



UniMAP

**Development of Video Image Acquisition for Traffic
Surveillance using Open Source Software**

By

**Zainab Nazar Khalil Wafi
(0730210206)**

A thesis submitted in fulfillment of the requirements for the degree of
Master of Science (Computer Engineering)

**School of Computer and Communication Engineering
UNIVERSITI MALAYSIA PERLIS**

2011

ACKNOWLEDGEMENT

بسم الله الرحمن الرحيم

Praised to Him, Allah The Most Gracious and The Most Merciful and may His shalawat with salam always for Prophet Mohammad (p.b.u.h) and his holy progeny eternally. Uncountable thanks for my father Professor Dr. Nazar K. Wafi and my mother for their love and blessings which always gives me strength through this life and further step of my way.

I would like to express my gratitude and thanks to Professor Dr. R. Badlisha Ahmad for his patience and guidance in supervising my work with high quality standard. I would never forget my most sincere thanks to my co-supervisor Professor Dr. Paulraj Murugesu Pandiyan for his guidance, and encouragement throughout attending his course in machine-vision. Without their help, I would not have been able to achieve what I have achieved so far.

I like to dedicate a special thanks to Professor, Mr. Moharif and Mrs Risindar family for their helping and supporting me to finish this work. Moreover I deeply appreciate a group of postgraduate students who support and help me during this research.

Last, but not least, my appreciation to my siblings and friends that were working together with me in General System Company (GSC) for their supporting to me during my study.

Zainab Nazar Khalil Wafi

zainabwafi@yahoo.com

TABLE OF CONTENTS

	Page
APPROVAL AND DECLARATION	i
ACKNOWLEDGEMENT	ii
TABLE OF CONTENTS	iii
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF ABBREVIATIONS	xiii
ABSTRAK	xvi
ABSTRACT	xvii
CHAPTER 1	
INTRODUCTION	1
1.1 Introduction	1
1.2 Overview of video surveillance system	3
1.3 Problem Statement	4
1.4 Objectives	5
1.5 Thesis layout	5
CHAPTER 2	
LITERATURE REVIEW	7
2.1 Introduction	7
2.2 Fundamental of Image Processing and Its Applications	7
2.2.1 Digital Image System Components	8

2.2.2	Literature about traffic surveillance	9
2.3	Camera System	10
2.3.1	Cameras and Video System	11
2.3.2	Using Video Systems in Roads	11
2.4	Related Projects	14
2.5	Reasons of choosing GNU- Linux in systems	16
2.5.1	Literature about to GNU-Linux	17
2.5.2	Open Source Software / Free Software	18
2.6	Image Acquisitions in Linux	18
2.7	OpenCV	20
2.7.1	OpenCV Structure	22
2.7.2	General Description of OpenCV	23
2.7.3	Features	23
2.8	Summary	24
CHAPTER 3		
METHODOLOGY		26
3.1	Introduction	26
3.2	System Design	26
3.3	System Installation	27
3.4	System Requirements	29
3.4.1	Desktop / Laptop	29
3.4.2	Camera	30
3.4.2.1	Camera Hardware description and characteristics	30
3.4.2.2	Camera Position	31

3.5	Software Development	33
3.5.1	Socket Programming	34
3.5.2	OpenCV Programming of Surveillance Application	37
3.6	Streaming Data	38
3.6.1	Design of Stream Capture	38
3.6.2	Server Implementation	39
3.6.3	Client Implementation	43
3.6.4	Data processing Design	47
3.6.4.1	Initialization Capturing	49
3.6.4.2	Conversion to Gray scale	50
3.6.4.3	Update the Background	52
3.6.4.4	Gaussian Filter algorithm	53
3.6.4.5	The MOG algorithm	54
3.6.5.6	Object contouring and bounding in a box	59
3.6.5.7	Specify the ROI	62
3.6.5	Resize the frames use scaling Images	64
3.6.6	Saving frames	67
3.7	Summary	68

CHAPTER 4

RESULTS, DISCUSSIONS AND PERFORMANCE ANALYSIS	70	
4.1	Introduction	70
4.2	Processing Time Measurement	71
4.3	Capture Initialization and reading frames	71
4.4	Gray Scale Conversion	74

4.5	Background dynamic modeling	76
4.6	Foreground detection Operation	80
4.7	Tracking the moving vehicles by bounding in Boxes	83
4.8	Save the abnormal movement from specific points	87
4.9	Performing Analysis of Traffic Surveillance Algorithm	88
4.10	Comparison of save Images in different formats	95
4.11	Scaling Frames Comparison	96
4.12	Summary	106
CHAPTER 5		
CONCLUSION		
		107
5.1	Introduction	107
5.2	Contribution	108
5.3	Future Works	109
REFERENCES		
		111
PUBLICATIONS		
		117
APPENDICES		
APPENDIX A	Choosing Image Format	120
APPENDIX B	OpenCV Installation in Debian/Ubuntu Linux OS	127
APPENDIX C	Linux Directories	129
APPENDIX D	OpenCV modules	131

LIST OF TABLES

Table No.	Table Description	Page
2.1	Projects comparison on surveillance	25
3.1	Computers specification for code test	29
3.2	Conversions available by means of cvCvtColor()	52
3.3	The resolution of captured frames and resizing frames	66
4.1	Processing Steps for the image in the system	70
4.2	Description files for experimental steps	72
4.3	Total time calculation of the processing steps using different CPU Speeds	92
4.4	Total time comparison for algorithm steps in three platforms	93
4.5	calculation times for three platforms comparison repeated processing for each frame	95
4.6	Processing Time differences for algorithm steps using different CPU platforms and different frames resolution	98
4.7	Overall Processing Time differences using different CPU platforms and different frames resolution	99
5.1	Comparisons between two SBC boards compatible with OpenCV	132

LITS OF FIGURES

Figure No.	Name of Figures	Page
2.1	Components of Digital Images processing System	9
2.2	The Basic structure of OpenCV	22
3.1	System Design	27
3.2	Block diagram for hardware and software integration	28
3.3	logitech-quickcam-communicate-stx-plus	30
3.4	Cars view depends on camera position; a. The camera on the traffic light system b. The camera on the road lighting system c. The Camera on the pedestrian bridge	32
3.5	Software design steps	33
3.6	Socket calls for connection-oriented mode	35
3.7	Socket calls for connectionless mode	36
3.8	Diagram of Server Stream	39
3.9	Algorithm of Frame Grabber streaming	40
3.10	a Lock mechanism for img1	41
3.11	Stream Server Thread algorithm	42
3.12	Client Stream diagram	43
3.13	Client Stream Algorithm	45
3.14	Algorithm of Video Display for client	46
3.15	Flowchart of Data Processing from Camera	48
3.16	Image results for the algorithm steps	49
3.17	Input data from Camera or a File	50

3.18	OpenCV Algorithm for MOG processing	55
3.19	Background Updating frames	57
3.20	Call the morphological transformations	58
3.21	smoothing image by filtering the noise a. Original frame b. binary image with noise c. filter the noise of the image	59
3.22	Contour calling function	59
3.23	MoG image result (a) a frame from a video (b) a result of MoG with contour the moving object	60
3.24	A frame from a video with a result of bounding the tracking object in a rectangle box	61
3.25	The use of cvRectangle in ITSS system	61
3.26	Images of vehicles turning abnormally in KL	62
3.27	ROI coding	62
3.28	Video frames of ROI, Gaussian and bounding by box	63
3.29	Resize frames function	64
3.30	Calling Resize function to resize frames	65
3.31	Algorithm of scaling frames	66
3.32	Image scaled from 320*240 to 80*60 Pixels	67
3.33	Saving frames code	68
4.1	Graph of Capture initialization for frames	72
4.2	Average of Initializing Capturing graph using different processor speeds	73
4.3	RGB image sunny	74
4.4	Gray scale image	74
4.5	RGB image in rainy	74
4.6	Grayscale image in rainy	74

4.7	Gray scale converting graph between three different processor speeds	75
4.8	Average of Gray scale converting graph using different CPU speeds	76
4.9	Shows Background modeling frames	76
4.10	Adapting Background Dynamic frames using 1.2GHz processor speed for 9 views	77
4.11	Adapting Background Dynamic modeling frames using 2.0GHz processor speed for 9 views	77
4.12	Adapting Background dynamic modeling frames using 2.6 GHz processor speed for 9 views	78
4.13	The different of Background dynamic modeling using three different CPUs for one of videos view	78
4.14	Average time of background modeling in different views frames	79
4.15	Process of foreground detection	80
4.16	Graph of foreground detection using 1.2 GHz processor speed	80
4.17	Graph of foreground detection using 2.0 GHz processor speed	81
4.18	Graph of foreground detection using 2.6 GHz processor speed	81
4.19	The differences of foreground detection using three different CPUs for one of videos view	82
4.20	Original vehicle and the vehicle bounding by box	83
4.21	Detecting vehicle movement on rainy day	83
4.22	Time caculation of bounding objects inside box use 1.2 GHz processor Speed	84

4.23	Time caculation of bounding objects inside box use 2.0 GHz processor Speed	84
4.24	Time caculation of bounding objects inside box use 2.6 GHz processor Speed	85
4.25	Time calculation average of tracking objects by bounding by box processing	85
4.26	The differences of objects bounding using three different CPUs for one of videos view	86
4.27	Detecting and saving the abnormal vehicles movement (a) Original images (b) Binary Images	87
4.28	Time calculation for Detecting and saving the abnormal vehicles movement using two diffrent processors speeds	88
4.29	Processing steps time performance in different times of the day use 1.2 GHz preprocessor Speed	89
4.30	Time of processing steps comparing during the day use 1.2 GHz processor speed	89
4.31	The processing steps time performance in different duration of the day use 2 GHz processor speed	90
4.32	The time of processing steps comparing during the day use 2 GHz processor speed	90
4.33	The processing steps time performance during the day use 2.6 GHz processor speed	91
4.34	The time of processing steps comparing during the day use 2.6 GHz processor speed	91
4.35	Average of time calculation for processing steps during the day using three different processor speeds	92

4.36	Saving images per second	95
4.37	Time calculation comparison of processing steps using different frames resolution with 2.6 GHz processor Speed	96
4.38	Time calculation comparison of processing steps using different frames resolution with 2.0 GHz processor Speed	97
4.39	Time calculation comparison of processing steps using different frames resolution with 1.2 GHz processor Speed	97
4.40	Tracking moving object and detecting abnormal turning	99
4.41	Tracking the movement of vehicles in sunny day	100
4.42	Tracking the movement of vehicles on a rainy day	101
4.43	Tracking the movement of vehicles in another view of rainy day	102
4.44	Tracking the movement of vehicles in rainy night	103
4.45	View of rain after noon site 1 view 1 (file V1)	104
4.46	View of rain after noon site 1 view 2 (file V2)	104
4.47	View of rain after noon site 2 view 1 (file V3)	104
4.48	View of rain after noon site 2 view 2 (file V4)	104
4.49	View of rain in night site 3 (file V5)	105
4.50	View of rain in night of site 4 (file V6)	105
4.51	View of rain in night of site 1 (file V7)	105
4.52	View of rain in night of site 2 (file V8)	105
4.53	View of sunny in site 1 (file V9)	105

LIST OF SYMBOLS, ABBREVIATIONS OR NUMENCLATURE

Term	Description
2D	<i>2 Dimension</i>
3D	<i>3 Dimension</i>
API	<i>Application Programming Interface</i>
BG	<i>BackGround</i>
BGR	<i>Blue Green Red</i>
CCTV	<i>Closed-Circuit Television</i>
CPU	<i>Central Processor Unit</i>
CT	<i>Computer Tomography</i>
CV	<i>Computer Vision</i>
DFSG	<i>The Debian Free Software Guidelines</i>
f/sec	<i>frame per second</i>
FHS	<i>Filesystem Hierarchy Standard</i>
FPGA	<i>Field Programmable Gate Array</i>
FTP	<i>File Transfer Protocol</i>
gcc	<i>Gnu C Compiler</i>
GHz	<i>Giga Hertz</i>
GIF	<i>Graphic Interchange Format</i>
GIMP	<i>GNU Image Manipulation Program</i>
GUI	<i>Graphical User Intrface</i>
HCI	<i>Human Computer Interface</i>
I/O	<i>Input / Output</i>
IMCASM2	<i>Intelligent Multi-Camera Surveillance and Monitoring 2</i>

IP	<i>Internet Protocol</i>
IPP	<i>Integrated Performance Primitives</i>
iSEE	<i>Internet Sensor Exploration Environment</i>
ISO	<i>International Organization for Standardization</i>
ITS	<i>Intelligent Transport Systems</i>
IVMA	<i>Intelligent Visual Monitoring Algorithms</i>
JPG	<i>Joint Photographic Group</i>
kB	<i>Kilo Byte</i>
KDE	<i>K Desktop Environment</i>
LAMP	<i>Linux, Apache, MySQL, PHP, Perl, and Python.</i>
LAN	<i>Local Area Network</i>
LSB	<i>Linux Standard Base</i>
MIME	<i>Multipurpose Internet Mail Extensions</i>
MLL	<i>Machine Learning Library</i>
MOG	<i>Mixture of Gaussians</i>
MRI	<i>Magnetic Resonance Imaging</i>
ms	<i>Millisecond</i>
OpenCV	<i>Open Computer Vision</i>
OS	<i>Operating System</i>
OSI	<i>Open System Interconnection</i>
OSS	<i>Open Source Software</i>
PC	<i>Personal Computer</i>
PDA	<i>Personal Digital Assistant</i>
PeMS	<i>Performance Measurement System</i>
PMMU	<i>Paged Memory Management Unit</i>

PNG	<i>Portable Network Graphics</i>
RAM	<i>Random Access Memory</i>
RGB	<i>Red Green Blue</i>
ROI	<i>Region Of Interest</i>
ROM	<i>Read Only Memory</i>
SCSI	<i>Small Computer System Interface</i>
SQL	<i>Sequential Language</i>
STEMS	<i>Smart Traffic Evacuation Management System</i>
T.R.A.C.E	<i>Total Recognition by Adaptive Classification Experiments</i>
TCP	<i>Transmission Controlling Protocol</i>
TIF	<i>Tagged Image File</i>
TV	<i>Television</i>
UDP	<i>User Datagram Protocol</i>
USB	<i>Universal Serial Bus</i>
VGA	<i>Video Graphic Array</i>

© This item is protected by original copyright

Mereka bentuk perolehan imej video untuk pemantauan Trafik menggunakan perisian sumber terbuka

ABSTRAK

Penyelidikan ini mempersembahkan inovatif dalam sistem memantau dihasilkan dengan mengaplikasi kaedah baru memprogram yang digunakan untuk Sistem Pengawasan Trafik/Jalanraya. Kini, perkembangan sistem pengawasan trafik di jalanraya adalah satu keperluan terutamanya di jalanraya dan lebuhraya kawasan bandar yang semakin sesak. Kesesakan ini menyumbang kepada peningkatan berlakunya kemalangan jalanraya. Sistem pengawasan trafik melakukan pengenalan imej dan proses menjejak objek. Kamera dipasang untuk mengawas jalanraya yang khas (lebuhraya, persimpangan...sb) dan berhubung dengan *server* utama. *Server* utama menyatukan sasaran imej dari kamera selepas memproses imej melalui beberapa langkah, kemudiannya berhubung dengan komputer yang lain dengan menghantar imej tersebut ke komputer itu. Prototaip pelaksanaan pengawasan trafik adalah berdasarkan program OpenCV dibawah GNU-Linux sokongan program *socket*. Objektif-objektif sistem pengawasan trafik adalah menjejaki pergerakan kenderaan, menjumlah bilangan kenderaan dan mengesan pergerakan kenderaan yang menyalahi undang-undang di tempat-tempat tertentu. Perkara tersebut dilakukan dengan menganalisa setiap *frame* yang di tangkap dan memproses *frame* tersebut menggunakan fungsi OpenCV. kegunaan sumber seperti fungsi OpenCV dalam GNU-Linux dari sumber terbuka adalah untuk menyediakan kaedah mudah bagi menggunakan rangka versi komputer mahupun dapat menjalankan kod visi dalam masa nyata. Pengawasan trafik berfungsi dalam proses masa-nyata untuk 5-33 fps perjalanan rakaman pada hari yang hujan mahupun hari yang terang. Sistem pengawasan trafik menerima rakaman sama ada dari tangkapan kamera ataupun dari fail. Sistem tersebut menjejak pergerakan kenderaan dengan pemrosesan imej dan pengenalan *algorithms* seperti penukaran ke *Gray scale* dan campuran kaedah Gaussian; tanda setiap kenderaan yang bergerak dengan petak segi empat sama dan mengira jumlahnya. Sistem ini mengandungi; bahagian menangkap, bahagian menjejak, bahagian mengenalpasti pergerakan yang menyalahi undang-undang di titik-titik tertentu dan mengalurkan pameran tersebut kepada komputer yang berhubung. Selain itu, sistem ini mengecilkan proses *frame* dalam keadaan resolusi tinggi tangkapan *frame* tersebut. Sistem ini berjaya di uji dengan tiga pemproses kelajuan yang berlainan, 1.2 GHz CPU, 2.0 GHz CPU dan 2.6 GHz CPU. Hasilnya di ukur dalam unit mili-sesaat dan hampir tepat. Sistem ini berfungsi di persekitaran luar yang rumit dengan ayunan dahan pokok dan hujan. Selain itu, sistem ini mengemaskini setiap *frame* dalam mana-mana modul latar belakang dari pandangan atas jalan bagi mendapatkan imej penting semua objek. Semua keputusan akhir boleh direkodkan dengan menyimpan kumpulan gambar-gambar terpilih dalam mana-mana format imej. Kelajuan pemproses CPU bersama saiz proses *frame* mewakili salah satu faktor utama analisis prestasi pengawasan trafik. Perbandingan kelajuan memproses langkah-langkah memproses keputusan dalam resolusi berlainan untuk saiz *frame* dilakukan dan keputusan yang terpenting diperolehi dengan mengurangkan pengiraan masa untuk memproses apabila saiz *frame* yang ditangkap dikecilkan. Eksperimen ini di laksanakan dengan menggunakan 9 pandangan dengan 700 *frame* perspektif berlainan bagi setiap pandangan. Situasi pergerakan kenderaan dalam hujan dan panas di ambil di sekitar negeri Perlis. Ujikaji ini berjaya mengesan pergerakan semua jenis kenderaan.

Development of Video Image Acquisition for Traffic Surveillance using

Open Source

ABSTRACT

The Research works to presents an innovative approach in monitoring system by applying new programming methods which can be used for Traffic/Road Surveillance Systems. Nowadays, the development of traffic surveillance on the road is compulsory as highways and roads are getting crowded especially in the cities. These crowded situations would increase the number of accidents on the roads. The Traffic Surveillance System performs image recognition and object tracking processing. A camera is used to monitor specific roads (highways, motorways, junctions...etc) and communicate with the main server. The main server integrates the target image from the camera after processing it in several steps then communicates with another computer by sending the image to it. A prototype implementation of traffic surveillance is based on OpenCV programming under GNU-Linux supported by socket programming. The objectives of traffic surveillance system are tracking moving vehicles, counting them and detecting the abnormal movement in specific places. This is done by analyzing the frames captured and processing them using OpenCV functions. The use of open source resources such as OpenCV functions in GNU- Linux provides an easy method to use the computer vision framework besides it can run vision code in real time. The traffic surveillance works in real-time process for (5-33 fps) of video stream in a rainy day and a sunny day. Traffic surveillance system accepts video images either from the camera captures or from the files. It tracks the moving vehicles using image processing and recognition algorithms such as converting to Gray scale and Mixture of Gaussian method; it marks each moving vehicle with a rectangle box and counts them. The system consists of: capturing part, tracking part, detecting abnormal movement from specific points and streaming the view to the other connecting computer. Besides it could minimize the frame processing in the case of the frame capture with high resolution. The system has successfully been tested in three different processor speeds which are 1.2 GHz CPU, 2.0 GHz CPU and 2.6 GHz CPU. The results are quite accurate and are measured in mille second. The system works in outdoor environment which is complex with wavering tree branches and flow of rain. Besides it updates frame by frame in any module of background views especially from the upper view of the road in order to get a significant images of all the objects. The results could be recorded by saving the requested group of photos in any image format. The CPU processing speed with the frame size process represents one of the key factors of the performance analysis of traffic surveillance. Processing speed comparison of the processing steps results in different resolutions for frame size made. And, a significant result is found by reducing the calculation time of processing when the frames size captured are reduced. The experiment was done by using 9 different views each with 700 frames consist of different views of moving vehicles in sunny or rainy situations on the roads of Perlis state. These experiments successfully detected the movements of all types of vehicles.

CHAPTER 1

INTRODUCTION

1.1 Introduction

Recent developments in electronic imaging technology mostly are improvement of image processing, incremental of its storage capacity and also transmission enhancement of color images and video details. Digital images by its nature from beginning are tremendously promising a lot of ambitious DreamWorks and possibly be implemented in parallel with rapid hardware revolution progress along the time.

Image Processing represents the heart of any image technology. Applications of image processing are primarily in the areas of recognition, transmission and storage of information. Image transmission applications are in television broadcasting, remote sensing via satellite, military communication via aircraft, radar and sonar teleconferencing, computer communications, facsimile transmission, etc. Image storage is required for educational and business documents, such as storage of medical image from the Computer Tomography (CT) (Farrokhi, F., 1997), Magnetic Resonance Imaging (MRI) (Delaney, 1995) and digital radiology, motion pictures, satellite images, weathers maps and geological surveys. The image recognition applications are important and found in medicine, surveillance systems, personal digital assistant (PDA), and road monitoring systems. Application of image recognitions is also possible in the processing of algorithms where the number of operations required to implement an algorithm are reduced by working with the recognized data.

The field of image processing is a wide subject. It detects objects, removes noise from images and, does contrast analyses to the objects. All these things are possible by using special techniques and methods on images to be processed. The results could be used for other scientific applications or for other benefits.

These special techniques and methods in processing images may use various codes in coding programming of the applications. Different coding techniques are used for different applications. These codes could be programmed by using C/C++ programming language, MATLAB, Visual Basic, Visual C/C++, Borland C, OpenCV as library of C and Python as recent popular programming languages. The programs can be installed in several kinds of operating systems such as Microsoft Windows operating systems (Windows95/ NT/ 2000/ XP/ Melenium / Vista / 7) beside that it could be installed in Linux Operating systems such as: (RedHat/Debian/Mandriva/Fedora/Ubuntu/Kubuntu,). These programs could be used for real time execution of the application and to operate devices such as smart digital cameras and any mobiles system..

The use of Edge Detection algorithm is the most common direct approach for detecting and discontinuities that could highlight object-boundary information in a digital image. Edge detection must be efficient and reliable since it is crucial in determining how successful subsequent processing stages will be. In order to fulfill the reliability requirement of edge detection, a great diversity of operators has been devised with various differences in their mathematical and algorithmic properties (Tania Stathaki, 2008).

Several approaches to edge detection focus their analysis on the identification of the best differential operator necessary to localize sharp changes of the image intensity. These approaches recognize the necessity of a preliminary filtering step, as a smoothing

stage. The most widely used smoothing filter is the Gaussian one which plays an important role in detecting edges. Canny's approach (J.F. Canny, 1986) is a standard technique in edge detection. This scheme, in substance, identifies edges in the image as the local maxima of the convolution of the image with an 'optimal' operator. The operator's optimality is subject to three performance criteria as defined by Canny and is very close approximation to the first derivative of the Gaussian function. After this process, candidate edge pixels are identified as the pixels that survive an additional thinning process known as non-maximal suppression (J.Canny , 1985). Then the candidate edges are thresholded to keep only the significant ones. Canny suggests hysteresis threshold to eliminate streaking of edge contours (R. Deriche ,1987;Tania Stathaki 2008).

1.2 Overview of video surveillance system

Image processing is used for intelligent Transport systems in the roads, which recently have been widely developed. There has been much research and implementation using various image sensors to get information for traffic and vehicle control. The image seen from a camera is located on the road can be used for vehicle detection, such as velocity of a car or car group measurement and parking of vehicles. But the image seen from a camera located in the vehicle can be used for preceding car detection such as measurement of distance to the preceding car, detection of lane and obstacle. Shinji Ozawa 1999 did a study for image processing for vehicles on the road using some sensing techniques in vehicle and traffic control, Ozawa applies some procedures to a sensor in Intelligent Transport Systems (ITS) whereby he could apply the image processing technique to find: (Kiyoharu Aizawa, 2004; Shinji Ozawa, 1999).

- To a moving car.
- To locate traffic congestion due to accident .
- To count cars in parking lot by tracking.
- To measure travel time.

1.3 Problem Statement

Many of road events are not accidents at all. They are events arising from conflicts contrived by the rules and design of the road and users, A report published by the World Health Organization in 2004 estimated that some 1.2 million people were killed and 50 million injured on the roads around the world each year (World Health Organization, 2010) and was the leading cause of death among children 10 – 19 years of age. The report also noted that the problem was most severe in developing countries and that simple prevention measures could reduce to half the number of deaths. (*BBC News* 2008). There are many reasons causing accidents to happen such as the velocity of the cars, crossing the traffic lights and turnings abnormally. The development of the traffic surveillance systems works to reduce some of these accidents. Therefore the use of cameras with video system technology will help to prevent accident from happening. The Intelligent traffic surveillance system, in this thesis, works to detect the abnormal turning of the vehicles in specific areas, and also counts the number of moving vehicles to reduce the traffic jam, besides that the view could be streamed to the base station or recorded as images and sends the important one to the base station.

1.4 Objectives

The development of road surveillance systems with multifunctional techniques has received increasing attention. The use of the web camera is one solution to solve the traffic problems. This work discuss about the camera-video-surveillance capabilities of tracking across different and varied road environments including detection of moving vehicles. This effort has done by developing image processing and recognition Using OpenCV to:

- Developed a prototype video imaging techniques using open source system.
- Design road surveillance systems for all weather condition.
- Detecting abnormal movement of vehicles from specific points on the road.
- Allow the capture images of abnormal movement to be recorded.
- Comparison the capability of saving images in different formats.

1.5 Thesis layout

This thesis includes five chapters which deals with scientific purposes needed to develop the recognition method. Beside Chapter 1, Chapter 2 deals with Literature review of Image Processing and recognition in traffic surveillance. Chapter 3 presents the methodology of algorithms and the software techniques used in traffic surveillance system and the processing steps of the image processing.

Chapter 4 discusses results and performance analysis of the tracking and movement recognition method developed in chapter four and tested it with different processor speeds. Beside it contains the performance evaluation from preparing

comparisons between the results of experimental that has been done, using different processor speeds and a comparison of saving the images in different formats. Finally Chapter 5 concludes the thesis by demonstrating the success present ideas and its requirements of road traffic surveillance application. And also some suggestion has been discussed as a future work.

© This item is protected by original copyright

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction:

The development of a traffic surveillance system is a main problem to the most of researchers and people around the world. The use of image processing techniques is useful for the intelligent detection of traffic and transport systems which recently have been widely developed. There has been much implementations and research using image sensors to get information and data for traffic control and vehicle control. Cameras could be used for vehicle tracking, measurement of the speed of the car and detection of perked car, etc. (Shinji Ozawa, 1999)

2.2 Fundamental of Image Processing and Its Applications

The interest in images and how to process them started a long time ago, most researchers and scientists were concerned about how to get a high quality of the photograph. Most of the previous work is in distributing the level of brightness and using lighting photograph technique to make images in gray scale better than what it was previously. The first computer was powerful enough to carryout meaningful image processing tasks which appeared in the early 1960s. The Birth of what is called digital image processing today can be traced to the availability of machines and the onset of the space program during that period. It took the combination of those developments to bring into focus the potential of digital image processing concept which work on using computer techniques for improving images from the space orbit programs at the same