The wireless sensor network (WSN) is also one of the tools used for measuring travel time. One study proposed a method in which a probe vehicle equipped with an on-board unit tag is continuously monitored by fixed access points. The moving vehicle (tag) communicates with two anchors and a base station, which collects data for calculating tag position and velocity. When a vehicle (tag) moves at a lower speed, the measurement and estimated results will be highly accurate. However, the result becomes less accurate when the vehicle moves at a higher speed due to noise and error (Saqib, Khan, & Basalamah, 2011).

The Global Positioning System (GPS) is another well-known tool for measuring travel time data (Gao, Zhu, Wan, & Wang, 2013; Kumar, Vanajakshi, & Subramanian, 2011; Osamura, Yumoto, & Nakayama, 2013). As presented by Gao et al. (2013), a taxi equipped with a GPS device could be used to collect and evaluate travel efficiency in a city. The taxi's GPS can be used for constructing routes, destination prediction, centrality computation, and outline detection.

Travel time also can be obtained through a loop detector. The loop detector contains of one or several loops of wire-embedded inductive on the roadway and linked to a control box. This loop generates an inductive element in combination with the control box. Once a vehicle goes by or stopover on the loop, the inductance of the loop is become low. A vehicle detection is then signalled in the control box (Bachmann, 2011; Xia, 2006).

Lastly, the imaging technique based on image sensor is another conventional and simple method for measuring vehicle speed and travel time (Goda, Chen, & Zhang, 2014; Pornpanomchai & Kongkittisan, 2009). A video image processor is a join up of software and hardware, which gains looked-for information from data fed by an imaging sensor. This imaging sensor can be a standard camera such as web camera or an infrared camera. This type of device can detect speed, occupancy, count, and presence of the vehicle.

These methods can be divided into two types—intrusive and non-intrusive. The intrusive method means that the device has a receiver and transmitter, in which one would be installed on the vehicle. The intrusive method, such as GPS, Bluetooth, wireless sensor network, and inductive loop detectors, are generally used, but they require complex installation and maintenance. Meanwhile, the non-intrusive methods consist of stand-alone devices and does not depend on the vehicle. The non-intrusive methods, such as toll tag matching, radar and video image processor, are able to avoid these problems, but they are usually less accurate and stable (Mimbela & Klein, 2000).

There are advantages and disadvantages of using different devices for measuring travel time. The GPS can accurately calculate the logging location and velocity of a vehicle from receiving information via GPS satellites. Even so, a tunnel or tree canopies along the highway can block its signal. Moreover, the data management in the system using GPS is quite difficult to handle because it consists of a large data set and the cost of collecting the data is also high due to the cost of vehicle maintenance probes, vehicle driver probes, and so on (Blogg et al., 2010; Khoei, 2014).

Meanwhile, when using WSN, multiple vehicles along the highway will need multiple tag devices, which can increase the cost for system development. For extensive networks such as a highway, the daily fluctuations indicated by Bluetooth data can be used to supplement traditional methods. However, the data must be collected over a longer period of time, which is cost prohibitive. Also, this method requires a lot of test cars to collect data. The same goes for RF-based measurement. Both systems are quite complex to implement for long-distance travel, and both require a good antenna and suitable antenna placement (Blogg et al., 2010).

Measurement using Radar, Lidar, or image sensors are therefore considered more suitable for measuring travel time and speed of a vehicle in the context of this study due to their ability to perform measurement on their own without depend on the measured object. The disadvantage of Radar and Lidar is that they use active sensors, which principally operate by sending a signal out and receiving the reflected signal for measuring the distance of an object. These types of sensors have some major drawbacks; especially range limitations, surface material, and air purity. They are also generally more costly compared to image sensors. Furthermore, active sensors sometimes cannot be used due to security reasons or interference (Makwana & Goel, 2013).

Travel time measurement using the imaging technique, on the other hand, passively captures the environmental scene and does not suffer from the drawbacks mentioned above. Generally, this type of device operates by selecting few vehicle detection zones covered by the limited field of view (FOV) of the camera. Image processing algorithms are put into real time operation to these zones in order to obtain the looked-for information such as vehicle position or elapsed time taken from the vehicle moving around the FOV (Doğan, Temiz, & Külür, 2010).

The imaging technique provides plentiful information to understand the environment scene. In fact, current computer technology can offer an extremely powerful computing process speed, which is able to assist the processing of digital images almost in real time. Therefore, research should invest more effort and time on computer vision system based on imaging technique to solve the problems compared using other methods (Ambardekar, 2007).

The imaging technique for travel time measurement has issues when a low-cost and low processing platform such as an embedded platform is used. The problems include creating the imaging technique-based application. Besides the quality of the capturing images that should be improved, the hardware performance, such as computer resources, also have to be considered in order to construct a robust application in terms of software and hardware (Muchala, Pothalaiah, & Brahmareddy, 2012). Table 2.1 summarizes the advantages and disadvatages of all available devices that can be used for estimating travel time.

Table 2.1 Advantage and disadvantage of travel time estimation devices

Device	Advantages	Disadvantages
ETC	No installation required in vehicles No training required for field staff	Large number of vehicles are used, cycle time between runs can be very large and not capture peak conditions
Bluetooth	No training required for field staff deploying roadside equipment.	Data collection in long period of time and increase cost Requires good antenna and suitable antenna placement
RF	No training required for field staff driving floating vehicles	Requires a lot of test cars to collect data. Requires a good antenna and suitable antenna placement
GPS	Accurately calculate the logging location and velocity	Signal can be blocked by tunnels and tree canopies Large data set and high cost
Loop Detector	No training required for field staff driving floating vehicles	Data management, particular with large data sets where one wishes to keep track of drivers
WSN	Accurately calculate the location and velocity of tag	Multiple tags is needed Increase cost
Radar & Lidar	No installation required in vehicles	Training field personnel on good data management field procedures. Sensors have some major drawbacks such as range limitations, surface material, and air purity
Image Sensor	No installation required in vehicles	The quality of the capturing images (hard to track)

2.3 Motion Detection

There are a wide-range of systems based on image and video that use different methodologies to detect vehicles and objects. Motion detection is the initial processing phase in this type of system. Practically every visual surveillance system begins with motion detection. Motion detection targets at segmenting areas by corresponding to the moving objects based on the image sequences. Following processes like tracking and recognition are greatly reliant on this step. The process of motion detection usually involves background modelling, foreground detection, and object classification, which intersect each other during processing (Hu, Tan, Wang, & Maybank, 2004).

2.3.1 Background Subtraction

Because motion detection is the first step in image processing, the process must be simple, light, and as fast as possible (Benezeth, Jodoin, Emile, Laurent, & Rosenberger, 2010; M.L.J. & Indumathi, 2017). One of the straightforward forms of motion detection is through using BGS. BGS is a ‘quick and dirty’ way for detecting moving objects from static cameras (Piccardi, 2004). Given an image sequence, the main objective of BGS is to identify from the current frame the number of pixels that are totally dissimilar from the earlier frames.

It is not easy to detect motion using BGS. The requirements and limitations of the detection algorithm are based on the proposed application and are different for different applications. The main challenges of the BGS algorithm that been addressed by Brutzer, Benjamin, & Heidemann (2011) and Von (2014) are :

- Illumination changes gradually and mainly happen in outdoor environments at different times of the day. It may affect the appearance of the objects in the observed scene.

- Illumination changes unexpectedly, which are mainly happen in outdoor environments with unstable climate conditions such as the clouds without warning cover the sun.
- Shadows are mainly generated by moving objects and cause difficulties in segmentation of objects.
- Dynamic background—those parts of the background presenting different looks due to moving objects such as waving trees, camera jitter, low frame rate video and so on.

2.3.2 Taxonomies of BGS

The BGS methods can be organized into the following groups: Basic background modelling, statistical background modelling, non-parametric method, neural and fuzzy background modelling, and background estimation. Another classification can be done in terms of prediction, recursion, adaption, or modality. All these modelling approaches are used in the BGS context, which present background initialization, background modelling, background maintenance, and lastly foreground detection (Bouwmans, El Baf, & Vachon, 2008).

Various BGS approaches have been proposed nowadays with different taxonomies. Based on the spatial level, BGS approaches can be divided into several classes :

i) Feature size: a pixel (Barnich & Van Droogenbroeck, 2011; Hofmann, Tiefenbacher, & Rigoll, 2012; Zivkovic, 2004; Zivkovic & Van Der Heijden, 2006), a block (Elharrouss, Moujahid, & Tairi, 2017; Savas, Demirel, & Erkal, 2018) or a cluster (H. Bhaskar, Mihaylova, & Achim, 2008, 2010; Fang et al., 2018)

ii) Feature size: colour features, motion features, edge features, texture features, and stereo features.

Based on feature size, pixel-type algorithms only use elements assembled at each single pixel position. This method is very fast and quick, but it does not have any kind of inter-pixel relationships. There have been countless proposals in the literature for these type of methods such as Running Gaussian Averages, Median Filtering, and Gaussian Mixture Models, have been of special relevance and have invented a huge number of developed system. While block-level based methodologies allocate an image into blocks and compute block-related features to define the background. Block-level approaches are normally more robust against noise than pixel-level approaches plus it provides alternative detections of the foreground objects and also computationally expensive. Lastly, cluster-level based algorithms split an image into a number of regions, which are then sorted as background or foreground. There is a small number of purely region level-based algorithms because finding great weight of regions in an image by means of spatial stability criteria can be computationally expensive (Von, 2014).

In reference to the process of updating the background model, background subtraction approaches can be separated into recursive and non-recursive. Recursive approaches update the background model as new observations arrive, therefore consuming low resources in terms of computational and memory requirements. On the other hand, non-recursive approaches uses a sliding-window approach for background estimation. It stores a buffer of the previous video frames to estimate the background. Therefore, non-recursive approaches have higher memory requirements. Nevertheless, since this method retains a copy of the most recent video frames, it can deal with some challenges such as fast convergence and outlier rejection, which cannot be simply handled by recursive techniques (Kaur, 2017).

2.3.3 Relevant BGS Approaches

In order to overcome BGS challenges, several techniques have been proposed in the past decade. These techniques have their own strengths and weaknesses in terms of performance and computational requirement. In choosing the type of BGS to use for the system, the researchers need to consider the position of the system that will be placed as well as the processing speed of the embedded board. Computational load and memory requirement is important information that one has to take into account before choosing which background subtraction technique to implement (Kaur, 2017; G. Yao, Lei, Zhong, Jiang, & Jia, 2017).

In general background modelling algorithms classified into recursive and non recursive models (Angelo, 2018). Non-recursive techniques such as frame differencing (Singla, 2014; Stalin Alex & Wahi, 2014), median filter (Hung, Pan, & Hsieh, 2014), linear predictive filter (Narayana, 2007) and the non-parametric model proposed by researches in general are techniques that use a sliding window method for background estimation. These techniques keep a buffer of previous video frames and model a background based on the statistical properties of these frames, which consume a high memory usage.

An alternative approach utilised in recursive techniques are the Kalman filter (Ambardekar, 2007; Harnisch, 2013) and a mixture of Gaussian (Das & Saharia, 2014). These methods do not keep a buffer for background estimation but update a single background based on input frame. The recursive method preserves a single background model that is updated with new incoming frame, resulting in minimal memory requirements and computational efficiency as compared to the non-recursive approach.

Frame difference, as shown in Figure 2.2 is a simple, fast, and basic background modelling technique, which only uses a single previous frame as the reference frame. It is commonly used approach for detecting moving objects (Ramya & Rajeswari, 2016). The disadvantage of this technique is it could not identify the interior pixel of a large, uniformly coloured moving object. This technique is fast because it only involves two differenced image subtraction for the foreground mask (Domínguez et al., 2011; B. Wang & Dudek, 2014). There are also an expansion from this technique by introducing three-frame difference algorithm (Sharma, Agrwal, Gupta, & Prajapati, 2017; Xue-lian, Jian-hua, & Wei, 2018) due to fewer problems of holes. Dynamic background technique is used for video with background change. Background is updated by averaging these three frames(*t-1*, *t+1* and previous update background) and the updated background is subtracted from frame t for foreground detection (Sharma et al., 2017). Even though this expansion was the improvement of the basic frame difference, Z. Xu, Zhang, & Du (2017) revealed that this method will cause a problem especially on elongated target, blurred outline and noise.

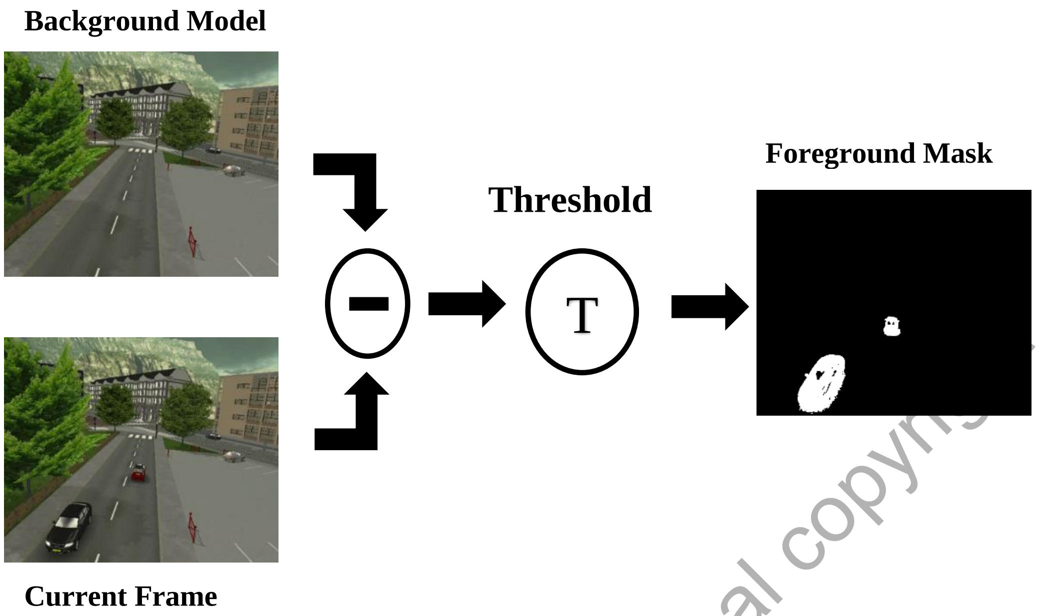

Figure 2.2 Frame difference method

The median based BGS algorithm is a modification from frame difference by taking the median value of a set number of previous frame to construct a back plate model. This algorithm compares the pixel in the same way as the frame difference (Komagal & Vinodhini, 2012). Due to higher number of false positives and the segmentation quality is depends on sensitive thresholds, a study from Cyganek (2017) proposed method has better accuracy than the classical median method. The study claimed that the proposed method also allows real-time operation on high definition (HD) video streams.

Adaptive background learning (ABGL) technique is also one of the basic background modelling techniques which based on frame different method (Sobral & Vacavant, 2014). It presents equations that are used to model each frame; and based on this background model, the background subtraction of the frames are done. In simple terms, the algorithm generates the current background based on the segmentation results produced from the frame difference method. As stated in paper by Zhang, Chen, Shyu, & Peeta (2003), ABGL is used for analysing the traffic video sequence by proposing vehicle detection and tracking, which include image and video.

The most widely used algorithms for BGS found in the literature and usually used for comparison purposes in the current work is the Mixture of Gaussians Model (MOG) (Mandellos, Keramitsoglou, & Kiranoudis, 2011). The MOG approach models each pixel history as a cluster of Gaussian-type distributions and get through an online approximation to update its parameters. Based on this step, the background is determined as the expected value of the distribution corresponding to the most populated cluster (Stauffer & Grimson, 1999). This methodology is greatly enhanced on grounds of performance by allowing recursive equations to adaptively update the parameters of the Gaussian model (Zivkovic, 2004). But MOG approach cannot handle fast variations with accuracy using a few Gaussians and therefore this approach has problems for sensitive detection (changes in illumination condition, background trembling and so on) of foreground region (Manaswini, 2013). The high computational complexity of the Gaussian based method acts as a bottleneck in the real time implementation in surveillance system with low computational power (Shah, Pingale, Patel, & George, 2017; Shen et al., 2016).

Another popular algorithm for background subtraction and frequently used for multimodal background is the so-called Codebook (Mandellos et al., 2011). Codebook is the most sensitive colour-based background subtraction methods and can be applied both indoor and outdoor scene. Based on a training sequence and without probabilities evaluation, the method allocates to each background pixel a series of key colour values