OPTIMIZATION OF PROCESSING TIME FOR VISION BASED ONE-CHANNEL AROUND VIEW MONITORING SYSTEM ON EMBEDDED PLATFORM

BY

NURUL HIDAYAH BINTI MAHAMUD

A thesis submitted in fulfilment of the requirement for the degree of Master of Science (Mechatronics Engineering)

Kulliyyah of Engineering
International Islamic University Malaysia

MAY 2021

ABSTRACT

Around View Monitoring (AVM) system is widely used as a part of driver assistance system. AVM helps driver to see blind-spot region and drive through narrow space by providing a 360-degree sight of the car's surrounding from the top view. The project will implement a vision based one-channel AVM system on embedded board and focused on the algorithm optimization to improve the system's processing speed.

Two embedded boards, Telechip TCC 8971 SoC (Cortex-A7 quad-core) and Renesas R-Car H2 (Cortex-A15 quad-core) are used for this project and a PC platform of Intel i7 core is used to benchmark the processing speed of running the same algorithm on PC. Embedded board has a limited processing power and memory compared to standard PC platform. The common challenges of running an image processing on embedded board is that image processing requires high processing power and large memory to support pixel by pixel computation.

The research objective focused on identifying the issues of running the AVM system on embedded board and implements several optimization methods to improve the initial processing time of the system. The thesis explains the use of Yocto Project to setup the embedded development board and algorithm porting. Several techniques of optimizations are presented in this thesis including multi-threading and the use of Qt OpenGL for GPU-accelerated output image rendering. The final optimization improved the total processing time of AVM system by 54.29 % on Telechip board, 52.04 % on Renesas board and 61.64% in PC platform.

خلاصة البحث

يستخدم نظام مراقبة الرؤية (AVM) على نطاق واسع كجزء من نظام مساعدة السائق. يساعد هذا النظام (AVM) السائق على رؤية منطقة النقطة العمياء والقيادة عبر المساحة الضيقة من خلال توفير رؤية بزاوية 360 درجة لمحيط السيارة من المنظر العلوي. سيقوم هذا المشروع بتنفيذ نظام (AVM) أحادي القناة للرؤية على لوحة مدمجة ويركز على تحسين الخوارزمية لتحسين سرعة معالجة النظام.

يتم استخدام لوحتين مدمجتين هما:(رباعي النواة Telechip TCC 8971 SoC (Cortex-a7 و (رباعي النواة Renesas R-Car H2 (cortex- a15 لهذا المشروع ، ويتم استخدام منصة الكمبيوتر الشخصي Intel i7 الأساسية لقياس سرعة المعالجة لتشغيل نفس الخوارزمية على جهاز الكمبيوتر. تتمتع اللوحة المدمجة بقوة معالجة وذاكرة محدودة مقارنة بمنصة الكمبيوتر القياسية. تتمثل التحديات الشائعة لتشغيل معالجة الصور على لوحة مدمجة في حين أن معالجة الصور تتطلب قوة عالية للمعالجة وذاكرة كبيرة لدعم حساب البيكسل بواسطة البيكسل.

ركز هدف البحث على تحديد تشغيل نظام AVM على لوحة مدمجة وتنفيذ العديد من أساليب وطرق التطوير لتحسين وقت المعالجة الأولي. تشرح هذه الأطروحة استخدام مشروع Yocto لإعداد لوحة مدمجة متطورة ومنفذ خوارزمية. تمت مناقشة العديد من تقنيات التحسين في هذه الأطروحة، بما في ذلك خيوط المعالجة المتعددة واستخدام Qt Open GL لعرض الصورة المخرجة بسرعة GPU. زاد التحسين النهائي من وقت المعالجة الكلي لنظام AVM بنسبة 54.29٪ على لوحة Telechip و52.04 ٪ على لوحة Renesas و61.64 ٪ في منصة الكمبيوتر الشخصي.

APPROVAL PAGE

I certify that I have supervised and read this study and that in my opinion, it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Science (Mechatronics Engineering).

……..
Zulkifli Zainal Abidin
Supervisor

……..
Yasir Mohd. Mustafah
Co-Supervisor

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master of Science (Mechatronics Engineering).

……..
Hazlina Md Yusof
Internal Examiner

……..
Zool Hilmi Ismail
External Examiner

This thesis was submitted to the Department of Mechatronics Engineering and is accepted as a fulfilment of the requirement for the degree of Master of Science (Mechatronics Engineering).

……..
Ali Sophian
Head, Department of Mechatronics Engineering

This thesis was submitted to the Kulliyyah of Engineering. and is accepted as a fulfilment of the requirement for the degree of Master of Science (Mechatronics Engineering).

……..
Sany Izan Ihsan
Dean, Kulliyyah of Engineering

DECLARATION

I hereby declare that this thesis is the result of my own investigations, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

Nurul Hidayah binti Mahamud

Signature ..... Date 21 September 2021

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

OPTIMIZATION OF PROCESSING TIME FOR VISION BASED ONE-CHANNEL AROUND VIEW MONITORING SYSTEM ON EMBEDDED PLATFORM

I declare that the copyright holders of this thesis are jointly owned by the student and IIUM.

By signing this form, I acknowledged that I have read and understand the IIUM Intellectual Property Right and Commercialization policy.

Affirmed by Nurul Hidayah binti Mahamud

…….. Signature

21 September 2021 Date

ACKNOWLEDGEMENTS

IN THE NAME OF ALLAH, THE MOST COMPASSIONATE AND THE MOST MERCIFUL
I thank Allah S.W.T, Sustainer and the Creator of the universe and its inhabitant, for He has given me HIS blessings and making it possible for me to complete this thesis.

I would like to take the opportunity to express my appreciations to my supervisor, Assoc. Prof. Dr. Zulkifli Zainal Abidin and my co-supervisor, Assoc. Prof. Dr. Yasir Mohd Mustafah for their teaching, support and guidance in helping me to complete this research.

I also would like to express my gratitude to Centre for Unmanned Technologies (CUTe) lab for providing comfortable workspace and equipment needed for this research. I also would like to thank all the lab members for their continues support and assistance.

Finally, I would like to express my gratitude to my parents and family for their support and patience, believing in me to complete my research.

Finally, I have learnt and experienced a lot of things while working for this collaboration research. This research helps me to require new skills and knowledge from lecturers, lab members and company staff which will be beneficial for my future engineering practices.

This research is supported by International Islamic University Malaysia (IIUM) with Collaborative Research in Engineering, Science & Technology Center (CREST) and Delloyd R&D (M) Sdn. Bhd. (Project ID P11C2-17 & SP17-029-0291)

TABLE OF CONTENTS

Abstract ii
Abstract in Arabic iii
Approval Page iv
Declaration v
Copyright vi
Acknowledgements vii
List of Tables x
List of Figures xi
List of Abbreviations xiii

CHAPTER ONE: INTRODUCTION 1
1.1 Background 1
1.2 Problem Statement and Significance 4
1.3 Research Objectives 5
1.4 Research Methodology 5
1.5 Research Scope 8
1.6 Report Organization 9

CHAPTER TWO: LITERATURE REVIEW 11
2.1 Introduction 11
2.2 Compiler Optimization 13
2.2.1 Function Inlining 13
2.2.2 Eliminating Common Sub-Expression 14
2.2.3 Loop Unrolling 14
2.2.4 GCC Optimization Techniques 15
2.3 Source Code Modification 16
2.3.1 Loop Termination 16
2.3.2 Loop Fusion 17
2.3.3 Variable Selection 18
2.3.4 Pointer Aliasing 18
2.4 Single Rate Data Flow 19
2.5 Graphics Processing Unit 21
2.5.1 OpenGL ES 22
2.5.2 OpenCL 23
2.5.3 CUDA 24
2.6 Other Optimization Methods 25
2.7 Image Processing using OpenCV 27
2.8 Yocto Project 28
2.9 Chapter Summary 30

CHAPTER THREE: SYSTEM DEVELOPMENT 34
3.1 Introduction 34
3.2 Platform Specification 34

3.2.1 Renesas R-Car H2 (Stout) Development Board 34
3.2.2 Telechip TCC 8971 SoC 35
3.2.3 PC Platform 37
3.3 Embedded Board Setup using Yocto Project 38
3.4 AVM System on Embedded Platform 40
3.4.1 AVM Algorithm Overview 40
3.4.2 Porting AVM Algorithm into Embedded Board 46
3.5 Experiment Overview 48
3.6 Determine Program Hotspot 48
3.7 Code Optimization 50
3.7.1 Initial Optimization 50
3.7.2 Multi-Threading 51
3.7.3 OpenCV Template Matching 55
3.7.4 Display using Qt OpenGL 57
3.8 Chapter Summary 58

CHAPTER FOUR: RESULT AND DISCUSSION 59
4.1 Introduction 59
4.2 Determine Program Hotspot 59
4.3 Result of Code Optimization Running on Different Platforms 60
4.3.1 Initial Optimization 60
4.3.2 Multi-Threading 62
4.3.3 OpenCV Template Matching 64
4.3.4 Display using Qt OpenGL 66
4.4 Optimizing AVM System on Embedded Platform 67
4.5 Chapter Summary 68

CHAPTER FIVE: CONCLUSION 69
5.1 Summary 69
5.2 Thesis Contribution 71
5.3 Recommendation 71

REFERENCES 72

LIST OF TABLES

Table 2.1	SRDF Task Description and Their Details	19
Table 2.2	Task Execution Result on LDWS Algorithm using SRDF Model	20
Table 2.3	Optimization methods summary	31
Table 3.1	Specifications summary of the platform for running AVM system	37
Table 4.1	Processing time of motion estimation task after initial optimization	61
Table 4.2	Processing time of process for display task after initial optimization	61
Table 4.3	Processing time of motion estimation task with multi-threading	62
Table 4.4	Processing time of motion estimation task with multi-threading	63
Table 4.5	Processing time of Motion estimation using OpenCV template matching	65
Table 4.6	Average processing time taken for output image display by using OpenGL	66
Table 4.7	Summary of processing speed of AVM application after optimized	67

LIST OF FIGURES

Figure 1.1	General view of ADAS applications with its typical sensors	1
Figure 1.2	(a) 3D surround view in the BMW X3 (b) Nissan's Around View Monitor System	3
Figure 1.3	Research methodology flowchart	6
Figure 2.1	Categories of embedded optimization method	13
Figure 2.2	(a) Original expression (b) Eliminating common sub-expression	14
Figure 2.3	(a) Conventional loop (b) Partial loop unrolling	15
Figure 2.4	(a) Loop with increasing counter (b) Loop termination method	17
Figure 2.5	(a) Two loops with same iteration (b) Loop fusion method	17
Figure 2.6	Example application of the restrict function	18
Figure 2.7	LDWS algorithm modeled as SDRF	20
Figure 2.8	Yocto Project development environment	29
Figure 3.1	Renesas R-Car H2 system block diagram	35
Figure 3.2	TCC8971 functional block diagram	36
Figure 3.3	Renesas R-Car H2 running Linux-yocto OS	39
Figure 3.4	General AVM flowchart	41
Figure 3.5	AVM Application GUI	41
Figure 3.6	Image transformation from (a) fisheye image (b) undistorted image (c) top down image	42
Figure 3.7	Searching region set from top down image for block matching algorithm	43
Figure 3.8	Improved searching region for block matching algorithm	43
Figure 3.9	Block matching algorithm concept illustration	44
Figure 3.10	Block matching algorithm flowchart	45
Figure 3.11	Final AVM system display	46
Figure 3.12	Running AVM application on embedded board (a) Renesas R-Car H2	

	(b) Telechip TCC 8971 board	47
Figure 3.13	Using std::chrono on searchTemplate() function	49
Figure 3.14	Using multi-threading for block matching algorithm	52
Figure 3.15	(a) Left searching region for block matching algorithm running on new thread (b) right searching region for block matching algorithm running on main thread	52
Figure 3.16	General flowchart of block matching algorithm using *cv::parallel_for_* function	55
Figure 4.1	Processing time of AVM system on three different platform	60
Figure 4.2	AVM application showing image stitching result based on different OpenCV template matching methods	64

LIST OF ABBREVIATIONS

ADAS	Advanced Driver Assistance System
API	Application Programming Interface
AVM	Around View Monitoring
BSP	Board Support Package
FCWS	Forward Collision Warning System
GCC	GNU Compiler Collection
GPU	Graphics Processing Unit
GUI	Graphical User Interface
LDWS	Lane Departure Warning System
PSNR	Peak Signal to Noise Ratio
RISC	Reduced Instruction Set Computer
ROI	Region of Interest
SDK	Software Development Kit
SoC	System on Chip
SRDF	Single Rate Data Flow
TBB	Threading Building Block

CHAPTER ONE
INTRODUCTION

1.1 BACKGROUND

Advanced Driver Assistance System (ADAS) is an intelligent, in-vehicle safety system that aims at improving on-road safety. General technology of ADAS uses sensors such as RADAR (Radio Detection and Ranging), LIDAR (Light Detection and Ranging), ultrasonic and front camera (vision-based) to sense the surrounding environment and give alerts to the drivers on the potentially dangerous traffic situation. Figure 1.1 shows an overview of ADAS applications and typical sensors used for each application.

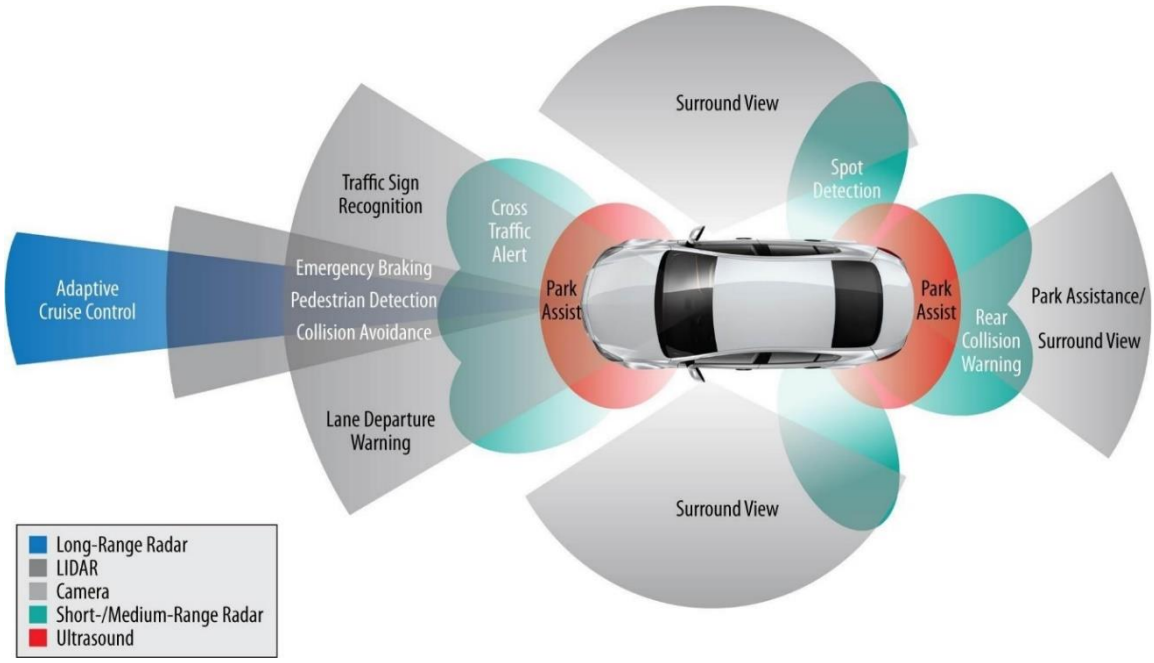

Figure 1.1 General view of ADAS applications with its typical sensors (Texas Instruments, TI Vision SDK, Optimized Vision Libraries for ADAS Systems, n.d.)

Researchers have successfully developed and implemented ADAS in several in-vehicle applications (Texas Instruments, n.d.). Some well-known ADAS applications such as:

(1) *Lane Departure Warning System (LDWS):* The system will alert the driver if the car's relative location is drifting out of the lane marking.

(2) *Forward Collision Warning System (FCWS):* The system will detect all possible hazardous in front of the car and gives alerts if the car's relative position is risky for an accident.

(3) *Pedestrian Detection System*: The system detects pedestrians and gives warning to the driver for any potential danger.

(4) *Traffic Sign Recognition*: The system detects all the traffic signs and passes the information to the driver.

(5) *Around View Monitoring (AVM)*: The system provides a 360-degree view of the vehicle for the driver to fully aware of the surrounding.

a) Introduction to AVM

One of the ADAS features is the Around View Monitoring or AVM system. According to Yu and Ma (2015), AVM helps drivers to see the blind-spot region and drive through narrow space by providing a 360-degree sight of the vehicle's surrounding from the top view. AVM also is very helpful to support drivers during parking. Some system even offers audio and visual warning technique to alert the driver if it senses any movement around the vehicle. Some system provides an interactive 3-dimensional display of the vehicle and its surrounding. Generally, AVM product uses four to six fisheye cameras installed around the vehicle. Figure 1.2 shows the existing AVM product in the market.

(a)

(b)

Figure 1.2 (a) 3D surround view in the BMW X3 (1001cars, 2017) and (b) Nissan's Around View Monitor System (Nissan, 2017)

b) Introduction to Embedded System

An Embedded system basically can be defined as a microprocessor-based system designed to run a specific task. The embedded system is not as flexible as a PC which the functions can be programmed or changed by end-users (ARM, 2014). has summarized several common types of embedded system in his thesis which includes

general-purpose CPU, microcontroller, digital signal processor (DSP), and programmable logic devices such as complex programmable logic device (CPLD) and field-programmable gate array (FPGA)).

1.2 PROBLEM STATEMENT AND SIGNIFICANCE

The technology of ADAS is widely used in which drivers are taking particular interest and understand the importance of road safety. For ADAS, it is important for the system to efficiently execute its function because it is related to the safety of drivers. Thus, it is importance to develop a reliable and fast system to be implemented in vehicles.

The development of ADAS technology on embedded platform should comply with several requirements. The algorithm processing and execution requires considerable computational power and should be robust enough to run on an embedded platform and various lighting conditions. Practically, the performance should be fast enough to be implemented in a real environment. Most of the systems are developed on the PC platform, but sometimes the performance and speed degraded when the algorithm is ported on an embedded platform. Thus, vision-based ADAS faces challenges in developing a robust algorithm and at the same time is able to run efficiently on an embedded platform (Nieto, Otaegui, Vélez, Ortega, & Cortés, 2013).

Vision-based ADAS requires high computational power for image processing. This usually causes some time delay when the system needs to process the data frame-by-frame. The embedded system itself provides limited processing power.

The project focused on implementing an AVM system with a graphical user interface (GUI) using one camera input (or one-channel AVM). The performance of the system will be analysed and optimized to be able to run on embedded platforms with acceptable speed.

1.3 RESEARCH OBJECTIVES

The objectives of the research are:

1. To identify the issues of the developed AVM system affecting the system's processing speed
2. To determine optimization techniques for improving the computational time of the image processing algorithm in a developed AVM
3. To evaluate the performance and processing time of optimized AVM algorithm running on embedded platforms

1.4 RESEARCH METHODOLOGY

The project focused on improving the processing time of an AVM system running on embedded platform. The research methodology is as described in Figure 1.3.

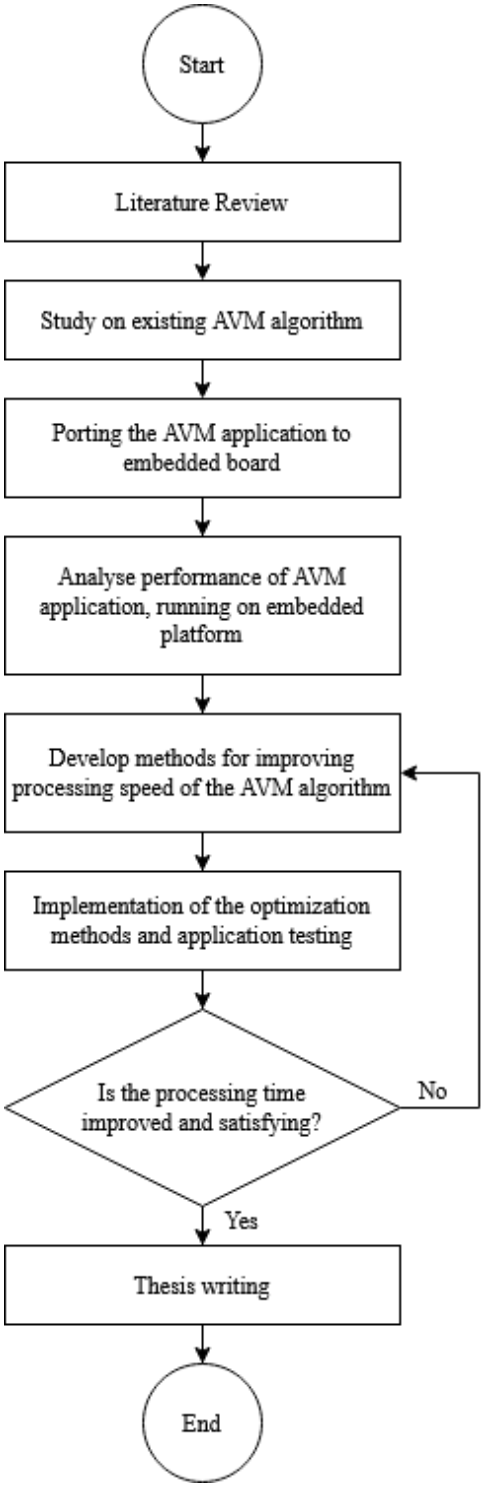

Figure 1.3: Research methodology flowchart

The research methodology above can be summarised in four phases which are (1) literature review and algorithm study, (2) data collection, (3) optimization of AVM application, and (4) documentation.

Phase 1: Literature review and algorithm study

- Literature review
    - Study on existing optimization techniques for improving the processing speed of image processing algorithm running on an embedded platform.
- Study on the AVM algorithm
    - Study and understand the algorithm flow of the one-channel, vision-based AVM system.

Phase 2: Data collection

- Porting the AVM application to embedded board.
    - The application is deployed on two embedded boards, (1) Renesas R-Car H2 development board and (2) *Telechips* - TCC8971 SoC custom development board.
    - The procedure includes board setup by developing the operating system (OS) for the embedded board and cross-compiling the AVM application.
    - The application is also tested on a PC platform for performance comparison.
- Analyze the performance of the AVM application running on embedded platform.
    - The AVM algorithm is run for several seconds and output data is recorded.
    - The initial processing time and frame-per-second (fps) data is calculated and tabulated as benchmarking result.

Phase 3: Optimization of AVM application

- Develop methods for improving the processing speed of the AVM application with GUI display.
- Implementation of the optimization methods and application testing
    - Several optimization methods are applied on the AVM application and the result are recorded.
    - The optimized algorithm will be tested for reliability and robustness.
    - The application is debugged and modified until a stable and satisfactory result is obtained. The final application should run with a frame rate of more than 10 fps.

Phase 4: Documentation

- Project documentation and thesis writing

1.5 RESEARCH SCOPE

The project focus on analysing the processing time of an AVM application which is a part of ADAS. The scope of this project is the optimization of the AVM application on an embedded board. The project will cover several optimization methods including optimization through source code/algorithm modification and GPU using the available framework. For the GPU support, an early review has found that both embedded boards do not support OpenCL and CUDA which limits the GPU framework that can be applied. Next, the AVM application and optimization will be done using input from video recorded at the project setup. The recorded video is obtained using a toy car with a camera attached to a rear body to simulate reverse parking action. The car is pushed with constant speed in one direction and the input image is

recorded. The project testing also will be done indoors in the lab with a homogeneous surface-type floor. Besides that, the input will not be affected by any weather change. The embedded board used is of automotive standard and provided by the sponsoring company.

1.6 REPORT ORGANIZATION

This research proposal is divided into several chapters.

Chapter 1: Introduction

This chapter will discuss the overview of the project including research objectives, problem statements, and methodology for this vision-based optimization on an embedded platform.

Chapter 2: Literature Review

This chapter will review several studies on optimization methods for embedded systema. The optimization methods including compiler optimization, source code/ algorithm-based optimization and, hardware-accelerated algorithm using available GPU framework supported. The review will help to investigate existing optimization techniques available for embedded platforms. Analysis of the advantages and disadvantages of each method will also be discussed in this chapter.

Chapter 3: System Development

This chapter will discuss the details of the embedded platform and experiment setup. This chapter will also discuss several optimization techniques implemented to optimize the processing speed of the AVM system running on embedded platforms.

Chapter 4: Result and Discussion

This chapter will present the resulted processing time after implementing optimization methods on the AVM system. The results will be analysed and discussed further in this chapter.

Chapter 5: Conclusion

This chapter will discuss the summary of this project, the thesis contribution, and several recommendations to further improve the current research.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

The technology of ADAS is not something new. The development of ADAS technology starts with the introduction of the Anti-lock Braking System (ABS) as a product for vehicle safety in the late 70s of the twentieth century (Ziebinski, Cupek, Erdogan, & Waechter, 2016). Ziebinski has summarized many other ADAS technologies in general in his research paper. This proves that the development of ADAS offers a great possibility for improving on-road and transport safety. Many researchers worldwide are working to further improve this technology.

The ADAS acts as a passive system (e.g. warning system) or even an active system (e.g. automatically assist vehicle control). Some of these systems use camera which run image processing algorithms to detect the obstacle in front of the car or sense the surrounding environment (Saussard, Bouzid, Vasiliu, & Reynaud, 2015). Besides the vision-based system using a camera, several systems also use radar or LIDAR for their input data. Existing ADAS applications include *Front Collision Warning System*, *Lane Departure Warning System*, *Driver Drowsiness Detection*, *Pedestrian Detection,* and *Traffic Sign Recognition* (Saussard, Bouzid, Vasiliu, & Reynaud, 2015; Hammond, Qu, & Rawashdeh, 2015).

An embedded system should satisfy a trade-off between several requirements such as real-time performance, cost, power consumption, spatial constraints, reliability, and limited resources (Velez & Otaegui, 2016; Ferreau, et al., 2017). When implementing the system on an embedded platform, the real-time performance of the system is one of the important issues, to fulfil the system demand. The system should be able to process the data and run the algorithm