DESIGN AND SIMULATION OF COLLISION WARNING SYSTEM FOR ROAD SAFETY IMPROVEMENT

TRAN DUC CHUNG

ELECTRICAL & ELECTRONIC ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
DECEMBER 2013

DESIGN AND SIMULATION OF COLLISION WARNING SYSTEM FOR ROAD SAFETY IMPROVEMENT

By

TRAN DUC CHUNG

FINAL PROJECT REPORT

Submitted to the Electrical & Electronic Engineering Department

in Partial Fulfillment of the Requirements for the

Bachelor of Engineering (Honors) - Electrical & Electronic Engineering - Degree

Universiti Teknologi Petronas

31750, Tronoh, Perak, Malaysia

CERTIFICATION OF APPROVAL

DESIGN AND SIMULATION OF COLLISION WARNING SYSTEM FOR ROAD SAFETY IMPROVEMENT

By

TRAN DUC CHUNG

A project dissertation submitted to the

Electrical & Electronic Engineering Department

in Partial Fulfillment of the Requirements for the

Bachelor of Engineering (Honors) - Electrical & Electronic Engineering - Degree

Approved by:

Dr. Likun Xia

Project Supervisor

Universiti Teknologi Petronas

31750, Tronoh, Perak, Malaysia

CERTIFICATION OF ORINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Tran Duc Chung

ABSTRACT

Collision warning (CW) systems, as part of automated emergency braking (AEB) systems, become more and more important in modern vehicles for assisting drivers during emergency driving situations. Most of AEB systems use sensor fusion system (camera and radar) or binocular vision based (stereo-vision) system for vehicle detection. The sensor fusion system and binocular vision based system results in more complicated algorithm and adds significant cost to CW systems. Hence the purpose of this project is to design and simulate a monocular vision based CW system in MATLAB environment. One genetic automobile detection algorithm utilizing an inverted U-shape back wheel structure of preceding vehicle is implemented in the CW system. The algorithm achieves 49% accurate detection results of multiple preceding four-wheel vehicles with processing speed of 4.90 frames per second (fps) for 1080x1920 pixels (1080p) image sequence. The CW system is able to generate single level warning signal if there is potential collision between host vehicle and preceding vehicle. At the end of this project, one review paper titled "A Review of Automated Emergency Braking System and the Trending for Future Vehicles" has been submitted and presented in "SAE 2013: Southeast Asia Safer Mobility Symposium". Additionally, one research paper titled: "An Automobile Detection Algorithm Development for Automated Emergency Braking System" has been submitted to the 51st Design Automation Conference and got accepted for review.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Explanation
1080p	1080x1920 pixels
AEB	Automated emergency braking
ACC	Adaptive cruise control
CW	Collision warning
CWAB-PD	Collision warning with full auto brake and pedestrian detection
Fps	Frames per second
FOV	Field of view
HFOV	Horizontal field of view
MIROS	Malaysian Institute of Road Safety
VFOV	Vertical field of view

ACKNOWLEDGEMENT

The preparation of this report is based on accumulated information by the student throughout the May 2013 and September 2013 trimesters under supervision of Dr. Likun Xia, Centre of Intelligent Signal and Imaging Research (CISIR), Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS (UTP).

The author would thank Dr. Likun Xia, final year project supervisor, and Dr. Mohd Zuki B Yusoff, final year project 1's internal examiner, from Department of Electrical and Electronic Engineering, UTP, for their assistances and comments on the project throughout the period of development. Additionally, the author would thank UTP and Malaysian Institute of Road Safety (MIROS) for their supports in the development of this project.

CHAPTER 1: INTRODUCTION

1.1.Background of Study

Every year, many road accidents (i.e. 414,421) take place resulting in deaths (i.e. 6,872), injuries (i.e. 21,397) and economic loss (i.e. involving of 760,433 vehicles) in Malaysia [1]. Among several causes of accidents, adverse weather and human error are considered as main causes [2]. Out of 9.18% of accidents happened due to adverse weather, up to 7.12% of accidents are caused by rainy weather [3]. About 94% of accidents are caused by no deceleration or insufficient deceleration of vehicles [4]. Because natural factor is difficult to predict and is unavoidable, it is important to develop CW system, as part of AEB system, to provide pre-collision warning to drivers to avoid or reduce fatal accidents.

Since early 2000s [2], AEB systems have been developed by many automakers to assist drivers in braking maneuver during emergency cases. Commercialized AEB systems do exist in the global market and have been integrated mainly into high-end vehicles i.e. “collision warning with full auto brake and pedestrian detection” (CWAB-PD) [2, 5, 6]. Most of AEB systems in high-end vehicles use sensor fusion system (radar and camera) or binocular vision (stereo-vision) based system (two cameras) for object detection [7]. In consideration of Malaysian car prices in [8-11], the AEB systems using radar are excessively costly due to high cost of devices [12, 13]. Therefore AEB systems using sensor fusion system is not suitable for Malaysian cars. For AEB systems using stereo-vision system, the detection’s accuracy strongly depends on calibration of the cameras [14]. Based on these reasons, monocular vision based CW system, as part of AEB system, using one camera for object detection is developed in this final year project.

1.2.Study Objective

- To design and simulate a monocular vision based CW system.

1.3.Study Scope

An intensive study of available AEB systems in the market is performed. A monocular vision based CW system is designed and developed in MATLAB environment.

1.4.Problem Statement

Monocular vision based CW system is not yet developed in recent researches. This addresses an area for development of monocular vision based CW system.

1.5.Solution Proposal

To design a monocular vision based CW system using one camera.

CHAPTER 2: LITERATURE REVIEW

2.1.Car Accident Statistics & the Need of AEB System

Car accidents happen everywhere in the world. Up to 75% of all car accidents occur at low speed i.e. less than 20 mph (about 32 kph) [15]. There are three main causes leading to car accidents [2]:

- sudden change in road direction at "sharp blends and angles";
- bad weathers (heavy rain, dusty road, and dense fog), slippery roads, natural disasters (earthquake, tsunami);
- careless or fatigued drivers.

The first cause could be addressed while the second cause is a natural cause and difficult to avoid. The third cause is due to human error and can be avoided. Less attention on road while driving is categorized as a human error that leads to insufficient brake force creating low deceleration rate: 21% has no deceleration, 73% decelerates less than 5 m/s^2 [4]. This type of human error can be prevented using pre-collision warning system.

Compensation for traffic accidents, including car accidents, is very costly in terms of traffic delay hours leading to significant fuel waste and "congestion cost" i.e. 4.2 billion hours, 2.9 billion gallons and $80 billion in US in 2007 respectively [2].

Therefore AEB system ("automatic emergency braking system" or "autonomous emergency braking system" [4, 5, 16]) needs to be developed to assist driver with braking maneuver. This is to avoid collision, reduce major and fatal injuries to passengers and collision aftereffect such as traffic congestion.

2.2.Review of Existing AEB & Supplementary Systems

2.2.1. AEB Overview

Many automobile makers have introduced such systems to their high-end vehicles since 2003 however most of the systems use sensor fusion system or binocular vision based system instead of monocular vision based system for preceding vehicle detection [2]. Typically an AEB system consists of:

- a set of sensors installed on a host vehicle measuring: its velocity, acceleration, distance to other movable/stable object while on the road such as human, other vehicle, big rock;

- a brake control mechanism: to control the vehicle's brake automatically by electronic controller(s);
- controller(s): to process signals fed from the sensors, calculate and estimate potential of collision with the detected objects, provide warning signals to driver and apply force to brake pedal to reduce vehicle's speed if the collision potential exceeds a threshold which might cause major or fatal injuries to the driver and/or passenger(s).

Controller is considered as the most important part and plays critical role in AEB system. Different manufacturers develop different AEB systems and currently there is no common standard for AEB systems yet. Table 1 lists several AEB systems and Pre-AEB systems of existing automobile manufacturers. The term "Pre-AEB systems" refers to the CW systems such as Lane Departure Warning System of Peugeot. It helps driver by providing warning signals when certain potential danger exists while driving. This type of system could be expected to be further developed and enhanced to become AEB system in the future.

Table 1: Several AEB Systems & Supplementary Technologies [17, 18]

Brand	AEB Systems & Supplementary Technologies
Audi	Pre-Sense Basic Pre-Sense Front Pre-Sense Front Plus Adaptive Cruise Control (ACC) with Stop & Go function Audi Side Assist + Pre-Sense Rear Active Lane Assist Night Vision Assistant
BMW	iBrake 3 Driving Assistant + Lane Departure Warning & Collision Warning Active Protection
Ford	Active City Stop
	Forward Collision Warning with Brake Support
Mercedes-Benz	Pre-Safe Brake
	Collision Prevention Assist
	Distronic Plus
Peugeot	LDWS - Lane Departure Warning System
Toyota	Pre-Crash System
Volvo	CitySafety
	CWAB-PD

2.2.2. Audi Technologies

Pre-Sense Basic detects potential collision from longitudinal, lateral acceleration and deceleration information while driving. If it detects emergency potential at speed greater than 30 kph, seatbelt will be tensioned and the action is reversible. In the case of high lateral acceleration, system will close side window and sunroof to avoid outside objects entering compartment in case of side collision [19].

Pre-Sense Front Plus is more advanced than Pre-Sense Basic and is designed to avoid or mitigate rear-end accidents and can be applied to either moving or stationary objects. The system uses two long range radars and one windscreen-mounted camera for detection of potential collision with preceding objects. Front view of host vehicle is captured by camera at rate of 25 fps [20]. Actions taken when potential collision appears are: provide optical and acoustic warning signals, pre-charge brake, further warning and apply partial brake if no action from driver, and finally apply full brake if collision could not be avoid. The system works at maximum speed of 200 kph, additionally warning signals are capable of working at speed even higher than 200 kph [21].

Adaptive Cruise Control (ACC) system is the most important and complex driver assistance system, recently has been introduced in Audi A6. It uses two long range high-end radars (76GHz - 77GHz, 40^o Field Of View (FOV), max range of 250 m) and one interior-mirror-based camera (40^o FOV, max range of 60 m). Hence control system is able to regulate host vehicle speed up to 250 kph. Furthermore, the control system interconnects with another 26 systems installed on the vehicle via FlexRay bus system to analyze data and provide assistance to driver. Several scenarios are developed in the system including changing lanes, passing or turning off at curvy roads, driving in city area. Interestingly, the system is able to slow down vehicle until it stops and start off again automatically through ultrasound sensors for parking assistance system installed at front bumper [22].

Pre-Sense Front together with ACC will serve purpose of preventing rear-end collision which might happen to host vehicle. Partial braking has effect to decelerate vehicle at rate of 3 m/s^2 and will further increase to deceleration rate of 5 m/s^2 before applying full brake force. In the combined system, time to collision, an important factor in designing AEB system, is 0.5 s. The system could reduce speed of vehicle up to 40 kph when impact occurs [22].

Side Assist and Pre-Sense Rear works together in detecting dangerous situation which might occur from the back of vehicle at speed greater than 30 kph. Similar to Pre-Sense Front, Pre-Sense Rear uses two radars with detectable range of up to 70 m to detect whether back-end approaching vehicle moves to critical range ,which might result in collision, or not. If it is in the critical range, led located in housing of exterior mirror will light up suggesting lane change for driver. In case driver ignores the signal, it will bright up and blink quickly to catch attention of driver [22].

Active Lane Assist is also added to Audi A6 to detect road markings using a camera installed in base of interior mirror. Up to eight markings and their colors can be detected by the system. If vehicle moves near a marking without turning signal, system will intervene electromechanical steering to move vehicle back to the lane. Adjacent objects nearby host vehicle will also be considered in the above strategy [22].

Night Vision Assistant is a system that highlights detected pedestrians and animals using thermal imaging camera as far infrared system (FIR) placed in the single-frame grill. Captured image are converted to black and white and displayed on a 7-inch screen. Working range of the camera is up to 300 m with FOV of 24^o. The camera will be heated during cold weather condition and it will be cleaned if dirty. Effective working range of the system is 100 m [22].

2.2.3. BMW Technologies

Driving Assistant & Camera-based Lane Departure Warning and Collision Warning Systems can detect lane markings and generate warning signal to driver if an unintentional lane change is detected when vehicle is at speed of 70 kph and above. The warning will be deactivated if turn indicator is on when vehicle is changing lane. The system can detect pedestrians and if there is a risk of collision, warning signal will be generated and brakes will be applied if needed. At speed from 15 kph, system will also warn driver if potential collision do exist and automatically pre-charge brake for faster response to emergency situation [23].

Active Protection system uses either front camera or front radar to detect potential collision in front of host vehicle. The system is enabled when vehicle has a speed of 18 km/h and above together with tightening seatbelt. It can detect dangerous situation created either by inattentive driver through symptoms of fatigue if vehicle is over or under steering. In this case front seatbelts will be tensioned, side windows and sunroof will be closed. At first system will recommend a brake in the Control Display installed on host vehicle, if crash could not be avoid, it will apply brake automatically until the vehicle be back to "standstill" state [23]. This strategy might help to avoid collision to occur or reduce impact if it happens.

2.2.4. Ford Technology

Active City Stop with price from £250 is a low-speed collision avoidance technology for vehicle moving in urban areas developed by Ford Motor Company. Up to 15 images will be captured and processed by its system to identify potential hazards on the road. Light detecting and ranging sensor is used in the system with setting scanning rate of 50 times per second. The system is claimed to prevent collisions at speed of up to about 16 kph and reduce severity level at speed up to about 32 kph. The system detects and calculates potential risk of hitting a stationary, slow-moving object. In case of emergency, it will firstly pre-charge brake and wait for response from driver, if no action is taken, system will apply brake and reduce engine torque, followed by light up rear hazard lights [24].

2.2.5. Volvo Technologies

CWAB-PD system, an active safety system, was developed by Volvo Car Corporation with features of applying full brake in emergency scenario, providing warning and brake support to avoid pedestrian accidents [5]. Full emergency braking action is reported to create effect of up to $10 m/s^2$ deceleration and could avoid collision at speed of 35 kph in this Volvo's third generation AEB design. The design of the system is based on the "Circle of Life" concept [5]. This means that statistical data of real-world accidents had been collected and analyzed to identify most common types of collisions, how often each collision happens and its causes before proceeding to selection of components and their specifications and come out with specific design of the system.

2.2.6. Time to Collision - The Most Common Risk Matrix in AEB Systems

Several common risk matrices are:

- Time to collision or time to impact;
- Predicted minimum distance;
- Predicted time to minimum distance;
- Time to react: Time to brake; Time to accelerate by kick down; Time to steer.

Among the above risk matrices, time to collision or time to impact is used mostly by the automakers:

Table 2: Time to Collision of AEB Systems [2, 5, 25]

Automaker	Time to Collision (s)
Audi • Pre-Sense Plus • Adaptive Cruise Control	0.5
Mercedes-Benz • Pre-Safe Brake • Brake Assist Plus • Distronic Plus - 2005	0.6
Volvo • CWAB-PD	1

The calculation of the above parameters is very dependent on collision scenario and "unobservable factors" [4]:

- driver's state and intention;
- surrounding objects;
- road condition.

Inefficient AEB system could be resulted from insufficient information of the above factors.

2.2.7. Stopping Distance - An Important Measure to Avoid Collision

One important feature that an AEB system needs to have according to Japanese Ministry of Land, Infrastructure and Transport (MLIT) [2] in Japan is that the "stopping distance" to avoid collision must be at least one meter [5]. This feature was tested for several pedestrian cases with car's speed ranged from 5 - 35 kph. Results from the test indicates that for all test cases, the vehicle could not achieve one-meter stopping distance [5].

2.2.8. Speed Reduction - An Important Measure to Mitigate Collision Effect

It is mentioned that a 10% reduction of speed before collision can reduce 30% fatal injuries and in pedestrian collision, a reduction of 25 kph from 50 kph can decrease 85% risk for fatal injuries [5]. Hence speed reduction can be considered as an important factor to mitigate collision effect.

2.2.9. Combination of Steering and Braking Maneuvers

Combination of both steering and braking maneuvers might result in more efficient collision avoidance than just steering only and therefore less false alarms are triggered [4]. However disadvantage of this combination is at the delay of system's response due to the extension of collision avoidance cases in consideration and analysis. Since the AEB system is design to handle situation in few seconds, the reaction time is an important factor to consider in developing the system. A small delay (e.g. 5 milliseconds [2]) also can result in insufficient braking and protective actions and might lead to more severe injury to driver and passengers.

2.2.10. Pre-Crash Detection & Post-Crash Detection Systems

Pre-Crash Detection System can be considered as part of AEB system, it detects potential crash between host vehicle and preceding object. Prior to the collision, pre-crash detection system can generate warning signals to driver, pre-charge brake, fasten seat belts, close windows, and apply partial brake then full brake force. These options vary among car manufacturers and car models.

Post-Crash Detection System which usually integrated in more advanced AEB system to reduce aftereffect of collision i.e. avoiding fire by using "fire-resistant" material for interior design of vehicle, isolate fuel tank or avoid fuel leakage after collision; using "black-box", a device normally is installed on aircraft, to record vehicle's conditions prior to and after the collision for after-collision investigation [2].

In general, CW system can be considered as pre-crash detection system. The system is used to feed information to AEB system to avoid collision or reduce collision's impact if collision occurs. The system should be able to detect type of crash, determine severity level and offer some safety features for passengers to avoid more serious or fatal injuries after collision such as avoiding fire to start.

2.2.11. Current Issues of AEB Systems - Causes and Potential Solutions

False Detection

None AEB system is perfect; hence there will be a certain error rate in detection of obstacles and false emergency alarm generated by the system. It is said that unnecessary emergency braking maneuver would create "unacceptable false activation" to driver [4]. Therefore, driver might feel less convenient while driving car and could deactivate the safety system to avoid the cause of inconvenience. In some cases, this action could introduce risk and hazard to driver and passengers and AEB system has no effect on improving safe driving conditions to them.

To avoid false alarm, [4] mentions that emergency brake could only be applied when driver already acknowledges the situation and starts to press brake pedal.

Ineffective AEB System

Factors that could result in ineffective AEB system:

- Infrastructure
- Weather conditions
- Measurement noises
- Vehicle uncertainties in condition estimation

It is suggested that future research should pay attention to infrastructure conditions to help reduce number of potential maneuvers to avoid collision. Information of road borders should be taken into consideration to help driver to brake earlier. During the one-hour "real-world" test in [4], only one false emergency brake was applied due to the formation of a "cloud of dust" by preceding vehicle. This means that the surrounding environment is important in the assessment of collision with preceding vehicle and more research attention should be placed to this point.

2.2.12. Trend and Future Application of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Technologies in AEB System

V2V and V2I communications allowing wireless transmission of information between two vehicles on the road or between one vehicle and roadside infrastructure could help to improve road safety [2] by preventing 76% of crashes on roadway [26]. Sending wireless information among vehicles helps the vehicles to obtain travelling information of each other in 360^{o} rather than just limited to line-of-sight as when using expensive radars or vision systems [2, 26]. Travelling information such as location, position, velocity and acceleration from one vehicle can be sent to another vehicle on road [26, 27]. The information is processed by both cars and potential dangers could be detected. Then warning signals can be generated to catch attention of drivers to avoid accident. When accident occurs, vehicles involved in the accident will get stuck on road and could be obstacles for other moving vehicles. It is dangerous if other drivers do not acknowledge that accidental vehicles are in front of their trajectories. Hence it is expected that with V2V and V2I technologies, modern vehicle could send accident information to approaching vehicles to alert their drivers about the potential danger. Additionally, modern vehicle could also send the information to infrastructures near roadside to notify about the accident, and provide necessary information for rescue service. Current V2V communication relies on radio communication at different frequency bands and faces signal distortion by "radio echoes" effect from objects existing on road side [28]. It is estimated that dedicated short range communication (DSRC) using IEEE 802.11p automotive W-Fi standard will be featured in 61.8 % new vehicles by 2027 [29]. However, the reliability of the communication standard in dense traffic scenarios such as in urban environment or traffic congestion is still questioned by researchers [27]. The communication issues are open areas for further research and development of V2V communication. Overall, future AEB system should offer V2V and V2I communications as basic needs for better accident avoidance and faster rescue service.

2.2.13. Summary

When detecting a potential collision, normally AEB system will first generate warning signals to driver for being more careful and cautious when driving. This function is provided by CW system. Then if collision potential increases, AEB system will pre-charge brake to assist driver in braking maneuver to avoid accidents. Once accident cannot be avoided or risk reaches a critical level, the system will apply full brake force to mitigate severity impact if accidents occur. Table 3 provides summary of AEB systems offered by several automakers:

Table 3: Summary of AEB Systems by Automakers

Automakers	Advantage	Disadvantage
Audi	- High Speed of Operation: 250 kph - Cover both front-end, back-end and side collisions - Back-end collision monitoring range up to 70 m - Speed Reduction: 40 kph - 3 rates of deceleration: 3 m/s^2, $5m/s^2$ and max brake force - Night Operation: Max 300 m, Effective 100 m - Can detect pedestrian	- Expensive e.g. 4 radar sensors for Audi A6 - Not consider weather condition
BMW	- Have change lane assistance at speed >70 kph - Warning function at speed >15 kph - Capable of monitoring driver's through symptoms of fatigue - Can detect pedestrian	- Still in early stage of development - Not consider weather condition

Ford	- Low-cost collision warning system at low speed (price from £250)	- Useful only for low speed vehicle - Not consider weather condition
Volvo	- Maximum deceleration rate: 10 m/s^2 - Can detect pedestrian	- Not function at night - Not consider weather condition

Time to collision is the most common risk matrix for assessing potential collision. Stopping distance and speed reduction are important factors in avoiding collision and mitigating collision aftereffect correspondingly.

It is important to develop various object classification classes for animals, pedestrians, bikes, different types of vehicles to have more effective AEB system. Additionally, future AEB system should offer V2V and V2I communications as standard for vehicles to have better accident avoidance and faster rescue service if accident occurs.

Sensors used in the system to detect obstacles surrounding host vehicle could be a combination of radar, laser and camera or a standalone of each sensor depending on automobile manufacturers' choices for their product ranges and prices. In this final year project, monocular vision based CW system using one camera is developed.

2.3.Sensors Used in AEB Systems

Overall, laser range finder, camera, accelerometer and speedometer are commonly used to obtain vehicle and surrounding's information for AEB system to function. The first two sensors could be used in combination or separation to identify obstacle on road while the third and fourth ones are used to measure current acceleration and velocity of vehicle.

2.3.1. Laser Range Finder

Laser range finder has three ranges: short range (i.e. less than 1 m), mid-range (i.e. less than 100 m) and long range (i.e. over 100 m) [6, 30-32]. Short range distance sensor is normally used for parking maneuvers while mid-range and long range distance sensors could be used for AEB system [6]. However only high-end vehicles will be offered with laser range finders, i.e. four laser range finders in Audi A6 [22].

2.3.2. Camera

Camera seems to be standard sensor to be used in AEB systems because of its affordable price. For lower end vehicle, camera becomes the only option to implement AEB system on the vehicle [2]. Automakers in Malaysia such as Proton and Perodua also should consider camera as the sensor to be used in their future cars.

2.3.3. Accelerometer

Acceleration of an object indicates a rate of change of its speed over a unit of time; acceleration's SI unit is m/s^2. In an AEB system, accelerometer measures acceleration of host vehicle at which it is attached to. Typical deceleration rate of AEB systems such as the first and second generations of Volvo CWAB-PD system is 5 m/s^2 or about 0.5 g [4, 5]. Third generation of Volvo CWAB-PD system even could provide double the typical deceleration rate at 10 m/s^2 or about 1 g [5].

2.3.4. Speedometer

For AEB system, it is required to have a speedometer to obtain speed of vehicle on road to perform calculation and estimation of vehicle's trajectory and identify potential dangers. Speedometer can be categorized into two types: analog and digital. Analog speedometer is used more widely than digital one. However, AEB system needs digital speedometer in order to obtain vehicle speed in digital data form then send to control unit for processing. Affordable digital automotive speedometers with two display options in kph and mph with unit price from about ~RM 480 to ~RM 870 [33].

2.4. Object Detection Methods and Techniques

Objects can be recognized based on decision-theoretic methods such as matching, optimum statistical classifiers and neural networks or based on structural methods [34]. Decision-theoretic method deals with patterns qualitatively and mostly does not consider any structural relationship of object. On the other hand, structural method deals with structural relationship of object to recognize boundary shape of unknown object. The former can be used for various types of application while the latter is mostly used in character or string recognition.

2.4.1. Object Detection by Matching - Minimum Distance Classifier

The technique represents each class by a prototype pattern vector. If pattern is unknown, it will be assigned to closest class in terms of predefined matric. Minimum distance classifier is the simplest approach in which Euclidean distance between unknown pattern and each of prototype pattern vectors will be calculated and compared, least distance will be used to make decision. It is stated that this technique works well when distances between means are large compared to randomness of each class with respect to its mean i.e. in Figure 1. Several sets of sample images of preceding vehicle back-ends at different distances are needed to be stored and used as predefined matrices. Because distance from host vehicle to preceding vehicle ranges from few meters to few tens of meters, this approach will require relatively large sets of preceding vehicle back-ends' sample images (representing different vehicle brands at different relative distances), hence classification will be time consuming task and the approach will not be suitable for this collision warning application.

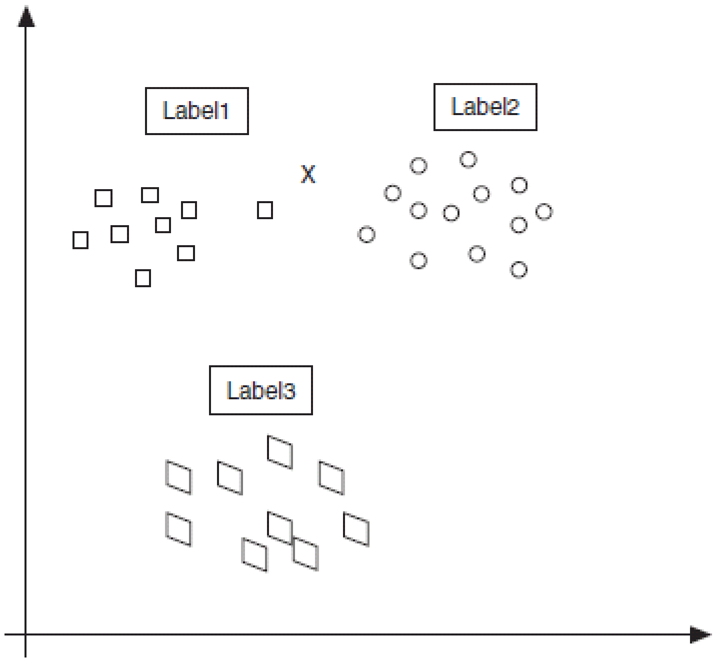


Figure 1: Minimum Distance Classifier [35]

2.4.2. Object Detection by Matching - Spatial Correlation

Figure 2 shows template matching mechanism by spatial correlation: a mask or template *w(x, y)* with size of *m x n* is required to visit every pixel of image *f* (larger than *w*) to assess its "normalized correlation coefficient" (NCE) [34]:

$$\gamma(x,y) = \frac{\sum_s \sum_t [w(s,t) - \bar{w}] x [f(x+s,y+t) - \bar{f}(x+s,y+t)]}{\sum_s \sum_t [w(s,t) - \bar{w}]^2 \sum_s \sum_t [f(x+s,y+t) - \bar{f}(x+s,y+t)]^{2^{1/2}}} \tag{1}$$

Where:

s, t: limits of summation are taken over the region shared by *w* and *f*

$\bar{w}$: average value of the mask (template), compute once

$\bar{f}(x+s,y+t)$: average value of *f* in region coincident with *w*

$\gamma(x, y)$: normalized correlation coefficient, in range [-1, 1]

The use of NCE will help to eliminate sensitivity of correlation result towards scale change in *f* and *w*. The highest value of NCE represents the most similarity between template *w* and corresponding coincident region on *f*. The respective position of *w* which results in highest value of NCE indicates center of the template. Since calculation of NCE is repeated for all pixels on *f*, advanced processors is required to handle image processing task. Applying to preceding vehicle detection, sample image of preceding vehicle will be referred as mask or template *w* while captured front view of host vehicle will be referred as image *f*.

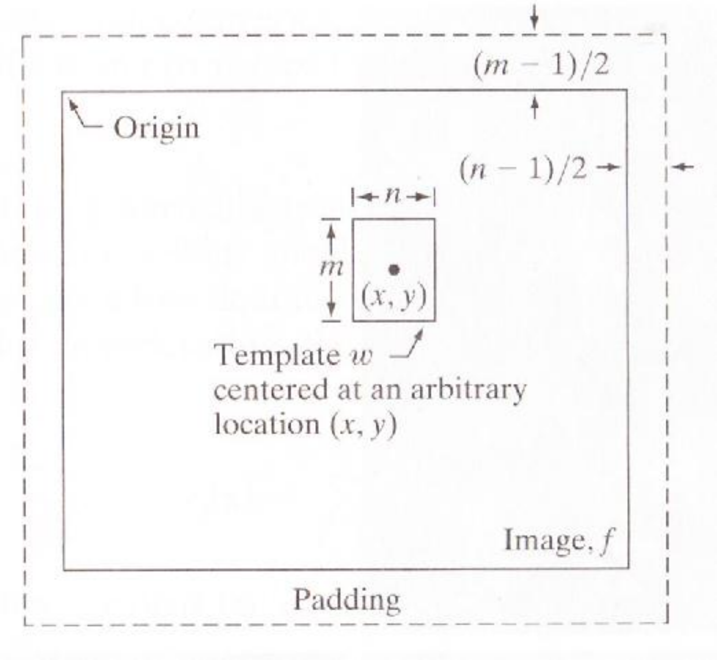


Figure 2: Template Matching Mechanism [34]

2.4.3. Object Detection by Optimum Statistical Classifiers

The technique deals with statistical information obtained from captured image for object classification purpose. The object will be categorized in to the class that yields the lowest probability of classification errors compared to that from other classes. Theoretically, the technique is difficult to apply in practice and only Bayes classifier for Gaussian pattern classes approach is commonly used. Typical application of this approach is in classification of "remotely sensed imagery generated by multispectral scanners" on aircraft, satellites and spacecraft [34].

2.4.4. Object Detection by Neural Networks

Two phases are involved in object detection using neural networks: training phase and classifying phase. In training phase, a set of patterns with known classes are used to feed in to the network and undergoing training. All parameters used to form structure of the network will be determined after training phase is done and these parameters remain unchanged throughout the classifying phase. In classifying phase, an unknown object will be tested by the trained network and the most appropriate class will be assigned to the object [34]. Overall this approach is much more advanced than previous approach discussed earlier and will not be used in this project.

2.4.5. Object Detection by Background Subtraction and Thresholding

Two basic pixel-by-pixel image operations is addition and subtraction [36]. Additional object can be placed into an image with addition operation while duplicated object can be removed from the images with subtraction operation. One application of subtraction operation is to detect change in images' sequence [36, 37]. Figure 3 shows human detection by background subtraction and thresholding technique. The technique is relatively good when background of image is well known or else more than one featured object will appear on the image i.e. extra background of subject's head. To estimate background of sequence of images, median filter could be used [37]. The attractiveness of subtraction and thresholding technique is at the simplicity of processing procedure hence processing speed is improved. It is noted that the technique is sensitive to noise, variation and illumination of preceding object.

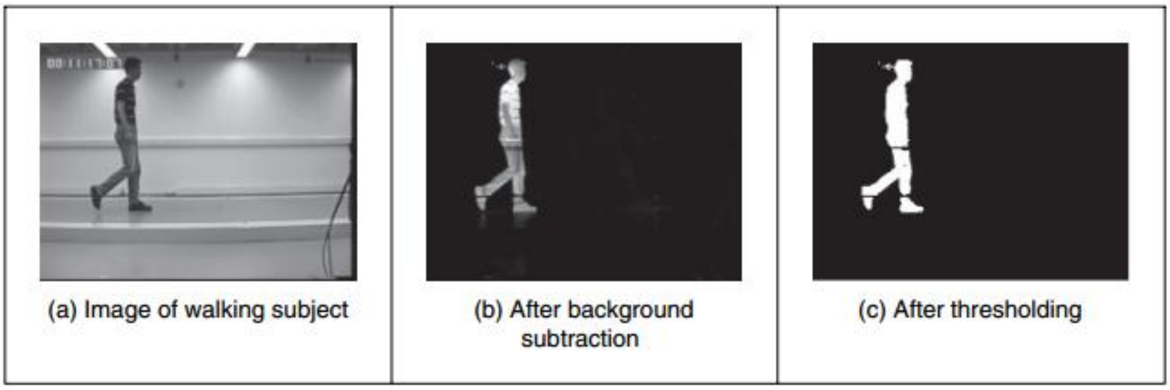


Figure 3: Object Detection by Background Subtraction and Thresholding [37]

2.4.6. Object Detection by Structure

Unlike object detection by decision-theoretical method, this method considers structure of unknown object for classification task. It is mainly suitable for object boundary detection and mostly used in text recognition. In development of CW system, inverted U-shape of preceding vehicle's back wheel structure is used in vehicle detection.

2.4.7. Summary of Object Detection Methods & Techniques

Table 4 presents summary of object detection techniques. Decision-theoretic method for object detection has wide area of application; however it does not take into consideration of object structure. On the other hand structural method for object detection has limited application to character recognition and does consider object structure. In this project, it is suggested that spatial correlation and background subtraction and thresholding techniques are used to detect preceding vehicle due to the algorithms' simplicity to ensure that the project is deliverable within time constraint. Additionally, vehicle's back wheel structure also should be considered to develop vehicle detection algorithm for CW system.

Table 4: Summary of Object Detection Method & Techniques

Method/Technique	Advantage	Disadvantage
Decision-Theoretic Method [34]	- Wide area of application	- Do not consider structure of object
• Minimum Distance Classifier [34]	- Simplest technique	- Require large set of samples - Time consuming
• Spatial Correlation [34]	- Simple algorithm - Small sample size - Less set of samples compared with minimum distance classifiers	- Time consuming - Accuracy subject to well defined object and object size
• Optimum Statistical Classifiers [34, 38]	- Good for classification of sensed imagery created remotely by multispectral scanners	- Limited application - Theoretically impractical
• Neural Networks [34, 39]	- Advanced algorithm - Dynamic - Can be used for various classification problems	- Time consuming - Complicated algorithm
• Background Subtraction and Thresholding [36, 37, 40]	- Simple algorithm	- Performance affected by noise, variation and illumination

Structural Method [34, 41]	- Consider structure of object, suitable for object boundary detection	- Narrower area of application, e.g. string recognition

CHAPTER 3: RESEARCH METHODOLOGY AND PROJECT ACTIVITIES

At first project title is proposed based on original idea formation and some search results of issues related to proposed project title. Later, an intensive literature review is carried out for deeply understanding about AEB and CW systems. An existing problem of the systems is clearly addressed and a solution for the existing problem is proposed. In this project, CW system is developed in MATLAB environment. A real-world drive video is captured and used to evaluate CW system. Finally performance of the system is assessed and report of the project is submitted as per requirement of the course.

3.1.Sensor Selection

Laser range finder, camera, accelerometer and speedometer are commonly used to obtain vehicle and surrounding's information for AEB system to function. The first two sensors could be used in combination or separation to identify obstacle on road while the third and fourth ones are used to measure current acceleration and velocity of vehicle. Based on the analysis of several sensor options in section 3.1.1 to section 3.1.5, camera integrated on Acer Liquid C1 phone will be used as the only one sensor for detection of preceding vehicle.

3.1.1. Laser Range Finder

Laser range finder has three ranges: short range (i.e. less than 1 m), mid-range (i.e. less than 100 m) and long range (i.e. over 100 m) [6, 30-32]. Short range distance sensor is normally used for parking maneuvers while mid-range and long range distance sensors could be used for AEB system [6]. Laser range finder is considered to be excessively expensive for mid-range and long range object detection application (i.e. Model UTM-30LX - Cytron Technologies Sdn. Bhd. at RM 20,622 for distance up to 30 m - Appendix A - Summary of Laser Range Finders Prices and Specifications [12, 13]). Therefore it is not suitable for AEB system for cars manufactured by dominant Malaysian automobile makers such as Proton and Perodua (i.e. Proton Exora at RM 59,548 or Perodua Viva 660 BX at RM 24,924 respectively - Appendix B - Starting Prices of Proton and Perodua's Cars [8-11]) and will not be put in to consideration in the development of this project.

3.1.2. Camera

Since laser range finder is costly and not suitable to be used in Malaysian cars, an alternative sensor would be camera. With the advancement of technologies, currently high quality cameras have affordable prices i.e. ABUS Security-Center TVIP51500 at RM 2,164 and resolution of 1280x1024 pixels - Appendix C - Summary of Outdoor Cameras' Prices and Specifications [25]. There are two options for using camera as sensor to detect object in front of a car: day camera with daytime functioning capability and day/night camera with both daytime and nighttime functioning capability [42, 43]. Several differences between CCTV camera and normal camera (i.e. DSLR vs compact cameras) do exist. Normal camera can be used to take picture and then edit or process it after the picture is taken (on and off operations). CCTV camera is mainly used for safety monitoring purpose and offers real-time monitoring or operation per preset schedule (continuous operation) [44]. Therefore an affordable CCTV camera would be used as an alternative to detect obstacles on road.

3.1.3. Accelerometer

At the time this report is written, there are about 90 choices for accelerometers offered by RS Malaysia Sdn. Bhd. with various unit prices from RM 6.55 for 3-axis digital accelerometer in model MAG3110FCR1 made by Freescale to RM 7,820.00 for 6-degree-of-freedom low-noise inertial sensor in model ADIS16375AMLZ manufactured by Analog Devices [25].

Selection of accelerometer for specific application such as in AEB system is suggested to be based on prioritized factors [45]:

- range of measurement in term of g (1 g ≈ 9.81 m/s^2);
- measurement sensitivity (mV/g);
- response frequency (Hz);
- sensitive axis (single axis, tri-axis);
- size and weight of sensor.

Typical deceleration rate of AEB systems such as the first and second generations of Volvo CWAB-PD system is 5 m/s^2 or about 0.5 g [4, 5]. Third generation of Volvo CWAB-PD system even could provide double the typical deceleration rate at 10 m/s^2 or about 1 g [5]. With this information, accelerometer with measurement limit of 1.5

g (about 50% over maximum deceleration rate in [5]) would be suitable for use in AEB system.

3.1.4. Speedometer

Digital automotive speedometers with two display options (kph & mph) having unit price ranges from about ~RM 480 to ~RM 870 [33]. This range of price is considerably suitable to install on Malaysian cars. For experiment purpose, an alternative to measure speed of vehicle is to use mobile phone, which nowadays normally comes with GPS sensor, and a specific software such as Digital Speedometer developed by Balabanimation LLC [46]. This solution serves the function of a speedometer, is free of charge and could provide recorded information of vehicle's speed during an experimental drive.

3.1.5. Powerful Android-based Mobile Phone as All-in-One Solution

Recently introduced mobile phones i.e. Acer Liquid C1 [47] normally have been integrated with various types of sensors i.e. main camera and front camera for taking picture and recording video, GPS sensor for location and speed measurements, speaker for alarming purpose. Figure 4 shows how a smart phone can be used as all-in-one solution for collision warning purpose. Nowadays, mobile phones are normally featured with powerful processor i.e. Intel Atom Z2420 duo core 1.2 GHz. Hence image processing, a typically time-consuming task, is expected not to overload processor processing's capability. On the other hand, modern and affordable mobile phone could become a standalone CW system with potential for real-time image processing and will not provide late warning signal to driver when emergency situation occurs. Since there is no application written for Android-based devices for this collision warning purpose, to ensure delivery of the project the mobile phone will only be used to capture video of a real-world drive and the video will be used to evaluate CW system in MATLAB environment.

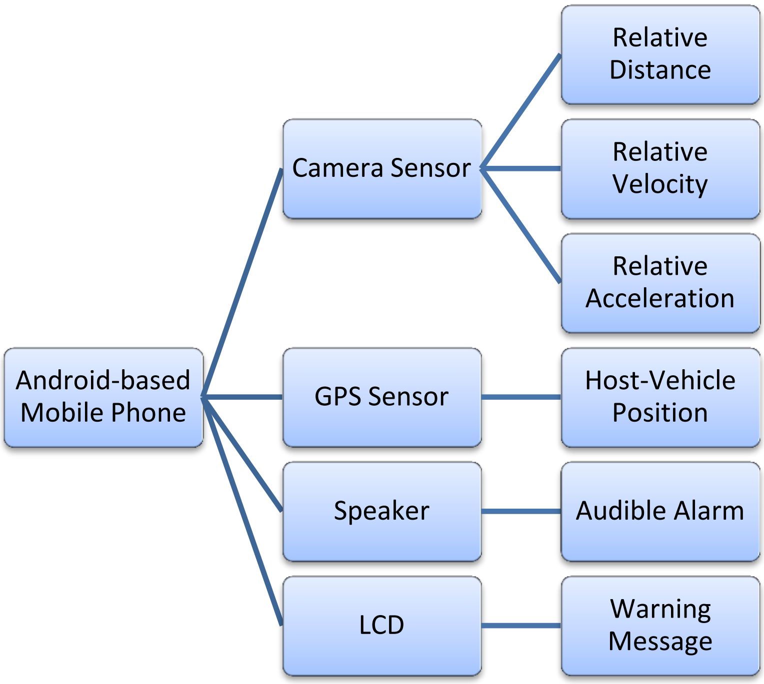


Figure 4: Android Mobile Phone as All-in-One Solution

3.2.Bottom-up Analysis and Method to Detect Potential Danger and Provide Warning Signal to Driver

In this section, a bottom-up analysis method will be used to perform further analysis of the project in order to breakdown the works into smaller sections ensuring feasibility for implementation of the project.

3.2.1. Time to Collision & Predicted Time to Minimum Distance

To provide warning signal e.g. audible warning, flashing message or lights, CW system needs to be able to identify potential danger to host vehicle based on front image capturing during the driving. Common risk matrices are time to collision or time to impact, “predicted minimum distance” or “predicted time to minimum distance”, “time to react” (including “time to break”, “time to accelerate by kick down”, “time to steer”) [4]. Time to collision indicates predicted time for accident to occur while predicted time to minimum distance indicates predicted time to avoid collision between two vehicles; these will be used as safety measures in the project.

To calculate time to collision and predicted time to minimum distance, it is required to know relative distance, velocity and acceleration of host vehicle towards preceding vehicle. The relationship is shown in the following formula:

$$d = v_0 t + a\frac{t^2}{2}$$

(2)

Where:

d: relative distance between host vehicle and preceding vehicle [m]

v_o: initial relative velocity between host vehicle and preceding vehicle [m/s]

a: relative acceleration between host vehicle and preceding vehicle [m/s^2]

t: time to collision or predicted time to minimum distance [s]

Root of equation:

$$t = \frac{-v_0 \pm \sqrt{v_0^2 + 2ad}}{a}$$

(3)

- When t>0, there is potential for collision between two vehicles. When t<0, there will not be potential for collision between two vehicles since they will move away from each other.
- When v_0>0, host vehicle is approaching preceding vehicle, collision might happen. When v_0<0, host vehicle is moving away from preceding vehicle, collision will not happen. In this study, only when v_0>0, collision warning is put into consideration.
- To avoid collision, relative distance is greater than zero: d>0.
- When a>0, v_0 increases, collision might occur. When a<0, v_0 decreases, collision will not occur. In this study only when a>0, collision warning is put into consideration.

With the above observations, expected time to collision (t>0) is calculated based on the positive case in equation (3).

3.2.2. Rear-end and Front-end Collisions

For rear-end collision, host vehicle will approach preceding vehicle from rear end, there will be few cases as shown below:

✓ Case 1: Both host vehicle and preceding vehicle increase speeds, further consideration is needed to identity existence of potential collision.

  - If speed increment of host vehicle is higher than speed increment of preceding vehicle, host vehicle will move nearer to preceding vehicle, therefore there will be potential for collision.

- If speed increment of host vehicle is smaller than speed increment of preceding vehicle, preceding vehicle will move farther than host vehicle, potential for collision will be decreased over time.

✓ Case 2: Host vehicle increases speed while preceding vehicle decreases speed, relative distance between two vehicles will decrease. There is potential for collision between the twos.

✓ Case 3: Host vehicle decreases speed while preceding vehicle increases speed, relative distance between two vehicles will increase. They will move away from each other hence there will be no potential for collision between the twos.

For front-end collision, preceding vehicle will approach host vehicle from front end, there will be always a potential for collision between the twos. Depending on direction and change in speed, potential for collision might increase or decrease.

✓ Case 1: Both vehicles increase their speeds, potential for collision will increase since they will approach each other with higher relative velocity equaled to sum of each individual speed.

✓ Case 2: One vehicle increases speed while the other one decreases speed. Further consideration is needed to identify change in potential collision.

- If speed increment is higher than speed decrement in term of value, overall relative speed will increase, two vehicles will approach faster, potential collision increases.
- If speed increment is smaller than speed decrement in term of value, overall relative speed will decrease, two vehicles will slower approach, potential collision decreases.

3.2.3. Relative Acceleration Measurement and Prediction

Similar to absolute acceleration of an object, a relative acceleration of an object has a relationship with its relative velocity through the following formula:

$$a = \frac{dv}{dt} \tag{4}$$

It also can be defined as change in relative velocity over an amount of time or:

$$a = \frac{\Delta v}{\Delta t} \tag{5}$$

If relative velocity is a function of time and is known, relative acceleration can be calculated.

3.2.4. Relative Velocity Measurement and Prediction

Similar to absolute acceleration of an object, a relative speed of an object has a relationship with its relative distance from another object through the following formula:

$$v = \frac{dd}{dt} \tag{6}$$

It also can be defined as change in relative distance over an amount of time or:

$$v = \frac{\Delta d}{\Delta t} \tag{7}$$

If relative distance is a function of time and is known, relative velocity can be calculated.

3.2.5. Relative Distance Measurement and Estimation

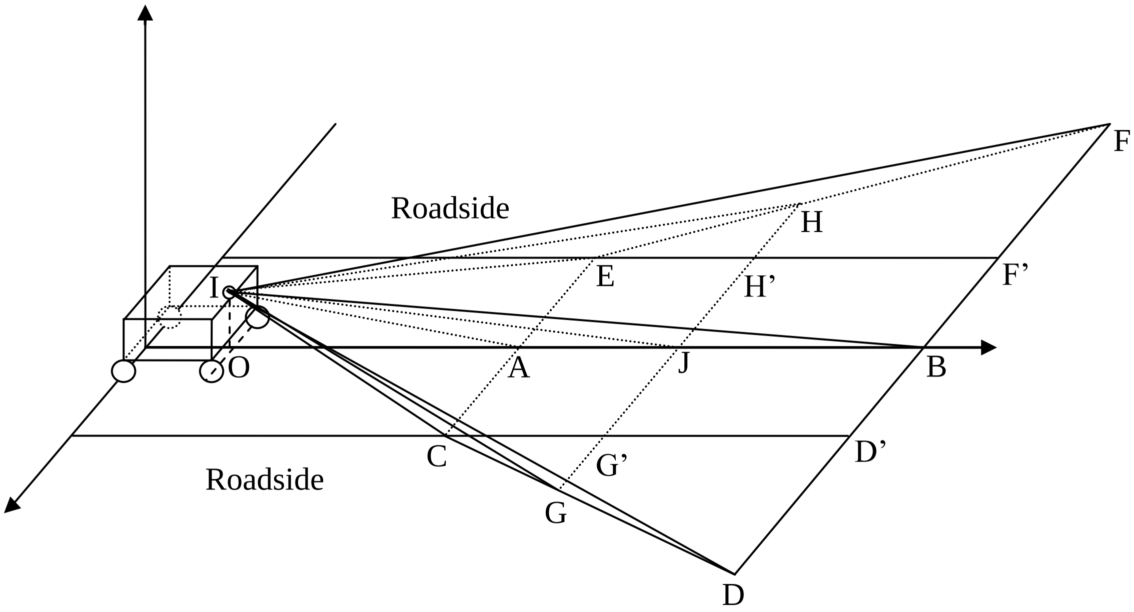


Figure 5: Camera Setup on Host Vehicle

To estimate host-to-preceding vehicle relative distance, the coordinate origin (O) is selected to be on host vehicle and become a movable origin. Since O is movable, the detected distance from host vehicle to preceding vehicle will always be relative distance between the two objects. Table 5 shows legend of Figure 5:

Table 5: Figure 5's Legend

I	Position of camera on host vehicle
OI = h	Height of camera over road surface, the value will be fixed once camera is installed on host vehicle
A	Nearest point from host vehicle that camera can capture

B	Furthest point from host vehicle that camera can capture
J	Projected center point of camera on road surface
OA	Blind area in front of host vehicle, any object within this range will not be seen by camera
AB	Capture range on road surface of camera
CDFE	Region on road surface to be captured by camera
CD'F'E	Region of interest for image processing and object detection
CE	Minimum width of road detected by camera
DF	Maximum width of road detected by camera
OIA = γ	Vertical setup angle of camera, this angle is adjustable upon setup of camera on host vehicle
AIB = $\Delta\gamma$	Vertical FOV (VFOV) of camera, for one camera, this angle will be physically fixed
IJ	Bisector of angle AIB
GIH = θ	Horizontal FOV (HFOV) of camera, for one camera, this angle will be physically fixed

Finding math expressions:

Right triangle OIA:

$$\tan\gamma = \frac{OA}{OI} = \frac{OA}{h} \quad (8)$$

$$OA = h\tan\gamma \quad (9)$$

Right triangle OIB:

$$\tan(\gamma + \Delta\gamma) = \frac{OB}{OI} = \frac{OB}{h} \quad (10)$$

$$OB = h\tan(\gamma + \Delta\gamma) \quad (11)$$

$$AB = OB - OA = h\,(\tan(\gamma + \Delta\gamma) - \tan\gamma) \quad (12)$$

Right triangle OIJ:

$$\tan(\gamma + \frac{\Delta\gamma}{2}) = \frac{OJ}{OI} = \frac{OJ}{h} \quad (13)$$

$$OJ = h\tan(\gamma + \frac{\Delta\gamma}{2}) \quad (14)$$

$$AJ = OJ - OA = h\,(\tan\left(\gamma + \frac{\Delta\gamma}{2}\right) - \tan\gamma) \quad (15)$$

$$IJ = \sqrt{IO^2 + OJ^2} = \sqrt{h^2 + h^2\tan(\gamma + \frac{\Delta\gamma}{2})^2} = h\sqrt{1 + \tan(\gamma + \frac{\Delta\gamma}{2})^2} \quad (16)$$

Right triangle JIG:

$$\tan\frac{\theta}{2} = \frac{GJ}{IJ} = \frac{GH}{2IJ} \quad (17)$$

$$GH = 2IJ \tan\frac{\theta}{2} = 2h\sqrt{1 + \tan\left(\gamma + \frac{\Delta\gamma}{2}\right)^2} \tan\frac{\theta}{2}$$

(18)

Right triangle OIA:

$$\cos\gamma = \frac{OI}{IA}$$

(19)

$$IA = \frac{OI}{\cos\gamma} = \frac{h}{\cos\gamma}$$

(20)

Right triangle AIC:

$$\tan\frac{\theta}{2} = \frac{CA}{IA} = \frac{CE}{2IA}$$

(21)

$$CE = 2IA \tan\frac{\theta}{2} = 2\frac{h}{\cos\gamma}\tan\frac{\theta}{2}$$

(22)

Right triangle OIB:

$$\cos\gamma + \Delta\gamma = \frac{OI}{IB}$$

(23)

$$IB = \frac{OI}{\cos\gamma + \Delta\gamma} = \frac{h}{\cos\gamma + \Delta\gamma}$$

(24)

Right Triangle BID:

$$\tan\frac{\theta}{2} = \frac{DB}{IB} = \frac{DF}{2IB}$$

(25)

$$DF = 2IB \tan\frac{\theta}{2} = 2\frac{h}{\cos\gamma + \Delta\gamma}\tan\frac{\theta}{2}$$

(26)

For each camera, its specification such as focal length, sensor diagonal length, frame size will be provided by manufacturer. Camera's FOV can be calculated using formula given in [48] or through experiment setup shown in [49].

It is observed that once camera is setup on host vehicle, its height (OI = h) compared with road surface will be fixed hence the following parameters can be calculated: OA, OB, OJ, AB, IJ, CE, GH, DF. The information will be useful for optimization of tracking algorithm by focusing on certain area of interest i.e. portion of image representing road side in triangles EF'F and CD'D will not be processed. This will help to save time taken for image processing and therefore improve overall performance of the system.

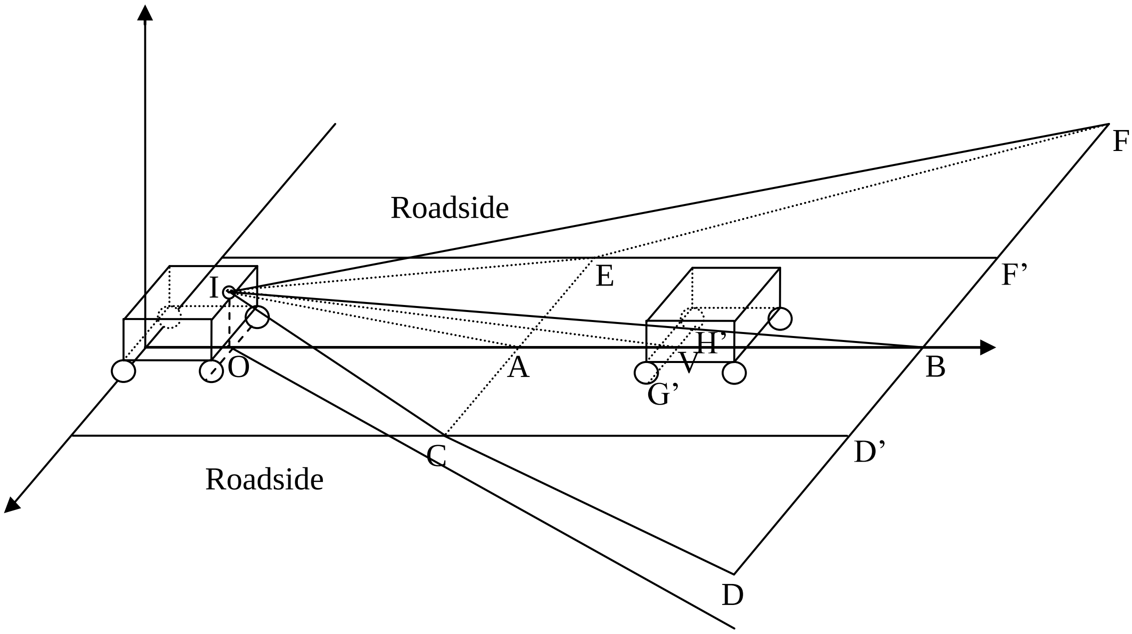


Figure 6: Host-to-Target Vehicle Relative Distance Measurement

Table 6 shows legend of Figure 6:

Table 6: Figure 6's Legend

V	Position of center of preceding vehicle's back-end on road surface
OV = d	Relative distance from host vehicle to preceding vehicle's back-end
G'H' = w_v	Width of preceding vehicle, the value will be fixed for a specific vehicle

If preceding vehicle moves within the range CD'F'E there is possibility for object tracking algorithm to detect the vehicle. Once the vehicle is detected, its center point on road surface (V) will be identified and relative distance to host vehicle (OV) will be calculated. Subsequently, host-to-preceding vehicle relative velocity and acceleration can be estimated. The better tracking algorithm is, the more accurate relative distance, velocity and acceleration is estimated.

3.2.6. Object Detection and Tracking Algorithms

Basically, there will be two steps involving in detecting preceding vehicle:

- Step 1: Apply filter(s) to image (if any)
- Step 2: Detect preceding vehicle

Various spatial filtering and filtering in frequency domain techniques are introduced in [34], throughout the development of the project, several filters will be considered to be used to filter image's noises. For comparison purpose, three object detection techniques will be used:

- Object Detection by Matching - Spatial Correlation technique presented in section 2.4.2;
- Object Detection by Background Subtraction and Thresholding and Object Detection by Structure presented in section 2.4.5 and section 2.4.6 respectively
- Object Detection by Structural presented in section 2.4.6.

The best technique is used to implement for vehicle detection in designed CW system.

3.3. Collision Warning Process

Figure 7 shows overall operation process of CW system: first preceding vehicle needs to be detected and its relative distance to host vehicle will be estimated, followed by relative velocity and acceleration. Then potential risk parameters, e.g. time to collision, will be calculated to identify potential collision between two vehicles. If risk parameters' values exceed preset thresholds, warning signal will be generated to alarm driver to be more cautious and careful when driving. For simplicity, one level alarm signal will be used to indicate potential danger.

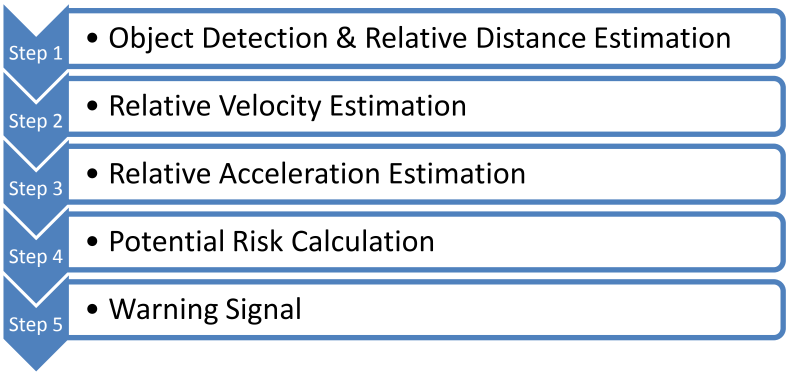


Figure 7: Overall Operation Process of CW System

3.4. Work Breakdowns

In order to make project feasible within limited time frame (two trimesters or equivalently 8 months), the project needs to be separated into smaller modules representing major works to implement the project:

- Literature Review of Existing AEB and CW Systems and Area for Improvement of Current Systems
- Image Processing and Object Detection Algorithm
- Host-to-Target Relative Distance Estimation Algorithm
- Host-to-Target Relative Velocity Estimation Algorithm

- Host-to-Target Relative Acceleration Estimation Algorithm
- Warning Signals

3.5. Required Tools

- **Software:**
  - ✓ MATLAB: to develop and simulate CW system.
- **Hardware:**
  - ✓ An integrated camera on a smartphone (i.e. Acer Liquid C1): to capture video while driving;
  - ✓ A universal car mount for smartphone: to mount smartphone on car's windshield to capture video while driving;
  - ✓ A rental car: to drive and capture video;
  - ✓ A valid Malaysian driving license: to legally drive in Malaysia;
  - ✓ Laptop (i.e. Dell Inspiron N4110, Core i5-2450M CPU 2.5GHz, Ram 4GB, Windows 7 64-bit): to run MATLAB.

3.6. Key Milestones

- 07/2013: Finish Literature Review
- 08/2013: Submit Final Year Project 1 Report
- 11/2013: Finish Development of CW System
- 11/2013: Finish Testing of CW System
- 12/2013: Submit Final Year Project 2 Report

3.7. Study Plan (Gantt Chart)

Task	**2013**						
	6	**7**	**8**	**9**	**10**	**11**	**12**
Literature Review and Extended Proposal Preparation	X						
Final Year Project 1 Report		X	X				
Purchase of Devices, Components, and Installation of Software for Experiment			X	X			
Development of CW System							
- Object Detection			X	X	X	X	
- Distance Estimation				X	X	X	
- Velocity Estimation					X	X	
- Acceleration Estimation					X	X	
- Warning Signals						X	
Collection of Real-world Experiment Data				X	X	X	X
Processing Experiment Data and Provide Results					X	X	X
Final Year Project 2 Report				X	X	X	X

CHAPTER 4: STUDY RESULTS AND DISCUSSIONS

4.1.General Observation through Captured Image of Host Vehicle's Front View

In Figure 8, preceding vehicle is represented by 3 red point groups illuminated from brake lights at the back of the vehicle. These groups indicate that preceding vehicle is decelerating or reducing its speed. The picture is slightly blurred due to rain drops appearing on windshield of host vehicle. Therefore, prior to the processing of image, suitable filter needs to be applied to remove "drop" noises and make picture clearer.

Figure 8: Front View of Host Vehicle in Rainy Day, Picture taken by Acer Liquid C1 at Universiti Teknologi Petronas

Zoom in preceding vehicle for more detail, Figure 9 shows that brake lights are above vehicle's wheels and will be above road surface. If detection of preceding vehicle is limited to only three red groups mentioned above, three issues might occur:

- The lights will be on only when driver apply brake pedal, if driver does not apply brake pedal, lights will not be on and therefore it is impossible to identify preceding object.
- Some vehicles have only two brake lights, some might have four brake lights, therefore number of brake lights will change from vehicle to vehicle and tracking algorithm needs to be dynamic to tackle the cases.
- About accuracy issue, detecting preceding vehicle based on brake lights will not be accurate since the lights are above ground surface.

Therefore brake lights are good indicator for detecting deceleration of preceding vehicle and will not be suitable for distance measurement from host vehicle.

Figure 9: Zoom-in Preceding Vehicle

For object detection by structure, in this final year project, an inverted U-shape structure of vehicle's back wheel is used to detect preceding vehicle. Figure 10 shows inverted U-shape back wheel structure of several vehicles. By utilizing this feature of four wheel vehicle, it is possible to detect touching line of vehicle's back wheels on road surface, hence, relative distance between host vehicle and preceding vehicle could be more accurate.

Figure 10: Inverted U-shape Back Wheel Structure of Several Vehicles

4.2.Theory Calculation

4.2.1. Detection Range

Nearest and furthest points, which could be captured by camera in the setup shown in Figure 5, can be calculated through (9) & (11).

Assumption:

H = 1.5 m

$\Delta\gamma$ = 30^o

Γ = [50 ^{o}, 51 ^{o}, 52 ^{o}, …, 60^o]

Result:

h	1.2	m	
DGamma	0.523	rad (30 degrees)	
Pi	3.14		
Gamma (degrees)	Gamma (rad)	OA (Nearest, m)	OB (Furthest, m)
50	0.87	1.43	6.78
51	0.89	1.48	7.54
52	0.91	1.53	8.49
53	0.92	1.59	9.71
54	0.94	1.65	11.34
55	0.96	1.71	13.60
56	0.98	1.78	16.98
57	0.99	1.85	22.56
58	1.01	1.92	33.61
59	1.03	1.99	65.78
60	1.05	2.08	1506.92

When upper edge of camera reaches horizontal line ($\gamma + \Delta\gamma = 90^o$), the system will be able to capture furthest point at distance of 1506.92 m, which is unrealistic. When $\gamma + \Delta\gamma = 89^o$, the furthest point that could be captured by camera, theoretically will be 65.78 m away from host vehicle. However, while vehicle is moving on the road, there are many factors that contribute to the vibration of the vehicle and vibration of camera e.g. uneven road surface, hence this angle will not be fixed resulting in distance estimation error. When the sum is equal to 88^o, maximum detection distance will be 33.61 m, about half of the previous case. Good measurement range would be from 8.49 m to 22.56 m. Since result from this estimation will be relative distance between host and preceding vehicle, realistic maximum relative distance could be 22.56 m.

4.2.2. Object Detection by Matching - Spatial Correlation

Number of NCEs

In this project, at first, object detection by matching, using spatial correlation technique (section 2.4.2) is used to detect preceding vehicle. Because the technique requires calculation of NCE for every pixel in a captured image, number of calculations can be calculated based on Figure 11:

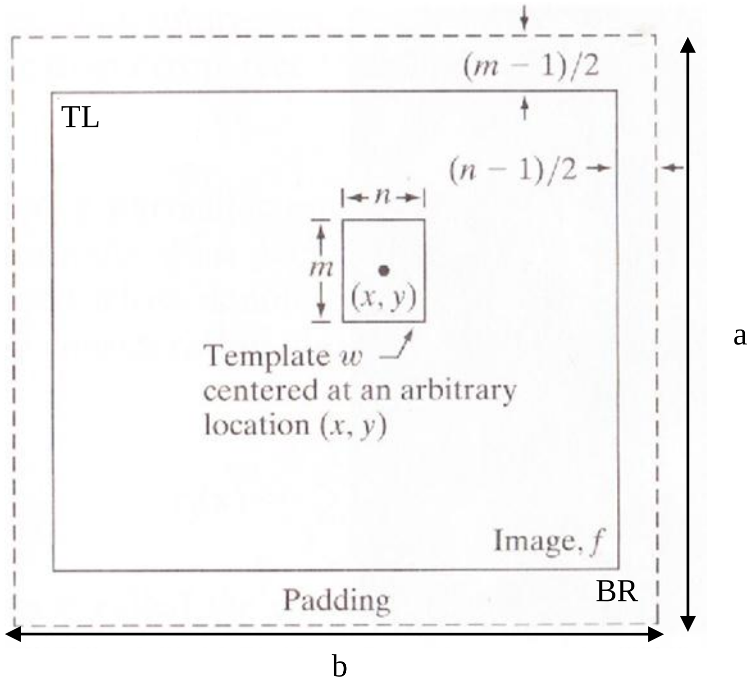


Figure 11: Calculation of Number of NCEs

Template size is *m x n* pixels. Center of template is a single point with coordinates *(x, y)*. Hence to make sure template is symmetrical, *n* and *m* should be odd numbers. Assume that image *f* has size of *m x n* pixels. Center of template will need to visit all pixels in the sub region (bounded by a continuous line) with top left (TL) coordinate of *((n-1)/2 +1, (m-1)/2+1)* and bottom right (BR) coordinate of *(b-(n-1)/2-1, a-(m-1)/2-1)*.

Number of NCEs is:

$$NCEs = \left(b - \frac{n-1}{2} - 1 - \left(\frac{n-1}{2} + 1\right)\right) x \left(a - \frac{m-1}{2} - 1 - \left(\frac{m-1}{2} + 1\right)\right)$$

$$NCEs = (b - n + 1)x(a - m + 1) \qquad (27)$$

In Table 9, video can be captured with maximum speed of 30 fps for 1080p image frame. Hence a = 1080 and b = 1920. If template width and height are about 10% of width and height of video frame: m = 107, n = 191. Number of NCE calculations for one frame is 1,685,020. To achieve real-time image processing, per second, number of NCE calculations would be pretty large: 50,550,600. This will exceed processing capability of dual-core 1.2 GHz processor (~ 23 processor cycles to calculate 1 NCE while processor has to carry other tasks as well).

There are two ways to reduce work load for processor when detecting preceding vehicle:

- To reduce image (video frame) size e.g. reduction of video frame size from 1080x1924 pixels to 600x800 pixels could reduce number of NCE calculations from 1,685,020 to 391,324 (76.78% reduction);
- To process only region of interest (ROI) on recorded frame i.e. in Figure 12, ROI is CD'F'E.

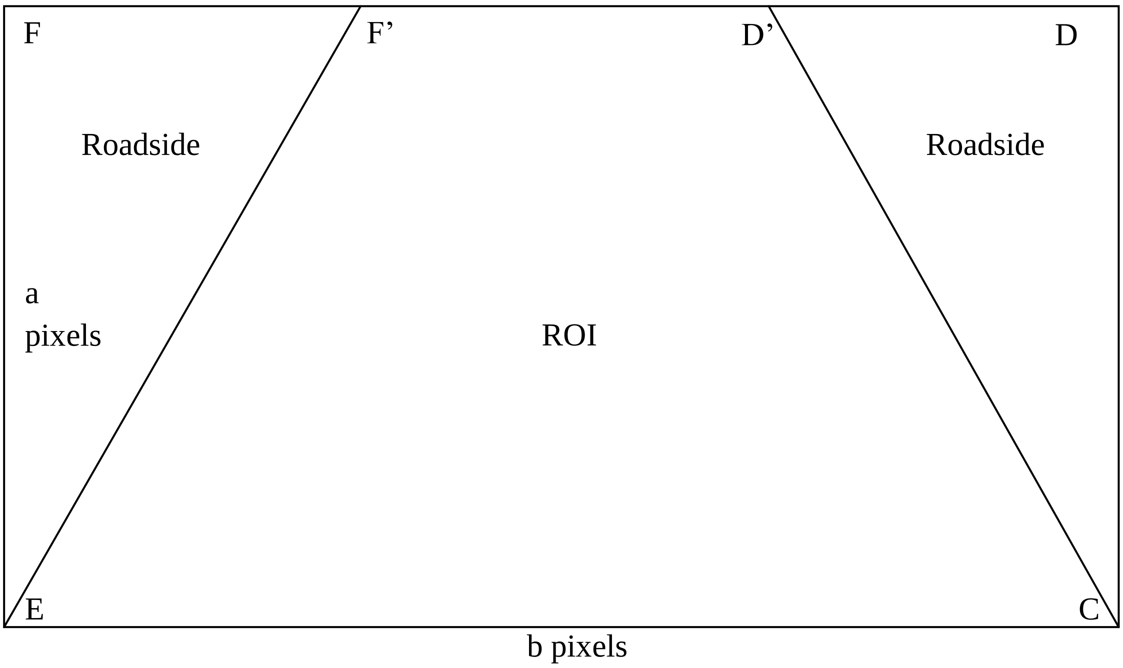


Figure 12: Region of Interest on Video Frame

In Figure 12, roadsides are represented by triangles EFF" and CDD'. When calculating NCEs, it is not important to calculate NCEs for all pixels located in these triangles because they represent roadsides. Number of pixels which should not be considered would be twice number of pixels located within triangle EF'F. It is suggested that ROI boundaries are set through experiments.

4.3. Experiments

4.3.1. Camera' FOVs and Risk Calculation

[49] shows experiment to measure camera's FOVs, results from experiment for 1080p images are shown in Figure 13.

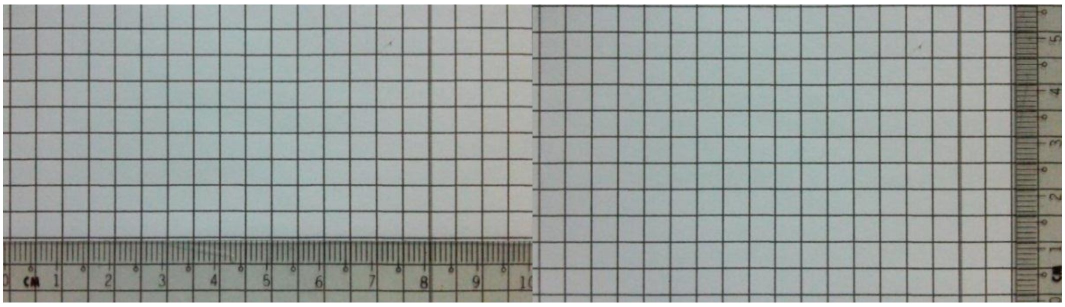


Figure 13: Measurement of Horizontal and Vertical FOVs

In this measurement, camera is placed 16.1 cm away from the paper, length and width of paper are 10.1 cm and 5.7 cm respectively. These result in HFOV = 34.83^o and VFOV = 20.08^o. In real-world drive, camera is placed 1.2 m above road surface. For ease of installation, camera setup in Figure 5 is modified so that the central point J represents furthest point that camera could detect. With this installation, point B is above horizontal plane crossing through I.

For ease of observation, camera installation along x-axis is shown in Figure 14:

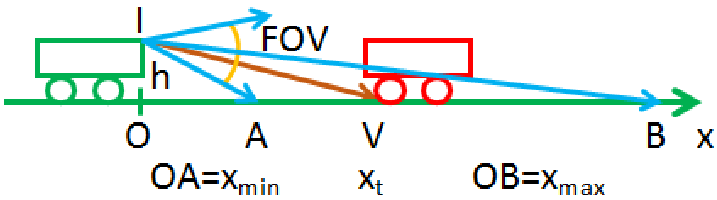


Figure 14: Modified Camera Installation (FOV = VFOV)

Formula (11) is modified:

$$OB = h\tan(\gamma + \frac{\Delta\gamma}{2}) \tag{28}$$

Where:

OIA = γ = 77^o

AIB = $\Delta\gamma/2$ = 10.04^o

Therefore minimum detectable distance is: OA = 5.2 m; maximum detectable distance is: OB = 22.9 m, similar to pre-modification calculation shown in section

4.2.1. Because detected distance is relative distance, maximum detectable distance of 22.9 m is sufficient for CW system.

For simplicity, risk calculation is based on the following formula:

$$RISK = \begin{cases} 1, if\ x_{new} \leq OA\ (will\ collide) \\ 0, if\ x_{new} > OA\ (will\ not\ collide) \end{cases}$$

(29)

In image frame, y-axis is pointing downward and representing relative distance between host vehicle and preceding vehicle in term of pixel, x-axis remains unchanged:

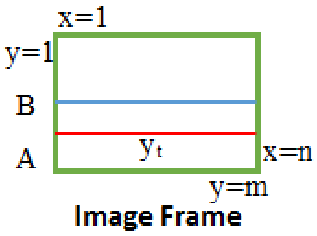


Figure 15: Image Frame and Coordinates

Formula (29) becomes:

$$RISK = \begin{cases} 1, if\ y_{new} \leq OA\ (will\ collide) \\ 0, if\ y_{new} > OA\ (will\ not\ collide) \end{cases}$$

(30)

In which:

$$y_{new} = y_{old} + vt + \frac{at^2}{2}$$

(31)

All parameters are calculated in pixels.

4.3.2. Object Detection by Matching - Spatial Correlation

Algorithm - Pseudo Code

- Step 1: Find image size *a x b* and template size *m x n*;
- Step 2: Find the boundary of region on image *f* that center *(x,y)* of template *w* can move to calculate NCE at this center;
- Step 3: Calculate NCE at each point on *f* that *w* moves through, using (1);
- Step 4: Determine the point that has highest NCE and the point's coordinates, this indicate the matched position of template *w* on image *f*.

MATLAB algorithm is shown in Appendix H - Object Detection by Matching - Spatial Correlation - MATLAB Codes.

Verify Algorithm

To verify algorithm, image *f* and template *w* are assumed to be the following matrices:

f =

1	4	3	4
5	**6**	**6**	**8**
9	**10**	**8**	**12**
9	**14**	**15**	**16**

w =

6	6	8
10	**8**	12
14	15	16

w is part of *f*. Based on the above algorithm, number of NCEs would be 4, matching coordinates would be *(3, 3)* on image *f*, and NCE would be 1 at this point.

NCE matrix from MATLAB:

0	0	0	0
0	0.8852	0.9362	0
0	0.8451	**1.0000**	0
0	0	0	0

4 NCEs are calculated, their values are 0.8852, 0.9362, 0.8451 and *1.0000*. Highest NCE is 1.0000 at coordinates (3, 3) on image *f*. It indicates that at this point template *w* completely matches with its coincident area on image *f*.

The algorithm is said to be correct.

Simulation with Captured Images and Results

Figure 16: Captured Image (Colour Image and Grayscale Intensity Image)

Figure 17: Cropped Template (Colour Image and Grayscale Intensity Image)

Figure 18: Detected Object (White Box) vs Targeted Object (Red Box)

The detected object has coordinates of (155, 79) and NCE = 0.3879. Since the template is part of original image, it is expected that NCE would be 1 instead of 0.3879 and coordinate of target would be (126, 259) (center of red box). In Figure 18, detected object is far from targeted object.

After reviewing the algorithm, the reason leading to inaccurate detected object is identified: the template used in the object detection algorithm should be extracted from the original image instead of cropping (by Snipping Tool in Windows) when viewing the image. The cropped image results in bigger template size than actual template extracted from the image. The cropped template (T2) has bigger size: 83x105 while extracted template (T1) has smaller size: 17x21. Additionally, extracted template is less detail compared to cropped template:

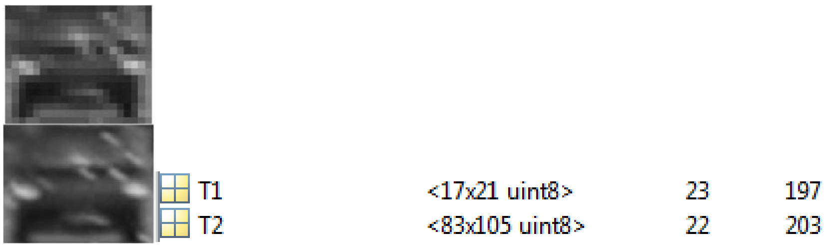


Figure 19: The Differences between Extracted Template (T1) and Cropped Template (T2)

Figure 20 shows result of the algorithm when using template extracted from the original image:

Figure 20: Detected Target (White Box) Using Template Extracted (Red Box) from Original Image

In Figure 20, red square indicates expected object to be detected while white box shows detected object using template extracted from the original image. The detection is correct and meets expectation: NCE_{max} = 1.0, center of detected object is at coordinates (126, 259). Using the same template for filtered image, object is detected at same coordinates with major drop in the value of NCE_{max} from 1.0 to 0.7614:

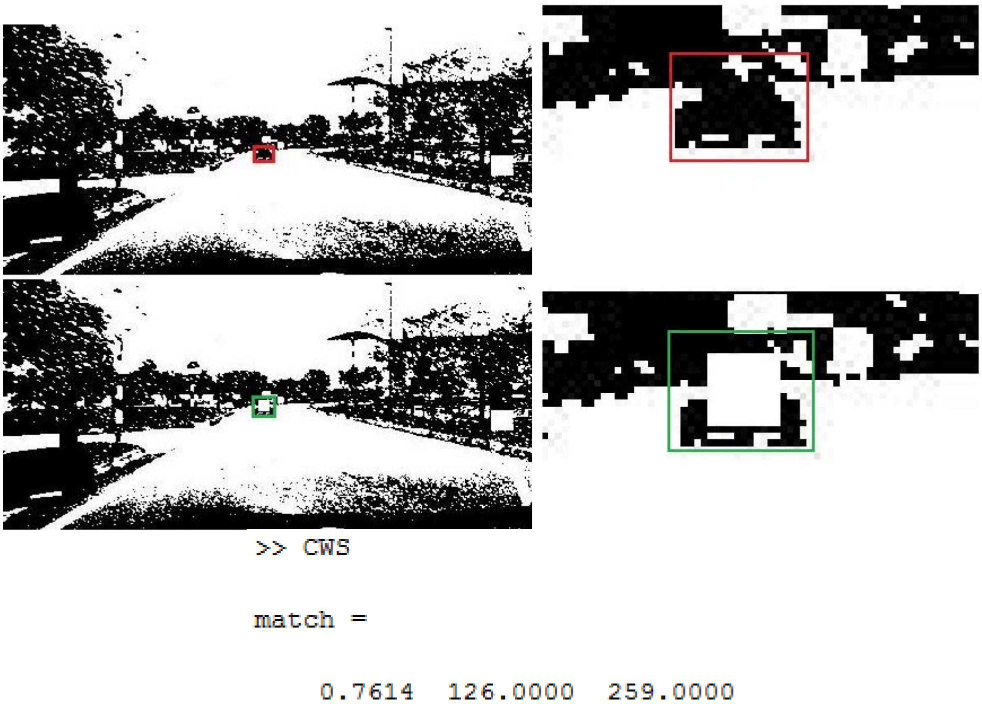


Figure 21: Result of Object Detection by Template Matching on Filtered Image

Template size affects accuracy of algorithm as well as processing time significantly. For image size of 248x524 pixels, when template size is 83x105 pixels, processing time is about 93 s meanwhile when template size is 17x21 pixels, processing time is only about 16 s (i.e. in Figure 18 & Figure 20). Oversize or undersize of template can result in inaccurate object detection, i.e. in Figure 22, $NCE_{max} = 0.5123$ and detected object is very far from target object.

```
>> CWS

match =

    0.5123  295.0000   89.0000
```

Figure 22: Inaccurate Object Detection due to Undersize Template: Expected Object (Red Box) & Inaccurate Detected Object (Yellow Box)

There are three drawbacks of object detection using template matching:

- Size of template greatly affects processing time and detection result, oversize or undersize of template could result in inaccurate object detection;
- For same object detection, at different distances, the algorithm requires different templates corresponding to targeted size of object in order to provide accurate object detection;
- Different object requires different template for matching, this will significantly increase time taken to detect an object if number of template increases.

In order to have accurate object detection using template matching, multiple templates of an object at different distances and multiple templates of multiple objects need to be used. However, the algorithm's speed will be affected significantly.

4.3.3. Object Detection by Background Subtraction , Thresholding and Structure

For object detection by background subtraction and thresholding, a sequence of images is needed to recognize object's movement. In this project, a sequence of 5 1080p images is used to detect road markings.

Algorithm - Pseudo Code

- Step 1: Preprocessing

  Convert color images to grayscale images; calculate the differences between two consecutive images in a sequence of five images; accumulate the four differences to form a single grayscale image; threshold the resulted image.

- Step 2: Road marking detection

  Perform road marking detection based on the nearest two road markings to the center of image.

- Step 3: ROI selection

  Left and the right boundaries of ROI are detected lane markings;

  Top boundary of ROI is the horizontal line in the middle of processing image and the bottom boundary of ROI is the bottom edge of the image.

- Step 4: Vehicle detection in ROI

  Convert the last image in the 5-image sequence to gray scale and perform vehicle detection in ROI;

  Highlight detected vehicle and display its coordinates for verification purpose.

MATLAB algorithm is shown in Appendix I - Object Detection by Background Subtraction, Thresholding and Structure - MATLAB Codes.

Results:

Figure 23, Figure 24, Figure 25, Figure 26, Figure 27 show results from road marking detection and detected target vehicle when running the algorithm in MATLAB. Preceding vehicle is detected accurately when it is in ROI. Table 7 shows coefficients of lines representing left and right boundaries of ROI and detected coordinates of preceding vehicle.

Table 7: ROI Boundary Lines' Slope (a) and y-Intercept (b) & Coordinates of Center of Detected Vehicle

Sequence	Left ROI		Right ROI		Detected Center	
	al	Bl	ar	br	y	x
1-5	-0.474	1299	0.509	296.3	999	890
7-11	-0.612	1364	0.553	200.3	1005	859
22-26	-0.527	1309	0.562	218.6	998	853
56-60	-0.544	1309	0.582	133.6	1030	820
98-102	-0.648	1381	0.590	15.0	1108	699

Figure 23: Detected Road Markings from Image Sequence 1-5 (Left), Detected Vehicle in Image 5 (Right)

Figure 24: Detected Road Markings from Image Sequence 7-11 (Left), Detected Vehicle in Image 11 (Right)

Figure 25: Detected Road Markings from Image Sequence 22-26 (Left), Detected Vehicle in Image 26 (Right)

Figure 26: Detected Road Markings from Image Sequence 56-60 (Left), Detected Vehicle in Image 60 (Right)

Figure 27: Detected Road Markings from Image Sequence 98-102 (Left), Detected Vehicle in Image 102 (Right)

The first three sets of results show detection of vehicle within the 1^{st} second because images are captured with frame rate of 30 fps. The fourth set is result during the 2^{nd} second and the last set is result during the 4^{th} second. The detection of preceding vehicle is accurate when ROI is correctly formed and it does not depend on point of time when the algorithm is run. Therefore the algorithm does not require a tracking algorithm to keep track of detected vehicle. Changes in coordinates of detected vehicle are useful for detection of speed, acceleration and trajectory of the vehicle. In MATLAB environment, times taken to get results for the five sequences are 10.53s, 11.00s, 12.02s, 12.43s, and 12.11s correspondingly. The average processing time is

11.62s for 1080p image. The feasibility to use inverted U-shape in detecting vehicle by structural method is proved.

The detection of touching line connecting vehicle's back wheels is expected to provide more accurate estimation of relative speed and acceleration between host and preceding vehicles. Therefore CW system can provide accurate result when estimating collision potential between the two vehicles.

Because of long processing time (i.e. 11.62 s per frame) the algorithm needs to be further developed and improved to reduce processing time.

4.3.4. Object Detection by Structure

A new genetic algorithm is developed based on the algorithm presented in section 4.3.3. It eliminates background subtraction and thresholding operations and only considers back wheel structure of preceding vehicle for vehicle detection. Pseudo code and results of the algorithms are shown below:

Algorithm - Pseudo Code

- Step 1: Preprocessing

  Preset ROI lines parameters are chosen from Table 7:

  al=-0.648; bl=1381;ar=0.590; br=15.0;

  Convert image to grayscale;

  Set pixels outside ROI to maximum intensity (i.e. 255).

- Step 2: Finding possible location of preceding vehicle

  Find isolated group of intensities having potential to represent vehicle based on intensity profile along x-axis of processing image.

- Step 3: Finding vehicle

  Based on inverted U-shape structure of vehicle's back wheels, incorporate isolated groups found in Step 2 with intensity profile along y-axis of processing image to find position of vehicle;

  Preceding vehicle's y coordinate is detected if along y-axis, intensity profile reach maximum (due to dark color of horizontal bar connecting two back wheel vehicles);

  Preceding vehicle's x coordinate is equal to average of coordinates of starting and ending points of isolated group found in Step 2.

- Step 4: Vehicle detection in ROI

  Highlight detected vehicle and save its coordinates for verification purpose.

MATLAB algorithm is shown in Appendix J - Modified Object Detection by Structure - MATLAB Codes.

Results:

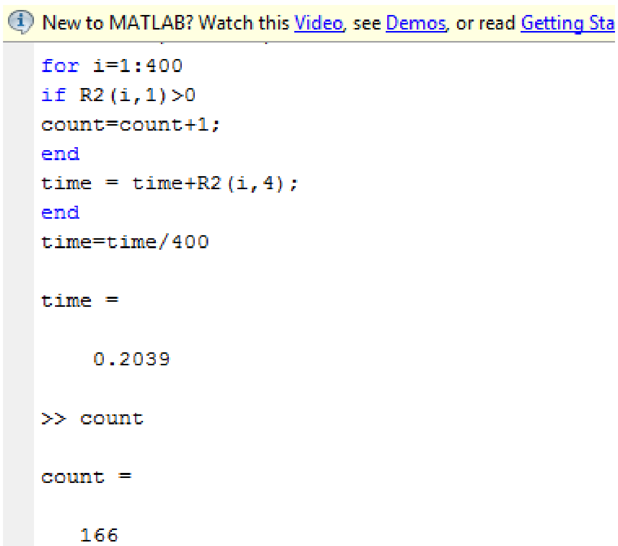
400 detection results are verified manually by reviewing saved result files on computer. Figure 28 shows successful detection of vehicle approaching from front. Figure 29 shows multiple vehicle detection results. Figure 30 shows calculation of detection rate and average processing time of the algorithm. Out of 400 frames, vehicles are detected in 166 frames, hence detection rate of the algorithm stands at 41.5%. Among 166 detected frames, 5 frames are false positives. 35 frames among 234 undetected frames are due to no appearance of preceding vehicle in ROI. Therefore algorithm's accuracy is:

$$\frac{166 - 5 + 35}{400} x100 = 49\ (\%) \quad (32)$$

It is noted that ROI parameters need to be adjusted to cover most of road surface to improve detection rate. Because camera is fixed on host vehicle, these parameters can be preset or adjusted only once. For 1080p image frame, per frame, average processing time is 0.2039s (98.25% reduction compared with the algorithm in section section 4.3.3) or about 4.90 fps. By processing only one image at a time instead of processing five images in previous algorithm, processing time is significantly improved. The algorithm is able to detect vehicle approaching from front and able to detect multiple vehicles in ROI even when host vehicle is changing its lane:

Figure 28: Detected Vehicle Approaching from Front

Figure 29: Multiple Vehicle Detection Feature



New to MATLAB? Watch this Video, see Demos, or read Getting Sta

```
for i=1:400
if R2(i,1)>0
count=count+1;
end
time = time+R2(i,4);
end
time=time/400

time =

    0.2039

>> count

count =

   166
```

Figure 30: Detection Rate and Average Processing Time per Frame

Table 8 shows risk calculation and estimation for the first 10 s (time interval of 0.5 s/frame) of a test video:

Table 8: Risk Calculation

Second	y (pixel)	v (pixel/s)	a (pixel/s^2)	y new (pixel)	RISK
0	0	0	0	0	0
	984	1968	**3936**	2460	1
1	996	24	**-3888**	522	0
	1006	20	-8	1015	0
2	1029	46	52	1058.5	0
	1039	20	-52	1042.5	0
3	1092	106	172	1166.5	1

	1018	-148	-508	880.5	0
4	1026	16	328	1075	0
	1035	18	4	1044.5	0
5	1065	60	84	1105.5	1
	1095	60	0	1125	1
6	1118	46	-28	1137.5	1
	1132	28	-36	1141.5	1
7	1142	20	-16	1150	1
	0	-2284	**-4608**	-1718	0
8	1141	2282	**9132**	3423.5	1
	1113	-56	**-4676**	500.5	0
9	1107	-12	88	1112	1
	1101	-12	0	1095	1

Bolded accelerations are outliers because they are extremely high compared to other values, hence they should not be used in calculation of risk. One reason leading to this extreme is because of undetected vehicle i.e. 2nd frame of second no. 7. Highlighted accelerations are considered high compared to other lower values. Based on results shown in Table 8, alarm should be generated for half of cases.

CHAPTER 5: RECOMMENDATIONS AND FUTURE WORKS

Object detection algorithm using template matching had been developed and tested using MATLAB. The algorithm is not accurate when detecting different types of cars at different distances. Because of some drawbacks of the algorithm such as the significant dependency of accuracy on size of template; distance to target object, and type of object, the algorithm is not used in this project. Object detection by structure is used as replacement of the above algorithm. Two genetic algorithms are developed: the first one uses a sequence of 5 images together with background subtraction and thresholding operations for object detection while the second one uses only one image and elimitates background subtraction operation for the same purpose. The latter improves up to 98.25% of processing time while maintaining same frame size. Additionally, risk calculation is done for a testing video. The project meets expectation and proposed deadlines. Current detection rate and accuracy of the algorithm stands at 41.5% and 49% respectively; further enhancement of the algorithms is needed to improve the detection rate and accuracy. It is suggested that ROI parameters are modified to cover most of road surface for higher rate of detection. Furthermore, Android-based mobile device might be considered for future development of the project.

CHAPTER 6: CONCLUSIONS

In this final year project, a CW system is proposed and designed. Two genetic algorithms are used for development of the system in MATLAB environment. The most reliable algorithm is used to implement in the system. A recorded video is used to feed into the system for vehicle detection, relative speed, acceleration estimation and risk calculation. Warning signals will be generated if a hazard exceeds acceptable risk limit preset by developer. To the date, the system is able to detect vehicle with frame rate of 4.90 fps for 1080p frame size. Aside from detecting preceding vehicle, the algorithm is capable of detecting vehicle approaching from front. The algorithm is also able to detect multiple vehicles even when the host vehicle is changing its lane. One review paper titled “A Review of Automated Emergency Braking System and the Trending for Future Vehicles” has been submitted and presented in “SAE 2013: Southeast Asia Safer Mobility Symposium”. Additionally, one research paper titled: “An Automobile Detection Algorithm Development for Automated Emergency Braking System” has been submitted to the 51st Design Automation Conference and got accepted for review. In the future, Android-based mobile device should be considered for the development of CW system.

REFERENCES

[1] MIROS. (2013). *General road accident data in Malaysia (1995 - 2010)*. Available: http://www.miros.gov.my/web/guest/road

[2] F. J. Martinez, T. Chai-Keong, J. C. Cano, C. T. Calafate, and P. Manzoni, "Emergency services in future intelligent transportation systems based on vehicular communication networks," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, pp. 6-20, 2010.

[3] Z. M. Jawi, M. H. M. Isa, R. Sarani, W. S. Voon, and A. F. M. Sadullah, "Weather as a road safety hazard in Malaysia - An overview," 2009.

[4] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 678 - 687, December 2009.

[5] E. Coelingh, A. Eidehall, and M. Bengtsson, "Collision warning with full auto brake and pedestrian detection - a practical example of automatic emergency braking," presented at the Annual Conference on Intelligent Transportation Systems, Funchal, 2010.

[6] A. Koelpin, G. Vinci, B. Laemmle, S. Lindner, F. Barbon, and R. Weigel. (2012) Six-Port technology for traffic safety. *IEEE Microwave Magazine*. 118-127.

[7] J. Jin, D. Kim, J. H. Song, V. D. Nguyen, and J. W. Jeon, "Hardware architecture design for vehicle detection using a stereo camera," presented at the 11th International Conference on Control, Automation and Systems (ICCAS), Gyeonggi-do, Korea, 2011.

[8] P. E. S. Bhd. (2012, June 28). *Proton Exora*. Available: http://www.proton-edar.com.my/Experience/Models/Exora/Overview.aspx

[9] Perodua. (2013, June 28). *Alza price*. Available: http://www.perodua.com.my/ourcars/alza/price/gha

[10] Perodua. (2013, June 18). *Myvi price*. Available: http://www.perodua.com.my/ourcars/myvi/price

[11] Perodua. (2013, June 28). *Viva price*. Available: http://www.perodua.com.my/ourcars/viva/price

[12] R. D. Inc. (2013, June 15). *High end scanning lasers and obstacle detectors*. Available: http://www.robotshop.com/high-end-lasers-obstacle-detectors.html?lang=en-us

[13] C. T. S. Bhd. (2013, June 15). *Laser range finder*. Available: http://cytron.com.my/listProductGroup.php?pid=PRY9EysWHScXLxwLEDcQGR0A!!!!!wTagIZ04QmaqPnXtZI=

[14] N. Vinh Dinh, N. Thuy Tuong, N. Dung Duc, L. Sang Jun, and J. Jae Wook, "A fast evolutionary algorithm for real-time vehicle detection," *IEEE Transactions on Vehicular Technology*, vol. 62, pp. 2453-2468, 2013.

[15] Thatcham. (2013). *Stop the crash*. Available: http://www.thatcham.org/files/pdf/stop_the_crash_AEB.pdf

[16] A. Group. (2011). *AEB Test Procedures - Autonomous Emergency Breaking*. Available: http://www.thatcham.org/safety/pdfs/AEB_test_procedures_nov11.pdf

[17] E. NCAP. (2012). *AEB fitment survey 2012*. Available: http://www.euroncap.com/results/aeb/survey.aspx

[18] Peugeot. (2013). *Driving control*. Available: http://www.peugeot.com/en/technology/safety/driving-control

[19] E. Ncap. (2013). *Audi Pre-Sense basic*. Available: http://www.euroncap.com/rewards/audi_pre_sense_basic.aspx
[20] Audi. (2013). *Audi Pre-Sense anticipatory safety*. Available: http://www.audi.com.au/au/brand/en/models/a6/a6_sedan/equipment/safety/audi_pre_sense.html
[21] E. Ncap. (2013). *Reward 2012 - Audi Pre-Sense Front Plus*. Available: http://www.euroncap.com/rewards/audi_pre_sense_front_plus.aspx
[22] Audi. (2011). *Driver assistance systems*. Available: http://www.audiusanews.com/newsrelease.do;jsessionid=708C7D93E7D9BDFF95056C98D268E7C4?&id=2194&allImage=1&teaser=driver-assistance-systems&mid=18
[23] BMW. (2013). *Safety*. Available: http://www.bmw.com/com/en/newvehicles/3series/sedan/2011/showroom/safety/index.html
[24] E. Ford Motor Company. (2012). *New Ford Fiesta 'Active City Stop' safety tech captures and processes 15 images in the blink of an eye*. Available: http://www.ford.co.uk/FordFleet/NewsAndReviews/FordForBusiness/2012/November/FiestaActiveCityStop
[25] R. C. S. Bhd. (2013). Available: http://malaysia.rs-online.com
[26] M. Schagrin. (2013). *Vehicle-to-vehicle (V2V) communications for fafety*. Available: http://www.its.dot.gov/research/v2v.htm
[27] Ü. Özguner. (2013). *Vehicle-to-vehicle communication for intelligent transportation*. Available: http://car.osu.edu/vehicle-vehicle-communication-intelligent-transportation
[28] G. C. Congress. (2013). *New roadside scattering model to improve vehicle-to-vehicle communication for intelligent transportation systems*. Available: http://www.greencarcongress.com/2013/07/v2v.html
[29] J.-P. Joosting. (2013). *Vehicle-to-vehicle communications to reach 62% by 2027*. Available: http://www.eetimes.com/document.asp?doc_id=1266622
[30] SICK. (2013). *Long range distance sensors* Available: http://www.sick.com/us/en-us/home/products/product_portfolio/distance_sensors/Pages/distance_sensors_long_range.aspx
[31] SICK. (2013). *Mid-range distance sensors*. Available: http://www.sick.com/group/EN/home/products/product_portfolio/distance_sensors/Pages/distance_sensors_mid_range.aspx
[32] SICK. (2013). *Short range distance sensors (displacement)*. Available: http://www.sick.com/group/EN/home/products/product_portfolio/distance_sensors/Pages/distance_sensors_short_range.aspx
[33] I. Dakota Digital. (2013, June). *Speedometers / Tachometers*. Available: http://www.dakotadigital.com/index.cfm/page/ptype=results/Category_ID=240/home_id=59/mode=cat/cat240.htm
[34] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. Upper Saddle River, New Jersey: Pearson Education Inc., 2010.
[35] N. I. Corporation. (2013). *Nearest neighbor*. Available: http://zone.ni.com/reference/en-XX/help/372916M-01/nivisionconcepts/classification_methods/
[36] D. Phillips, *Image Processing in C*. 1601 West 23rd Street, Suite 200, Lawrence, Kansas 66046-0127: R & D Publications, 2000.

[37] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*. Hungary: Academic Press, 2008.

[38] B. Alefs and D. Schreiber, "Accurate speed measurement from vehicle trajectories using AdaBoost detection and robust template tracking," presented at the IEEE Intelligent Transportation Systems Conference, Seattle, WA, 2007.

[39] T. K. Bharathi, S. Yuvaraj, D. S. Steffi, and S. K. Perumal, "Vehicle detection in aerial surveillance using morphological shared-pixels neural (MSPN) networks," in *Advanced Computing (ICoAC), 2012 Fourth International Conference on*, 2012, pp. 1-8.

[40] Y. Liu, X. Lu, and J. Xu, "Traffic scenes invariant vehicle detection," presented at the 9th Asian Control Conference (ASCC), Istanbul, 2013.

[41] C. Oh, S. G. Ritchie, and S.-T. Jeng, "Anonymous vehicle reidentification using heterogeneous detection systems," presented at the IEEE Transactions on Intelligent Transportation Systems, 2007.

[42] BOSCH. Selecting the right CCTV camera. Available: http://www.boschsecurity.us/NR/rdonlyres/1A4F9B44-0376-4FC8-A735-151F02021082/0/SelectingtheRightCCTVCamera.pdf

[43] C. Ltd. (2013). *CCTV camera buyers guide*. Available: http://www.cctv42.co.uk/cctv-camera-buying-guide.aspx

[44] BORSCHE. (2012). *Difference between CCTV camera and normal camera*. Available: http://www.borschecctv.com/en/news/Difference-between-CCTV-Camera-and-Normal-Camera-185.html#.UcQkwPnVDzI

[45] SENSR. (2013), Practical guide for accelerometers. Available: http://www.sensr.com/pdf/practical-guide-to-accelerometers.pdf

[46] B. LLC. (2013, June). *Digital Speedometer*. Available: https://play.google.com/store/apps/details?id=com.balabanimation.digitalspeedometer&hl=en

[47] IntelPR. (2013). *Acer and Intel join hands to introduce Acer Liquid C1 smartphone with Intel inside*. Available: http://newsroom.intel.com/community/intel_newsroom/blog/2013/01/31/acer-and-intel-join-hands-to-introduce-acer-liquid-c1-smartphone-with-intel-inside

[48] J. Turberville. (2003). *Camera field of view*. Available: http://www.jayandwanda.com/digiscope/vignette/camerafov.html

[49] J. Jongerius. (2013). *Measuring lens field of view (FOV)*. Available: http://www.panohelp.com/lensfov.html

APPENDICES

Appendix A - Summary of Laser Range Finders Prices and Specifications [12, 13]

Model	Unit Price	Range	Speed	Area Scanning Range	Resolution
RobotShop Distribution Inc.					
Hokuyo URG-04LX Scanning Laser Rangefinder	$2,375	20mm - 4,000mm	100ms/scan	240^o	0.36^o
Hokuyo URG-04LX-UG01 Scanning Laser Rangefinder	$1,174	20mm - 5,600mm	100ms/scan	240^o	0.36^o
Hokuyo UBG-04LX-F01 (Rapid URG) Scanning Laser Rangefinder	$2,850	20mm - 5,600mm	28ms/scan	240^o	0.36^o
Hokuyo UTM-30LX Scanning Laser Rangefinder	$5,590	100mm - 30,000mm	25ms/scan	270^o	0.25^o
Hokuyo UHG-08LX Scanning Laser Rangefinder	$3,945	20mm - 8,000mm	67ms/scan	270^o	0.36^o
Hokuyo PBS-03JN Scanning Infrared LED Obstacle Detection Sensor	$1,135	200mm - 3,000mm	100ms/scan	178.2^o	1.8^o
Hokuyo UBG-05LN Scanning Laser Obstacle Detection Sensor	$1,750	100mm - 5,000mm	100ms/scan	180^o	0.36^o
Hokuyo UTM-30LN Scanning Laser Rangefinder (30m)	$5,589	100mm - 30,000mm	NA	270^o	0.25^o
Hokuyo UXM-30LN Scanning Laser Rangefinder	$5,375	100mm - 30,000mm	50ms/scan	190^o	0.25^o
Loke 0.2 to 30m Industrial Laser Distance Meter (10Hz)	$2,250	200mm - 30,000mm	100ms/scan	NA	NA
Loke 0.2 to 35m Industrial Laser Distance Meter (10Hz)	$2,465	200mm - 35,000mm	100ms/scan	NA	NA
Loke 0.2 to 35m Industrial Laser Distance Meter (50Hz)	$2,750	200mm - 35,000mm	20ms/scan	NA	NA

Loke 0.2 to 35m Industrial Laser Distance Meter (WiFi)	$2,845	200mm - 35,000mm	NA	NA	NA
Parallax 15-122cm Laser Rangefinder	$129.99	150mm - 1,220mm (optimal)	1,000ms/scan	NA	NA
Cytron Technologies Sdn. Bhd.					
HOKUYO URG Laser Range Finder 4m URG-04LX-UG01	RM 4,857	Up to 5,600mm	100ms/scan	240^o	0.36^o
HOKUYO URG Laser Range Finder 4m URG-04LX	RM 9,062	Up to 4,000mm	100ms/scan	240^o	0.36^o
HOKUYO UTM Laser Range Finder 30m UTM-30LX	RM 20,662	Up to 30,000mm	25ms/scan	270^o	0.25^o
SICK Laser Measurement Sensor 80m LMS511-10100	RM 36,250	Up to 80,000mm	10ms/scan	190^o	NA
SICK Laser Measurement Sensor 80m LMS500-20000	RM 29,760	Up to 80,000mm	10ms/scan	190^o	NA
SICK Laser Measurement Sensor 20m LMS111-10100	RM 26,580	Up to 20,000mm	20ms/scan	270^o	0.25^o
SICK Laser Measurement Sensor 20m LMS100-10000	RM 22,500	Up to 20,000mm	20ms/scan	270^o	0.25^o

Appendix B - Starting Prices of Proton and Perodua's Cars [8-11]

Brand	Model	Starting Price
Proton	Exora	RM 59,548
	Exora Prime	RM 89,012
	Inspira	RM 78,549
	Persona	RM 46,499
	Prevé	RM 59,540
	Saga FLX	RM 38,148
	Satria Neo R3	RM 60,800
Perodua	Alza GHA/ZHA	RM 60,517
	Myvi SX/EZ	RM 41,924
	Viva 660 BX/850	RM 24,924

Appendix C - Summary of Outdoor Cameras' Prices and Specifications [25]

Model	Unit Price	Night Mode	Max Resolution (pixels)	Water Proof
ABUS Security-Center TVCC40000	RM 599	Yes	537x597	No
ABUS Security-Center TVCC74010	RM 2,407	Yes	795x596	No
ABUS Security-Center TVIP51550	RM 2,466	Yes	1280x1024	No
ABUS Security-Center TVIP51500	RM 2,164	Yes	1280x1024	No
RS (Bullet CCTV Camera)	RM 909	No	752x582	No
ABUS Security-Center TVCC74020	RM 3,011	Yes	742x552	No
ABUS Security-Center TVCC71500	RM 2,760	Yes	1028x596	No
ABUS Security-Center TVCC60020	RM 3,313	Yes	742x552	Yes
ABUS Security-Center TVCC71000	RM 2,577	Yes	1028x596	No
ABUS Security-Center TVCC40030	RM 1,384	Yes	795x596	No

Appendix D - Summary of Developed AEB Systems from Automobile Manufacturers

Brand	AEB Technology	Time to Collision	Speed When Avoid Collision	Model	Sensor		Deceleration Rate	Day/Night Operation	Consideration		Ref.
					Radar	Camera			Weather	Pedestrian	
Volvo	Volvo's Collision Warning with Brake Support – 2006			Volvo S80							[2]
	Collision Warning with Brake Assist – 2007			Volvo S80, V70, XC70							
	CWAB-PD 1^{st} Generation 2^{nd} Generation 3^{rd} Generation – 2011	1 s	35 km/h	MY Volvo S60	Long-range	640x480 CMOS	Max 5 m/s^2 Max 5 m/s^2 Max 10 m/s^2		No	Yes	[5]
Audi	Pre-Sense Plus	0.5 s					Max 5 m/s^2				[2]
	Pre-Sense Rear – 2011			Audi R8	Yes						
Ford	Collision Warning with Brake Support – 2009			Lincoln MKS, MKT Ford Taurus							
GM	Sixth Sense – 2005				GPS, LAN for V2V						
Honda	Collision Mitigation Brake System – 2003				Yes						
	Intelligent Night Vision System - 2004					Yes		Yes			
Mercedes-Benz	Pre-Safe - 2002										
	Pre-Safe Brake, Brake Assist Plus, Distronic Plus - 2005	0.6 s			Yes		Min: 0.4 g Max: Full brake				
Toyota	Pre-Collision System				Yes						
	Advanced Pre-Collision System				Yes	"Twin-lens"		Yes		Yes	

						stereo					
	Driver Monitoring System - 2007			Lexus LS		CCD					
				Toyota Crown - 2008							
				Toyota Crown Majesta - 2009	Yes					Yes	

Appendix E - AEB Technologies Available on Automobiles [17, 18]

Brand	Model	Technologies	City	Inter-Urban	Pedestrian	Standard
Audi	A1, A3, Q3, R8, RS Range, TT	NA				
	A3 (2012), A4, A5, Q5, A6, A7, A8, Q7	Pre Sense Front		Yes		
BMW	1 Series, 3 Series, X1, X3, X5, X6, Z4	NA				
	3 Series (2012), 5 Series, 6 Series, 7 Series	iBrake 3		Yes		
Ford	B-MAX, C-MAX	Active City Stop	Yes			
	Fiesta, Ka, Kuga	NA				
	Galaxy, Mondeo, S-MAX	Forward Collision Warning with Brake Support		Yes		
	Focus (2011)	Active City Stop, Forward Collision Warning with Brake Support	Yes	Yes		
Honda	Accord, Civic (2012), CR-V	Collision Mitigation Braking System		Yes		
	CR-Z, Insight, Jazz	NA				
Infinity	EX, FX, G37, M	Forward Collision Warning & Intelligent Brake Assist		Yes		Yes, Some Models
Jaguar	XJ	Advanced Emergency Brake Assist		Yes		
	XK	NA				
	XF	Advanced Emergency Brake Assist, Intelligent Emergency Braking		Yes		
Lexus	CT, GS, IS, RX	Pre Crash System		Yes		Yes, Some Models
	LS	Advanced Pre Crash		Yes	Yes	Yes,

		System				Some Models
Mazda	CX-5	Smart City Braking System	Yes			Yes, Some Models
	2, 3, 5, 6, CX-7, MX-5	NA				
Mercedes-Benz	R-Class	NA				
	C-Class, CLS-Class, E-Class, SL-Class, SLK-Class,	DISTRONIC PLUS (inc. PRE-SAFE Brake)	Yes	Yes		
	CL-Class, S-Class	DISTRONIC PLUS (inc. PRE-SAFE Brake)	Yes	Yes		Yes, Some Models
	A-Class (2012)	Collision Prevention Assist DISTRONIC PLUS		Yes Yes		Yes Yes, Some Models
	B-Class	Collision Prevention Assist DISTRONIC PLUS		Yes Yes		Yes No
	GL-Class (mid 2012), M-Class (Q3 2012)	Collision Prevention Assist DISTRONIC PLUS (inc. PRE-SAFE Brake)	 Yes	Yes Yes		Yes No
	G-Class (2012), GLK-Class (2012, facelift)	DISTRONIC PLUS		Yes		
Opel/Vauxhall	Agila, Ampera, Antara, Astra, Astra GTC, Corsa, Meriva, Zafira	NA				
	Insignia, Zafira Tourer	Front Camera + Emergency Braking		Yes		
Peugeot	All Models	NA Note: LDWS - Lane Departure Warning System				
Seat	Alhambra, Altea, Exeo, Ibiza, Leon	NA				
	Mii	Brake Assistant	Yes			

		Town				
Skoda	Fabia, Octavia, Roomster, Superb, Yeti	NA				
	Citigo	City Safe Drive	Yes			Yes, Some Models
Toyota	Auris, AYGO, GT86, iQ, RAV4, Urban Cruiser, Verso, Verso-S, Yaris	NA				
	Avensis, Land Cruiser, Prius	Pre-Crash System		Yes		Yes, Some Models
Volvo	C30, C70, S40, V50, XC90	NA				
	S60, S80, V40, V60, V70, XC60, XC70	CitySafety Collision Warning with Full Autobrake & Pedestrian Detection	Yes Yes	 Yes	 Yes	Yes No
Volks-wagen	up!	City Emergency Braking	Yes			Yes, Some Models
	Amarok, Beetle, Caddy, Eos, Golf, Golf Cabriolet, Jetta, Polo, Scirocco, Sharan, Tiguan, Touran	NA				
	Phaeton	Front Assist (Phaeton)		Yes		
	CC, Passat, Touareg	Front Assist		Yes		

Appendix F - Acer Liquid C1 Specifications

Table 9: Acer Liquid C1 Specifications [47]

Processor	Intel® Atom™ Z2420 (2x1.2GHz)
Modem	Intel® XMM™ 6265 slim modem with quad-band capability
Screen	4.3" IPS qHD touch screen (960x540, 16:9 ratio)
Operating System	Android* Ice Cream Sandwich
Main Camera	8 Megapixel at 5 frames per second, AF & LED Video Recording: 30 fps, 1080 x 1920 pixels
Front Camera	0.3 Megapixel
Internal Storage	4GB
Wireless LAN	802.11 b/g/n (/n HT20)
GPS / Bluetooth	GPS / Bluetooth® 2.1 with EDR
Microphone	Yes
Speaker	Integrated
MicroSD	32GB, SDHC 2.0 compatible
Battery	2,000 mAh
Micro-USB	Charging and PC Connect with USB 2.0
Dimensions	127.3 x 65.5 x 9.95 mm
Weight	140 g

Appendix G - List of AEB Systems & Related Technologies

Table 10: List of AEB Systems & Supplementary Technologies [17, 18]

Brand	AEB Systems & Supplementary Technologies
Audi	Pre-Sense Basic Pre-Sense Front Pre-Sense Front Plus Adaptive Cruise Control (ACC) with Stop & Go function Audi Side Assist + Pre-Sense Rear Active Lane Assist Night Vision Assistant
BMW	iBrake 3 Driving Assistant + Lane Departure Warning & Collision Warning Active Protection
Ford	Active City Stop
	Forward Collision Warning with Brake Support
Honda	Collision Mitigation Braking System
Infinity	Forward Collision Warning & Intelligent Brake Assist
Jaguar	Advanced Emergency Brake Assist
	Intelligent Emergency Braking
Lexus	Pre Crash System
	Advanced Pre Crash System
Mazda	Smart City Braking System
Mercedes-Benz	Pre-Safe Brake
	Collision Prevention Assist
	Distronic Plus
Opel/Vauxhall	Front Camera + Emergency Braking
Peugeot	LDWS - Lane Departure Warning System
Seat	Brake Assistant Town
Skoda	City Safe Drive
Toyota	Pre-Crash System
Volvo	CitySafety
	Collision Warning with Full Autobrake & Pedestrian Detection (CWAB-PD)
Volkswagen	City Emergency Braking
	Front Assist (Phaeton)
	Front Assist

Appendix H - Object Detection by Matching - Spatial Correlation - MATLAB Codes

```
time=cputime;
I = imread('Test1c.jpg'); %test image
I1 = rgb2gray(I); %grayscale

figure(1);

T = imread('T2cc.jpg'); %crop template
T2 = rgb2gray(T); %grayscale
subplot(2,1,1); imshow(T2);

T = zeros(17,21);
T1 = uint8(T);
for i=118:134
    for j=249:269
        T1(i-117,j-248) = I1(i,j); %extracted template in file
    end
end
subplot(2,1,2); imshow(T1);

%imwrite(T1,'T1g.jpg'); %write to file for storage
%T1 = imread('T1g.jpg'); %extracted templage in file
%T1=T2; %using crop template

figure(2);
subplot(2,1,1); imshow(I1);

f = double (I1);
w = double (T1);

%f =
[83,78,72,70,70,69,65;75,71,64,62,60,60,61;71,63,58,52,50,51,54;66,57,57,
52,44,44,46;61,54,53,57,50,46,45;67,54,47,46,50,53,48;75,61,50,46,48,48,4
6;75,61,51,45,49,47,45]
%w =
[57,52,44,44,46;53,57,50,46,45;47,46,50,53,48;50,46,48,48,46;51,45,49,47,
45]

%f = [91,86,82,79;83,78,72,70;75,71,64,62;71,63,58,52;71,63,58,52]
%w = [78,72,70;71,64,62;63,58,52]

[a,b] = size(f);
[m,n] = size(w);

tlx = (n-1)/2 +1;
tly = (m-1)/2+1;
brx = b-(n-1)/2;     %must be careful to avoid noninteger number
bry = a-(m-1)/2;     %must be careful to avoid noninteger number

temp = sum(w,2);
wbar = sum(temp)/(m*n)*ones(m,n);
wst = w-wbar;

wst2 = times(wst,wst);
temp = sum(wst2,2);
wstd = sum(temp);     %square denominator

wf = zeros(m,n);
gamma = zeros(a,b);
```

```
for y = tly:1:bry %row
    for x = tlx:1:brx %col
        %display(x);
        %display(y);
        %copy coincident image on f
        for i = 1:m %row
            for j = 1:n %col
                wf(i,j) = f(y-(m-1)/2-1+i,x-(n-1)/2-1+j);
            end
        end
        %display(wf);

        temp = sum(wf,2);
        wfbar = sum(temp)/(m*n)*ones(m,n);
        wfst = wf-wfbar;

        wfst2 = times(wfst,wfst);
        temp = sum(wfst2,2);
        wfstd = sum(temp);   %square denominator

        wwf = times(wst, wfst);
        temp = sum(wwf,2);

        num = sum(temp);
        den = sqrt(wstd*wfstd);

        gamma(y,x) = num/den;
    end
end

match = zeros(1,3);
for y=1:1:a
    for x=1:1:b
        if match(1,1)<gamma(y,x)
            match(1,1) = gamma(y,x);
            match(1,2) = y;
            match(1,3) = x;
        end
    end
end
display(match)

for i = -5:5
    for j = -5:5
        I1(match(1,2)+i, match(1,3)+j)=255;
    end
end

subplot(2,1,2); imshow(I1);
time=cputime-time;
fprintf('Processing time = %f s\n',time);
```

Appendix I - Object Detection by Background Subtraction, Thresholding and Structure - MATLAB Codes

CWS8.m (Vehicle Detection in ROI)

```
%Modified CWS8
%Vehicle detection in ROI

function [im, M, U, foundx, foundy] = CWS8(order, im, al, bl, ar, br)
%process only latest image with given lines for ROI on both sides

D21 = rgb2gray(im);
[m,n] = size(D21);  %row, col

for i=1:m/2-1 %remove upper region
    for j=1:n
        D21(i,j)=255;
    end
end
for i=m/2:m   %remove out of ROI region
    xEF = uint16((i-bl)/al); %left boundary
    xCD = uint16((i-br)/ar); %right boundary
    if xEF<=0
        xEF=1;
    end
    if xCD>n
        xCD=n;
    end
    if xEF>n
        xEF=n;
    end
    if xCD<=0
        xCD=1;
    end
    for j=1:xEF
        D21(i,j)=255;
    end
    for j=xCD:n
        D21(i,j)=255;
    end
end
imwrite(D21,'D21.jpg');

min = 255;        %find min, max intensity
max = 0;

for i=m/2:m
    xEF = uint16((i-bl)/al); %left boundary
    xCD = uint16((i-br)/ar); %right boundary
    if xEF<=0
        xEF=1;
    end
    if xCD>n
        xCD=n;
    end
    if xEF>n
        xEF=n;
    end
    if xCD<=0
        xCD=1;
    end
    for j=xEF:xCD
        if D21(i,j)<min
```

```
            min=D21(i,j);
        else
            if D21(i,j)>max
                max=D21(i,j);
            end
        end
    end
end

avg = (min+max)/2;  %average intensity

count=0;

for i=m/2:m   %process only pixel in ROI
    xEF = uint16((i-bl)/al); %left boundary
    xCD = uint16((i-br)/ar); %right boundary
    if xEF<=0
        xEF=1;
    end
    if xCD>n
        xCD=n;
    end
    if xEF>n
        xEF=n;
    end
    if xCD<=0
        xCD=1;
    end
    for j=xEF:xCD
        if (D21(i,j)>avg)
            D21(i,j)=255;   %highlight pixel with intensity higher than
average only
            count=count+1;  %number of highlighted points
        end
    end
end
%disp(min); disp(max);disp(avg);
%imshow(D21);

%rethreshold with mean intensity
mean=double(0); %if not double size, values will reach limit of int type
count=0;
for i=m/2:m
    xEF = uint16((i-bl)/al); %left boundary
    xCD = uint16((i-br)/ar); %right boundary
    if xEF<=0
        xEF=1;
    end
    if xCD>n
        xCD=n;
    end
    if xEF>n
        xEF=n;
    end
    if xCD<=0
        xCD=1;
    end
    for j=xEF:xCD
        mean=mean+double((D21(i,j)));
        count=count+1;
        %disp(count);
        %disp(mean);
    end
```

```
end
mean=uint16(mean/count);
%disp(mean);

for i=m/2:m
    xEF = uint16((i-bl)/al); %left boundary
    xCD = uint16((i-br)/ar); %right boundary
    if xEF<=0
        xEF=1;
    end
    if xCD>n
        xCD=n;
    end
    if xEF>n
        xEF=n;
    end
    if xCD<=0
        xCD=1;
    end
    for j=xEF:xCD
        if D21(i,j)<(2*mean/3)
            D21(i,j)=0;
        else
            D21(i,j)=255;
        end
    end
end

%imshow(D21);

M=zeros(m,8);

for i=m/2:m   %row
    xEF = uint16((i-bl)/al); %left boundary
    xCD = uint16((i-br)/ar); %right boundary
    if xEF<=0
        xEF=1;
    end
    if xCD>n
        xCD=n;
    end
    if xEF>n
        xEF=n;
    end
    if xCD<=0
        xCD=1;
    end
    for j=xEF:xCD
        if (D21(i,j)==0)
            M(i,1)=1; %mark row of interest
            count=1;
            k=j;
            while(1)
                count=count+1;
                k=k+1;
                if (k>xCD) || (D21(i,k)==255)
                    break;
                end
            end
            M(i,2)=uint16 (j+count/2);
            M(i,3)=uint16 (count);
            break;
        end
```

```
    end
    for j=xCD:-1:xEF
        if (D21(i,j)==0)
            M(i,1)=1; %mark row of interest
            count=1;
            k=j;
            while(1)
                count=count+1;
                k=k-1;
                if (k<xEF) || (D21(i,k)==255)
                    break;
                end
            end
            M(i,5)=uint16 (j-count/2);
            M(i,4)=uint16 (count);
            break;
        end
    end
end

%focus on verticle lines only
for i=m/2:m
    if M(i,1)==1
        if (M(i,5)-M(i,4)/2 - M(i,2)+M(i,3)/2) > (M(i,3)+M(i,4))
%clearance distance must > 2 tire's width
        else
            M(i,1)=2;
        end
    end
end
%%%%%%%%%%%%%%%%%%%focus on horizontal line only
for i=m/2:m
    if M(i,1)==2
        if M(i,5)<=M(i,2)
            %M(i,2)=uint16 ((M(i,2)+M(i,5))/2); M(i,5)=M(i,2);
M(i,3)=M(i,2); M(i,4)=M(i,2);
        else
            for j=1:5
                %M(i,j)=0;
            end
        end
    end
end
%%%%%%%%%%%%%%
count=0;
reset=0;
for i=m/2:m %count transition from horizontal line to verticle lines
            %very nice algorithm***
    if M(i,1)==2 && reset==0
        for j=i+1:m
            if M(j,1)==1 && reset==0
                count=count+1;
                reset=j;
            end
            if M(j,1)==2
                reset=0;
                break;
            end
        end
    end
end
```

```
U=zeros(2*count,7); %matrix to determin the inversed U shape (7
transitions 2->1, 14 rows needed)

count=0;
reset=0;
for i=m/2:m %count transition from horizontal line to verticle lines
            %very nice algorithm***
    if M(i,1)==2 && reset==0
        for j=i+1:m
            if M(j,1)==1 && reset==0
                count=count+1;
                U(count*2-1,1) = 2;
                U(count*2,1) = 1;
                U(count*2-1,2) = i; %last pos having horizontal bar
                U(count*2,2) = j; %1st pos has 1 (verticle line)
                reset=j;
            end
            if M(j,1)==2
                reset=0;
                break;
            end
        end
    end
end

%count no. of components
for k=1:count
    count2=1;
    %disp(U(k*2-1,1)-1);
    for i=U(k*2-1,2)-1:-1:m/2
        if M(i,1)==2
            count2=count2+1;
        else
            break;
        end
    end
    %disp(count2);
    U(k*2-1,3)=count2;

    count1=1;
    %disp(U(k*2,1)+1);
    for i=U(k*2,2)+1:m
        if M(i,1)==1
            count1=count1+1;
        else
            break;
        end
    end
    %disp(count1);
    U(k*2,3)=count1;
end
%find average coordinates and spans
for i=1:count
  for k=1:4
    sum2 = 0;
    for j=U(i*2-1,2):-1:(U(i*2-1,2)-U(i*2-1,3)+1)
        sum2 = sum2+M(j,1+k); %col 2 in M
    end
    U(i*2-1,3+k)=uint16 (sum2/U(i*2-1,3));
  end

  for k=1:4
    sum1 = 0;
```

```
    for j=U(i*2,2):(U(i*2,2)+U(i*2,3)-1)
        sum1 = sum1+M(j,1+k); %col 5 in M
    end
    U(i*2,3+k)=uint16 (sum1/U(i*2,3));
  end
end

%now process only U matrix only if count>0

if count>0
min=U(1,3);
max=U(1,3);
for i=1:count*2
    if U(i,3)>max
        max=U(i,3);
    end
    if U(i,3)<min
        min=U(i,3);
    end
end
avg = (min+max)/2;
end

for i=1:count*2
    if U(i,3)<avg
        U(i,3)=0;
    end
end
%determine nearest line
foundy=0; %if cant find, also need to return 0s
foundx=0;
for i=count*2:-1:1
    foundy=0;
    foundx=0;
    if U(i,3)>0 && U(i,1)==2
        if U(i+1,3)>0
            foundy=U(i+1,2)+U(i+1,3)-1;
            foundx=uint16 ((U(i+1,4)+U(i+1,7))/2);
            xEF=U(i+1,4)-U(i+1,5);
            xCD=U(i+1,7)+U(i+1,6);
            break;
        else
        for j=i-1:-1:1
            if U(j,3)>0 && U(j,1)==1
                foundy=U(j,2)+U(j,3)-1;
                foundx=uint16 ((U(j,4)+U(j,7))/2);
                xEF=U(j,4)-U(j,5);
                xCD=U(j,7)+U(j,6);
                break;
            end
        end
        end
    end
    if foundy>0
        break;
    end
end
%disp(foundx);disp(foundy);
%xEF = uint16((foundy-bl)/al); %left boundary
%xCD = uint16((foundy-br)/ar); %right boundary
if foundy>0
    if xEF<0
        xEF=1;
```

```
    end
    if xCD>n
        xCD=n;
    end
    if xEF>n
        xEF=n;
    end
    if xCD<=0
        xCD=1;
    end
end

if foundy>0 %only process if found object
    for i=xEF:xCD
        for j=-5:5
        im(foundy+j,i,1)=255;
        im(foundy+j,i,2)=0;
        im(foundy+j,i,3)=0;
        end
    end

    start1=1;
    for i=1:m
        xEF = uint16((double(i)-double(bl))/al); %left boundary
        xCD = uint16((double(i)-double(br))/ar); %right boundary
        %disp(xEF);disp(xCD);
        if (xEF==xCD) || (xCD == xEF-1) || (xEF == xCD-1) || (xCD == xEF-
2) || (xEF == xCD-2)
            start1=uint16(i);%disp(start1);
            break;
        end
    end

    for i=start1:m
        xEF = uint16((double(i)-double(bl))/al); %left boundary
        xCD = uint16((double(i)-double(br))/ar); %right boundary

        for j=1:5
            im(i,xEF+j,1)=0;
            im(i,xEF+j,2)=255;
            im(i,xEF+j,3)=0;

            im(i,xCD+j,1)=0;
            im(i,xCD+j,2)=255;
            im(i,xCD+j,3)=0;
        end
    end
    %imshow(im);
end

if foundy==0
    start1=1;
    for i=1:m
        xEF = uint16((double(i)-double(bl))/al); %left boundary
        xCD = uint16((double(i)-double(br))/ar); %right boundary
        %disp(xEF);disp(xCD);
        if (xEF==xCD) || (xCD == xEF-1) || (xEF == xCD-1) || (xCD == xEF-
2) || (xEF == xCD-2)
            start1=uint16(i);%disp(start1);
            break;
        end
    end
    disp(start1);
```

```
    for i=start1:m
        xEF = uint16((double(i)-double(bl))/al); %left boundary
        xCD = uint16((double(i)-double(br))/ar); %right boundary
        for j=-5:5
            im(i,xEF+j,1)=0;
            im(i,xEF+j,2)=255;
            im(i,xEF+j,3)=0;

            im(i,xCD+j,1)=0;
            im(i,xCD+j,2)=255;
            im(i,xCD+j,3)=0;
        end
    end
end
imwrite(im,[int2str(order) 'result.jpg']);
%imshow(im);
end
%imshow(im);
```

line8.m (Road Marking Detection)

```
%detect very well road lines

function [D21,al,bl,ar,br] = line8(order,im1,im2,im3,im4,im5)

im1=rgb2gray(im1);
im2=rgb2gray(im2);
im3=rgb2gray(im3);
im4=rgb2gray(im4);
im5=rgb2gray(im5);

i21=im2-im1;
i32=im3-im2;
i43=im4-im3;
i54=im5-im4;

%{
i21=im1-im2;
i32=im2-im3;
i43=im3-im4;
i54=im4-im5;
%}

%imshow(i32+i21+i43+i54); %accumulated differences of 5 consecutive
images
%thresholding
id = i21+i32+i43+i54; %imshow(id);
[m,n]=size(id);

%initialization if cant find, will use the values
al=-1;
ar=1;
bl=m+1;
br=m-n;

%thresholding
for i=m/2:m-1 %processing lower half of image only
    min=id(i,1);
    max=id(i,1);
    for j=1:n
        if id(i,j)<min
            min=id(i,j);
        end
        if id(i,j)>max
            max=id(i,j);
        end
    end
    avg=(min+max)/2;
    for j=1:n
        if id(i,j)>avg
            id(i,j)=255;
        else
            id(i,j)=0;
        end
    end
    %remove isolated points
    for j=2:n-1
        if id(i,j-1)==0 && id(i,j+1)==0
            id(i,j)=0;
        end
        if id(i-1,j)==0 && id(i+1,j)==0
```

```
            id(i,j)=0;
        end
    end
end
%imshow(id);

D=id;
D21=id;
%new algorithm

for i=m/2:m
    min = 0;        %find min, max intensity (locally)
    max = 0;
    for j=1:n
        if D(i,j)<min
            min=D(i,j);
        else
            if D(i,j)>max
                max=D(i,j);
            end
        end
    end
    avg = (min+max)/2;  %average intensity
    for j=1:n
        if D(i,j)>avg
            D(i,j)=255;
        end
    end
end

%{
% find mean intensity
for i=m/2:m
    mean = 0;        %find min, max intensity (locally)
    count = 0;
    for j=1:n
        if D(i,j)>0
            mean=mean+D(i,j);
            count=count+1;
        end
    end
    mean=mean/count;  %mean intensity
    for j=1:n
        if D(i,j)>mean
            D(i,j)=255;
        end
    end
end
%}

%{
%find outside in
M=zeros(m,4);
for i=1:m
    for j=1:n %left edge
        if D(i,j)==255
            M(i,1)=j;
            break;
        end
    end
    for j=n:-1:1 %right edge
        if D(i,j)==255
            M(i,3)=j;
```

```
            break;
        end
    end
end
%}

%find inside out
M=zeros(m,4);
for i=1:m
    for j=n/2:-1:1 %left edge
        if D(i,j)==255 %take middle point
            count=1;
            k=j;
            while(1)
                count=count+1;
                k=k-1;
                if (k<1) || (D(i,k)<255)
                    break;
                end
            end
                M(i,1)=uint16 (j-count/2);
            break;
        end
    end
    for j=n/2:n %right edge
        if D(i,j)==255 %take middle point
            count=1;
            k=j;
            while(1)
                count=count+1;
                k=k+1;
                if (k>n) || (D(i,k)<255)
                    break;
                end
            end
                M(i,3)=uint16 (j+count/2);
            break;
        end
    end
end

%2nd derivative
for i=2:m-1
    if (M(i-1,1)>0) && (M(i+1,1)>0)
        M(i,2)=M(i-1,1)+M(i+1,1)-2*M(i,1);
    end
end

for i=2:m-1
    if (M(i-1,3)>0) && (M(i+1,3)>0)
        M(i,4)=M(i-1,3)+M(i+1,3)-2*M(i,3);
    end
end

%set limit = 10% of original value, mark abrupt change if exceeding limit
for i=1:m
    if abs(M(i,2))>0.1*M(i,1)
        M(i,2)=0;M(i,1)=0;
    end
end

for i=1:m
    if abs(M(i,4))>0.1*M(i,3)
```

```
        M(i,4)=0;M(i,3)=0;
    end
end

%find verticle points (same x)
for i=1:m-1
    if M(i,1)==M(i+1,1)
        M(i,2)=1;M(i+1,2)=1;
    end
    if M(i,3)==M(i+1,3)
        M(i,4)=1;M(i+1,4)=1;
    end
end
%remove vertical points
for i=1:m
    if M(i,2)==1
        M(i,1)=0; M(i,2)=0;
    end
    if M(i,4)==1
        M(i,3)=0; M(i,4)=0;
    end
end

%merging lines
for i=1:m
    if M(i,1)>n/2
        M(i,1)=0;
    end
    if M(i,3)<n/2
        M(i,3)=0;
    end
end

%connecting points
for i=2:m-1
    if (M(i,1)==0) && M(i-1,1)>0 && M(i+1,1)>0 && (((M(i+1,1)-M(i-
1,1))/(M(i-1,1)))<0.1)
        M(i,1)=uint16 (M(i-1,1)+M(i+1,1))/2;
    end
    if (M(i,3)==0) && M(i-1,3)>0 && M(i+1,3)>0 &&(((M(i+1,3)-M(i-
1,3))/(M(i-1,3)))<0.1)
        M(i,3)=uint16 (M(i-1,3)+M(i+1,3))/2;
    end
end

%remove isolated points
for i=2:m-1
    if (M(i-1,1)==0) && (M(i+1,1)==0)
        M(i,1)=0;
    end
    if (M(i-1,3)==0) && (M(i+1,3)==0)
        M(i,3)=0;
    end
end
if (M(1,1)>0) && (M(2,1)==0)
    M(1,1)=0;
end
if (M(m,1)>0) && (M(m-1,1)==0)
    M(m,1)=0;
end

%clear cols 2 & 4
for i=1:m
```

```
    M(i,2)=0; M(i,4)=0;
end

%plot(M);

%find best fit lines
cc=0;   %count components
cg=0;   %count groups
for i=2:m
    if (M(i,1)>0) && (M(i-1,1)==0)
        cg=cg+1;    %count groups
    end
end
cl = zeros(cg,6);   %store info of groups & no. of comps

cg=0;
for i=2:m-1
    if (M(i,1)>0) && (M(i-1,1)==0)
        cg=cg+1;    %count groups
        cl(cg,1)=i; %store pos
        cc=0;
    end
    if M(i,1)>0
        cc=cc+1;
    end
    if (cg>0) && (M(i,1)>0) && (M(i+1,1)==0)
        cl(cg,2)=cc;%store no. of comps
    end
end
%find line coefficients
for i=1:cg
    temp = zeros(cl(i,2),2);
    for j=1:cl(i,2)
        temp(j,1)=M(cl(i)-1+j,1);    %x
        temp(j,2)=cl(i,1)+j-1;       %y
    end
    [cl(i,3),cl(i,4)]=bestfitline(temp);    %best fit line
end
for i=1:cg
    if cl(i,3)>=0
        cl(i,3)=0; cl(i,4)=0;   %ignore alpha > 0 for left line
    end
end

%find lines with intersection point on lower edge most close to middle
%verticle line

%find max no. of continuous points
max=0;
for i=1:cg
    if cl(i,2)>max
        max=cl(i,2);
    end
end

for i=1:cg
    if (cl(i,3)>0) || (cl(i,2)<=max/2) %remove lines with alpha > 0 %50%
max continuous points and above
        cl(i,3)=0; cl(i,5)=0;
    end
end
```

```
for i=1:cg
    if (cl(i,3)<0) %50% max continuous points and above
        cl(i,5) = (m - cl(i,4))/cl(i,3);
        max = cl(i,5);
    end
end

for i=1:cg
    if cl(i,5)<-n/2 %intersection with lower edge exceed 2 times image
width
        cl(i,3)=0;
    end
end

%find probability of found line matching with background, +- 5% points
left
%& right
limit=uint16 (0.05*n);  %5% of width
for i=1:cg
    matchl=0;     %match pixel
    countl=0;
    if cl(i,3)<0
    for j=m/2:m %y
        k=uint16 ((j - cl(i,4))/cl(i,3)); %x
        %20 points left & right side of found line
        if (k>0) && (k<=n)
            if (k-limit)>0 %left side
                for l=1:limit
                    countl=countl+1;
                    if D21(j,k-l)==255
                        matchl=matchl+1;
                    end
                end
            else
                for l=1:(k-limit)
                    countl=countl+1;
                    if D21(j,k-l)==255
                        matchl=matchl+1;
                    end
                end
            end
            if (k+limit)<=n %right side
                for l=1:limit
                    countl=countl+1;
                    if D21(j,k+l)==255
                        matchl=matchl+1;
                    end
                end
            else
                for l=1:(n-k)
                    countl=countl+1;
                    if D21(j,k+l)==255
                        matchl=matchl+1;
                    end
                end
            end
        end
    end
    %disp(matchl);
    if countl>0
    cl(i,6)=matchl/countl;  %probability line matching
    end
    end
```

```
end

mark=n/2;
for i=1:cg
    if (cl(i,3)<0) && (cl(i,5)>max)
        max=cl(i,5);
        mark=uint16 (max);
    else
        mark=max;
    end
end

for i=1:cg
    if cl(i,5)==mark
    else
        cl(i,3)=0; cl(i,5)=0;
    end
end

%highlight ROI
for i=1:cg
    if (cl(i,3)<0)
        for k=m/2:m
            j = (k-cl(i,4))/cl(i,3);      %x=(y-b)/a
            j = uint16 (j);
            if (j==0) || (j>n)
                j=1;
            end
            %disp(j);
            for l=-5:5
            D21(k,j+l)=255;
            end
        end
        al = cl(i,3);
        bl = cl(i,4);
    end
end
%imshow(D21);

cg=0;
cc=0;
for i=2:m
    if (M(i,3)>0) && (M(i-1,3)==0)
        cg=cg+1;    %count groups
    end
end
cr = zeros(cg,6);   %store info of groups & no. of comps

cg=0;
for i=2:m-1
    if (M(i,3)>0) && (M(i-1,3)==0)
        cg=cg+1;    %count groups
        cr(cg,1)=i; %store pos
        cc=0;
    end
    if M(i,3)>0
        cc=cc+1;
    end
    if (cg>0) && (M(i,3)>0) && (M(i+1,3)==0)
        cr(cg,2)=cc;%store no. of comps
    end
end
```

```
%find line coefficients
for i=1:cg
    temp = zeros(cr(i,2),2);
    for j=1:cr(i,2)
        temp(j,1)=M(cr(i)-1+j,3);    %x
        temp(j,2)=cr(i,1)+j-1;       %y
    end
    [cr(i,3),cr(i,4)]=bestfitline(temp);    %best fit line
end
for i=1:cg
    if cr(i,3)<=0
        cr(i,3)=0; cr(i,4)=0;   %ignore alpha <= 0 for right line
    end
end

%find probability of found line matching with background, +- 5% points
left
%& right
limit=uint16 (0.05*n);
for i=1:cg
    matchr=0;    %match pixel
    countr=0;
    if cr(i,3)>0
    for j=m/2:m %y
        k=uint16 ((j - cr(i,4))/cr(i,3)); %x
        %20 points left & right side of found line
        if (k>0) && (k<=n)
            if (k-limit)>0 %left side
                for l=1:limit
                    countr=countr+1;
                    if D21(j,k-l)==255
                        matchr=matchr+1;
                    end
                end
            else
                for l=1:(k-limit)
                    countr=countr+1;
                    if D21(j,k-l)==255
                        matchr=matchr+1;
                    end
                end
            end
            if (k+limit)<=n %right side
                for l=1:limit
                    countr=countr+1;
                    if D21(j,k+l)==255
                        matchr=matchr+1;
                    end
                end
            else
                for l=1:(n-k)
                    countr=countr+1;
                    if D21(j,k+l)==255
                        matchr=matchr+1;
                    end
                end
            end
        end
    end
    %disp(matchr);
    if countr>0
    cr(i,6)=matchr/countr;  %probability line matching
    end
    end
```

```
end

%find lines with intersection point on lower edge most close to middle
%verticle line

max=0;
for i=1:cg
    if cr(i,2)>max
        max=cr(i,2);
    end
end

for i=1:cg
    if (cr(i,3)<0) || (cr(i,2)<=max/2) %remove lines with alpha < 0 %50%
max continuous points and above
        cr(i,3)=0; cr(i,5)=0;
    end
end

for i=1:cg
    if (cr(i,3)>0) %%50% max continuous points and above
        cr(i,5) = (m - cr(i,4))/cr(i,3);
        min = cr(i,5);
    end
end

for i=1:cg
    if cr(i,5)>3*n/2 %intersection with lower edge exceed 2 times image
width
        cr(i,3)=0;
    end
end

mark=n/2;
for i=1:cg
    if (cr(i,3)>0) && (cr(i,5)<min)
        min=cr(i,5);
        mark=uint16 (min);
    else
        mark=min;
    end
end

for i=1:cg
    if cr(i,5)==mark
    else
        cr(i,3)=0; cr(i,5)=0;
    end
end

%highlight ROI
for i=1:cg
    if (cr(i,3)>0)
        for k=m/2:m
            j = (k-cr(i,4))/cr(i,3);     %x=(y-b)/a
            j = uint16 (j);
            if (j==0) || (j>n)
                j=1;
            end
            %disp(j);
            for l=-5:5
            D21(k,j+l)=255;
```

```
            end
        end
        ar = cr(i,3);
        br = cr(i,4);
    end
end
%disp(al);
%disp(bl);
%disp(ar);
%disp(br);
%imshow(D21);
imwrite(D21,[int2str(order) 'line.jpg']);
%imshow(D21);
end
```

bestfitline.m (Find best fit line based on line coefficients data set)

```
function [m,b] = bestfitline(M)
%find best fit line of input M
%ref:http://hotmath.com/hotmath_help/topics/line-of-best-fit.html
n = size(M,1);  %no. of samples
s = sum(M,1);   %sum of element in cols in M
sx = s(1);      %sumx
sy = s(2);      %sumy
p = prod(M,2);  %prod xy element
sxy = sum(p);   %sum xy
sxx = 0;
for i=1:n
    sxx = sxx+M(i,1)*M(i,1);  %sum xx
end
m = (sxy-sx*sy/n)/(sxx-sx*sx/n);
b = sy/n - m*sx/n;
end
```

Appendix J - Modified Object Detection by Structure - MATLAB Codes

CWS11.m

```
%CWS11 for submission
function [N, R, foundx, foundy, confidence, time] = CWS10(order, im, al,
bl, ar, br)
time=cputime;

foundx=0;
foundy=0;
confidence=0;

%clear all;
%im=imread('Test4.jpg');
D21=rgb2gray(im);

[m,n] = size(D21);  %row, col

start1=1; %start row of consideration (intersection of 2 lines)
end1=m; %end row of consideration (when either left or right ROI reaches
edge of image

for i=1:m
    xEF = uint16((double(i)-double(bl))/al); %left boundary
    xCD = uint16((double(i)-double(br))/ar); %right boundary
    %disp(xEF);disp(xCD);
    if (xEF==xCD) || (xCD == xEF-1) || (xEF == xCD-1)
        start1=uint16(i); %disp(start1);
        break;
    end
end

for i=start1+1:m
    xEF = uint16((double(i)-double(bl))/al); %left boundary
    xCD = uint16((double(i)-double(br))/ar); %right boundary
    if (xEF==0) || (xCD==0)
        end1=uint16(i)-1; %disp(end1);
        break;
    end
end

for i=1:start1-1 %remove upper region
    for j=1:n
        D21(i,j)=255;
    end
end

for i=start1:end1   %remove out of ROI region
    %disp(i);disp(bl);disp(i-bl);disp(al);disp(double((i-bl)/al));
    xEF = uint16((double(i)-double(bl))/al); %left boundary
    xCD = uint16((double(i)-double(br))/ar); %right boundary
    if xEF<=0
        xEF=1;
    end
    if xCD>n
        xCD=n;
    end
    if xEF>n
        xEF=n;
    end
    if xCD<=0
```

```
        xCD=1;
    end
    for j=1:xEF
        D21(i,j)=255;
    end
    for j=xCD:n
        D21(i,j)=255;
    end
end

temp=D21;
[m,n] = size(temp);
X=zeros(1,n); %profile along x
Y=zeros(m,1); %profile along y
levels=4; %levels of intensities

for i=1:n %profile along x
    count=0;
    for j=start1:m
        if temp(j,i)<=255/levels
            count=count+1;
        end
    end
    X(1,i)=-count;
end
%figure(1);plot(X);title('Intensity along Y');

for i=start1:m %profile along y
    count=0;
    for j=1:n
        if temp(i,j)<=255/levels
            count=count+1;
            D21(i,j)=0;
        else
            D21(i,j)=255;
        end
    end
    Y(i,1)=count;
end

%processing profile
X2=zeros(3,n);
for i=2:n-1
    X2(1,i)=X(1,i);
    X2(2,i)=X(1,i+1)-X(1,i); %1st derivative
    X2(3,i)=X(1,i-1)+X(1,i+1)-2*X(1,i); %2nd derivative
end

count=0; %count signature groups
for i=2:n-1
    if X2(1,i)==0 && X2(1,i+1)<0
        count=count+1;
    end
end
%disp(count);

N=zeros(count,n); %store signatures
count=0;
for i=2:n-1
    if X2(1,i)==0 && X2(1,i+1)<0 %start signature
        count=count+1;
        N(count,1)=i; %1 col before
    end
```

```
    if X2(1,i)<0 && X2(1,i+1)==0 %end signature
        N(count,2)=i+1; %1 col after
    end
end

%figure(2);title('Intensity along Y');
for i=1:count
    Y=zeros(m,1);
    max=0;mark=0;
    for j=start1:m %profile along y
        count2=0;
        for k=N(i,1):N(i,2)
            if temp(j,k)<=255/levels
                count2=count2+1;
            end
        end
        Y(j,1)=count2;
        if max<count2
            max=count2; mark=j;
        end
    end
%     subplot(count,1,i); plot(Y);

    N(i,3)=mark; %mark y pos max intensity profile
    N(i,4)=max; %max profile

    %within 10% of max intensity profile, find the mark (line) nearest to
    %host vehicle
    for j=m:-1:start1
        if Y(j,1)/max >= 0.9
            N(i,3)=j; break;
        end
    end

    %confidence: if width of signature = max profile, then confidence =
    %100%
    N(i,5)=max/(N(i,2)-N(i,1)-1); %confidence

    %consider width and pos of vehicle (intensity profile) with respect
to y
    xEF = uint16((double(N(i,3))-double(bl))/al); %left boundary
    xCD = uint16((double(N(i,3))-double(br))/ar); %right boundary
    if N(i,1)>xEF && N(i,1)<xCD && N(i,2)>xEF && N(i,2)<xCD
        N(i,6)=double(N(i,4))/double(xCD-xEF);
    else
        N(i,6)=0;
    end
    %overall confidence
    N(i,7)=N(i,5)*N(i,6);
end

presetconfidence = 0.10;
count2=0; %count no. of detected vehicles
for i=1:count
    if N(i,7)<presetconfidence
    else
        count2=count2+1;
    end
end

R=zeros(count2,4); %result matrix
count2=0; %count no. of detected vehicles
```

```
for i=1:count
    if N(i,7)<presetconfidence
    else
        count2=count2+1;
        R(i,1)=N(i,7); %copy confidence
        R(i,2)=N(i,3); %copy detected row
        R(i,3)=uint16((N(i,1)+N(i,2))/2); %detected col
        R(i,4)=N(i,2)-N(i,1)-1; %width of vehicle
    end
end
disp(count2);
%highlight results
if count2>0
    for i=1:count2
        if R(i,2)>0 && R(i,3)>0
            foundx=R(i,3);disp(foundx);
            foundy=R(i,2);disp(foundy);
            confidence=R(i,1);
            break;
        end
    end
end

disp(count2);

for i=1:count2
  if R(i,2)>0 && R(i,3)>0
    for j=-5:5
        range=uint16(R(i,4)/2);
        range=double(range);
        for k=-range:range
            im(j+R(i,2),k+R(i,3),1)=255;
            im(j+R(i,2),k+R(i,3),2)=0;
            im(j+R(i,2),k+R(i,3),3)=0;
        end
    end
    if foundy>R(i,2)
        foundy=R(i,2);
        foundx=R(i,3);
        confidence=R(i,1);
    end
  end
end

%figure(1);subplot(2,2,1);plot(X);title('Intensity along
X');subplot(2,2,2);plot(Y);title('Intensity along Y');
%figure(1);subplot(2,2,3);imshow(D21);subplot(2,2,4);imshow(temp);
%figure(3);imshow(im);

time=cputime-time;

%highlight ROI in resulted image
for i=start1:end1
    xEF = uint16((double(i)-double(bl))/al); %left boundary
    xCD = uint16((double(i)-double(br))/ar); %right boundary
    for j=-5:5
        im(i,xEF+j,1)=0;
        im(i,xCD+j,1)=0;
        im(i,xEF+j,2)=255;
        im(i,xCD+j,2)=255;
        im(i,xEF+j,3)=0;
        im(i,xCD+j,3)=0;
    end
```

```
end

if confidence>0
    imwrite(im,['C:\Users\Tran Duc Chung\Desktop\CWS
Matlab\Results\CWS11\Detection\' int2str(order) 'result.jpg']);
else
    imwrite(im,['C:\Users\Tran Duc Chung\Desktop\CWS
Matlab\Results\CWS11\No Detection\' int2str(order) 'result.jpg']);
end
%imwrite(D21,[int2str(order) 'D21.jpg']);
end
```