UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ____

____

____

Academic Session: _____________

I ____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ _________________________

(Author's signature) (Supervisor's signature)

Address:

____

____

____

____

Supervisor's name

Date: ____________________ **Date**: ____________________

Implementing Lane Departure Warning System on Raspberry Pi

By

CHEE WINSERNG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

In partial fulfillment of the requirement

for the degree of

BACHELOR OF INFORMATION AND COMMUNICATION TECHNOLOGY (HONS)
COMMUNICATION AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

May 2016

DECLARATION OF ORIGINALITY

I declare that this report entitled "______________________________" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ________________________

Name : ________________________

Date : ________________________

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest appreciation to my supervisor, Dr. Lau Phooi Yee for all the useful comments and ideas which motivate me continuous support in my Final Year Project. Without her supervision and constant help this project would not have been possible as digital image processing is not in my study syllabus. I offer my sincere appreciation for the learning opportunities which is a bonus for my study.

My grateful thanks are also extended to my moderator, Dr Robithoh Annur, who encourage me and enlighten me in various aspect.

Last but not the least, many thanks to my parents and my friend for their continuous support to keep me in track and help in solving all the challenges that I have face in this trimester.

ABSTRACT

The main purpose of this project is to develop a real-time Lane Departure Warning System (LDWS) that able to be implemented on a Controller Area Network (CAN) Bus. As part of Automatic Driver Assistance System (ADAS), LDWS exist to alert the driver when a car is diverting from its original lane. The LDWS require 3 major operation, being 1) retrieve video input and detect the propagating lane. 2) obtain and track the current position of the vehicle. 3) provide an alert system to maintain the driver's speed and driving direction.

Raspberry Pi is the primary platform that this system will implement on. It has various Input/output (I/O) ports that allow developer to utilize and many module and component have been design for it, such as the Raspberry Pi Camera Module which will be used for the system purposed.

There are several problem which will be solve by the system. First of all, the road condition such as weather and low light condition that will cause the system not able to detect the lane should be addressed with several image processing technique. Other than that, this system should be portable and compact enough to be installed on the rear of a windshield mirror and implemented on a vehicle.

Contents

LISTS OF FIGURES

LISTS OF ABBREVIATIONS

LDWS	Lane Departure Warning System
ADAS	Automatic Driver Assistance System
CAN	Controller Area Network
HE	Histogram Equalization
HE-VED	Histogram Equalization Variable Enhancement Degree
I/O	Input/output
PWM	Pulse Width Modulation

1. INTRODUCTION

Henry Ford mentioned in one of his speeches, [1] 'If I had asked people what they wanted, they would have said faster horses.' Since then, car have been developed and improved for centuries from steam-powered vehicle, internal combustion engine vehicle to today hybrid vehicle. The development of vehicle have tremendously changed the way of people live all over the world as vehicle became less of a luxury item and became a necessary item, to own for daily communication. While vehicle have been affordable and become part of our life, the rate of accident increase as well. A report [2] was written based on British and American crash reports in year 1995 stated that driver is the major factor of collision which is 57%.

Besides that, accident occurs might also cause by driver fatigue. The fatigue are caused by [3]:

- Travelling on long distance journeys
- Driving during midnights especially 2 a.m. to 6 a.m.
- Taken medicines that will cause side effect such as drowsiness.
- Way back to home after long shifts or night shifts

Many Automatic Driver Assistance System (ADAS) have been developed to assist in driving and to reduce the risk of vehicle accident, such as Lane Departure Warning System (LDWS), Adaptive Cruise Control System and Forward Crash Avoidance System has been introduced to reduce need of driver attention in certain road condition. Unfortunately, these safety systems only integrated into most of the luxury cars due to cost and design.

Let’s take a look at Google’s Self-driving Car [4]. This robotic car was develop by Google was equipped with Light Detection and Ranging (LIDAR), radar, camera, Global Positioning System (GPS) and wheel encoder. The core system of the vehicle, a Velodyne 64-beam laser, scanning and mapping the environment. Then it will be computed together with Google’s own map solution to produce data models that allow the car being driverless. Besides, the self-driving car also able to differentiate emergency vehicle [5], such as ambulance, from ordinary vehicles. It can behave accordingly depends on the scenario like giving way to the ambulance.

According to Google Self-Driving Car Project Monthly Report of May 2015, the self-driving cars have been travelling 1.8 million miles combining both autonomous mode and manual mode [6], but they are only involve in 14 accident which none of them are caused by the driverless vehicles.

1.1 Problem Statement

Figure 1. Various road condition and scenario.

Various road condition and scenario will lead to high failure rate in detecting the lane of the road. For instance, a road with heavy traffic congestion or road absent of lane mark will cause the system unable to detect the lane of the road. Some others scenario like weather and low light condition might interrupt the system due the ambiguous lane mark present on the road. Figure 1 shows some of the scenario that will cause the system not to function properly. The system might not able to detect the lane.

The system proposed should able to eliminate various factor using edge extraction algorithms such as Canny algorithms, steerable filters and line detection algorithms such as Hough transform algorithm to have higher success rate.

However, the ability to compute is limited since personal computer or laptop are not suitable to be used in ADAS. The system will be developed in an embedded system that use an ARMv7 processor and 1GB of RAM. The system should be optimized so that the platform able to process the task given.

LDWS are usually presented in mid to high end car but not affordable car. Besides, it was developed in different platform by different manufacturer and only exist as proprietary solution. This system was purpose so that it can be implemented on a CAN bus which most affordable car use.

1.2 Project Background and Motivation

This purpose of this project is to develop a Lane Departure Warning System (LDWS) which use a Raspberry Pi development board, a camera module design for the Raspberry Pi. The system will be able to identify lane of the road base on the road mark and alert the driver if the vehicle is departing from the lane without intention.

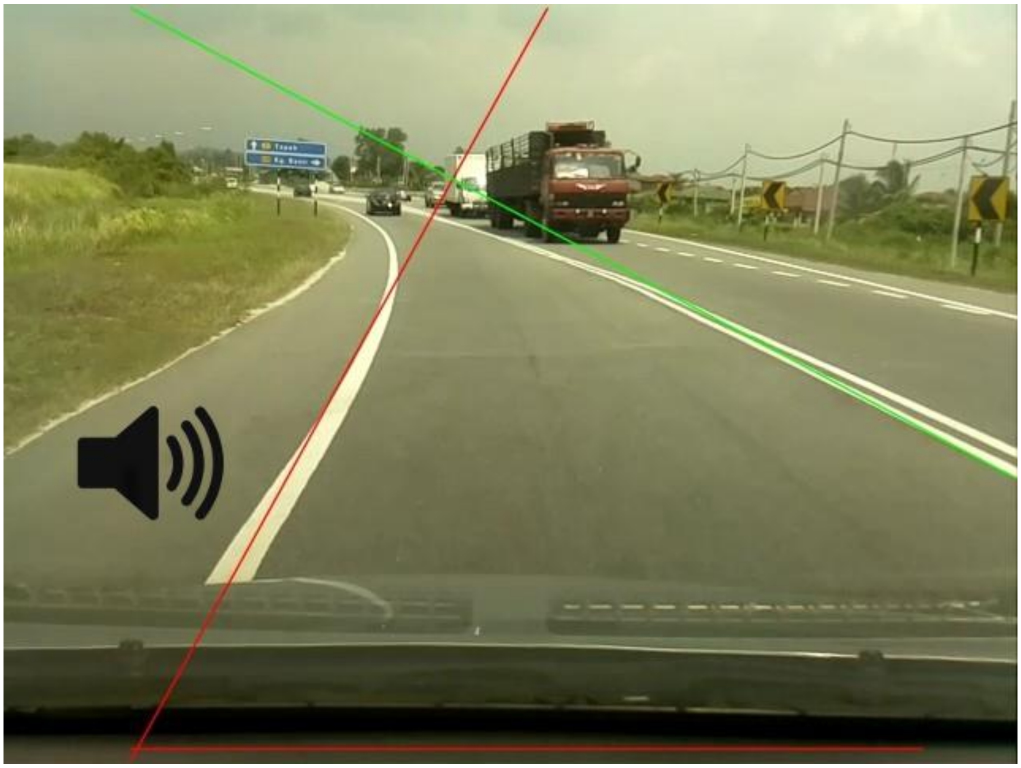

Figure 2 Simple illustration of how LDWS work.

In figure 2, the vehicle is departing from the lane without turning on the signal. Once the LDWS identify that the vehicle is out from its original lane, it will emitting alert sound to get attention from the driver.

Some of the image process algorithms and techniques will be evaluate in this project to determine which is suitable for most road condition such as Hough transform, Canny algorithm and Sobel operator.

The reason this system is develop is that, among most of the existed platform, most of them are not portable. This not only cause the system not applicable on a vehicle, it also does not provide real time response to the driver.

1.3 Objective

The main objective of this research project is to develop a prototype that could:

a. detect the lane of the road.
b. track the position of the vehicle and compare it with the lane detected.
c. emit warning sound to alert the driver if the vehicle is departing out of the lane.
d. make the above mentioned prototype portable.

The LDWS system proposed is designed to detect the lane and warn the driver if the car is diverting from its original lane. The system will be develop as an embedded system that using Linux and implements API provided in Open Source Computer Vision library (OpenCV).

In order to assist the driver to stay in lane, the system will first obtain video stream form the camera as input. Then it will detect the lane of the road. By comparing the lane and the position of the vehicle, the system should be able to determine whether the vehicle is within the lane.

The system of this project is expected able to be mounted on a sedan size vehicle because the size of the vehicle are required to calculate the region the will trigger the lane departure alert as wider and longer vehicle tend to trigger it earlier. Besides that, there are few limitation will be ignored in the prototype. The first constraint is that the lane of the road must have its own road mark. Without road mark, even if the road is detected, without knowing the numbers of lane will cause confusion to the system. Other than that, the system will assumes that the gaps between each the broken white line marks are identical. These factors can greatly affect the success rate of edge detection and Hough transform finding the road mark.

Figure 3. Sample of road taken in UTAR and Westlake.

Road in figure 3a shows that the road have road mark on the middle of the road, but no road mark around the shoulder. Road in figure 3b shows that the road have very obvious and good road marks. The system will be sampling and tested during day time and on road from Kampar to Ipoh and from Kampar to Kuala Lumpur the North-South Expressway.

1.5 Proposed Approach

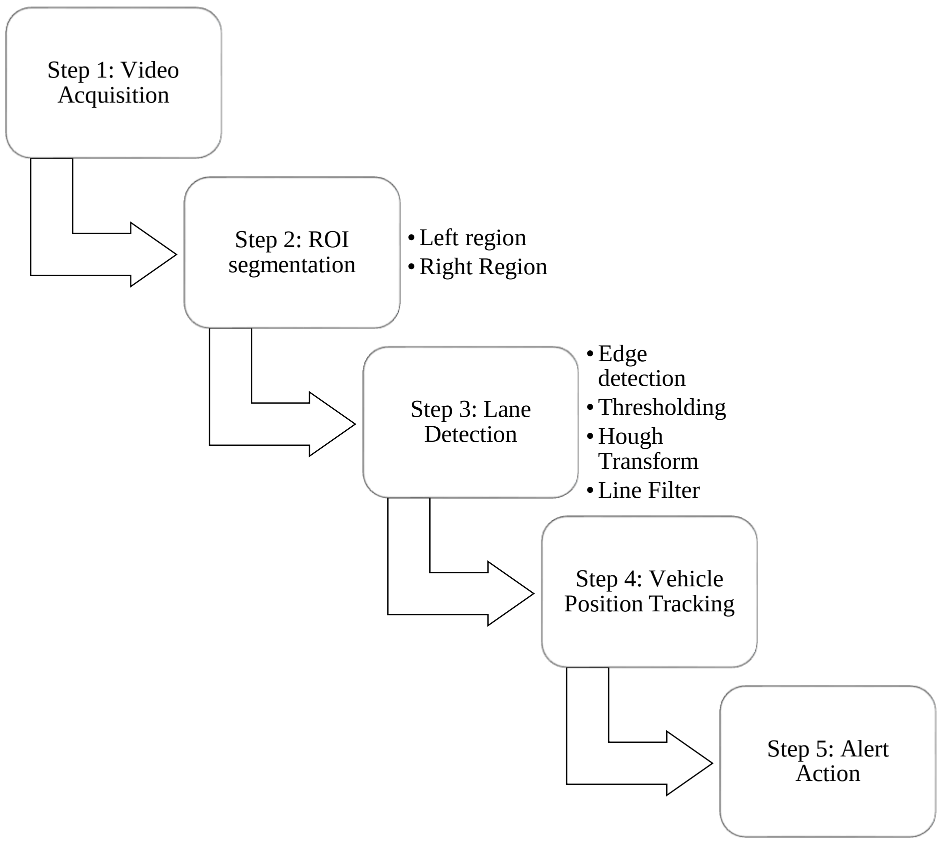

Figure 4 System module.

The first step, video acquisition in early are to store the video stream in a raw H264 container which can be processed in both Windows and Raspberry Pi platforms. The mounting of the camera will be identical to the final stage where the system is implemented.

Step 2 is used to reduce the computation requirement, noise of environment and the unwanted line detected in following steps.

The system proposed in this system will heavily depend on digital image processing of step 3. This step is crucial for the accuracy, error occurrence rate, performance in terms of frame per seconds and response time required for step 5 to alert user.

Next step will be used to determine whether if the car in on its lane or departing the lane. It then will return the value to Step 5 so that the system can alert user if the car is moving of the lane.

1.5 Report Organization

This paper consists of five chapters and is organized as following sequence. Chapter 1 will consist the introduction, the discussion on problem statement, background information and motivation, objectives, proposed approach. Chapter 2 consist of all literature reviews of the project and some of the technique that help increase the performance of the system. Chapter 3 is the design of the system of the project. There are flowcharts and various technique used in the system will be describe here. Then in Chapter 4, the details such as describing the system designs together with the methodology and tools used, requirement, specification, as well as implementation and testing of the others method will be discussed here. Last chapter which is Chapter 5 is the conclusion and overview of this project.

2. LITERATURE REVIEW

After conducting an in-depth review. I have found that LDWS is a very active title and many scholar have contributing their work.

This chapter will be divided into three subchapter. First subchapter will discuss about various embedded system that developed on Raspberry Pi. Second subchapter will discuss some image processing technique research and application of these technique using OpenCV library. The last subchapter will discuss some of the LDWS system developed and implemented on different platform.

2.1 Embedded System

In [7], the developers have proposed a real time system that enhance image regards of restoration of human skin color using Raspberry Pi. The technique used for the system is Histogram Equalization (HE) with Variable Enhancement Degree (VED). This technique was used to enhance the contrast of the image and vary the degree of enhancement to reproduce proper skin color. The system has been proposed because the existing solution apply the HE to the Hue, Saturation and Intensity (HSI) colour space. The result will mostly not desirable because if hue of the color space was altered, the skin tone will also be altered. Applying the HE directly will also cause some of the region too bright or too dark. Thus, the authors proposed that using the HE-VED technique to compute HSI colour space of the images from HSI colour space a specified image as baseline reference. This system was develop in an official Raspberry Pi OS, Raspbian Wheezy, uses Debian Linux kernel. The programming language and library used were Python and OpenCV. The execution of the system was fully automatic where the result image will be produce once an image was captured.

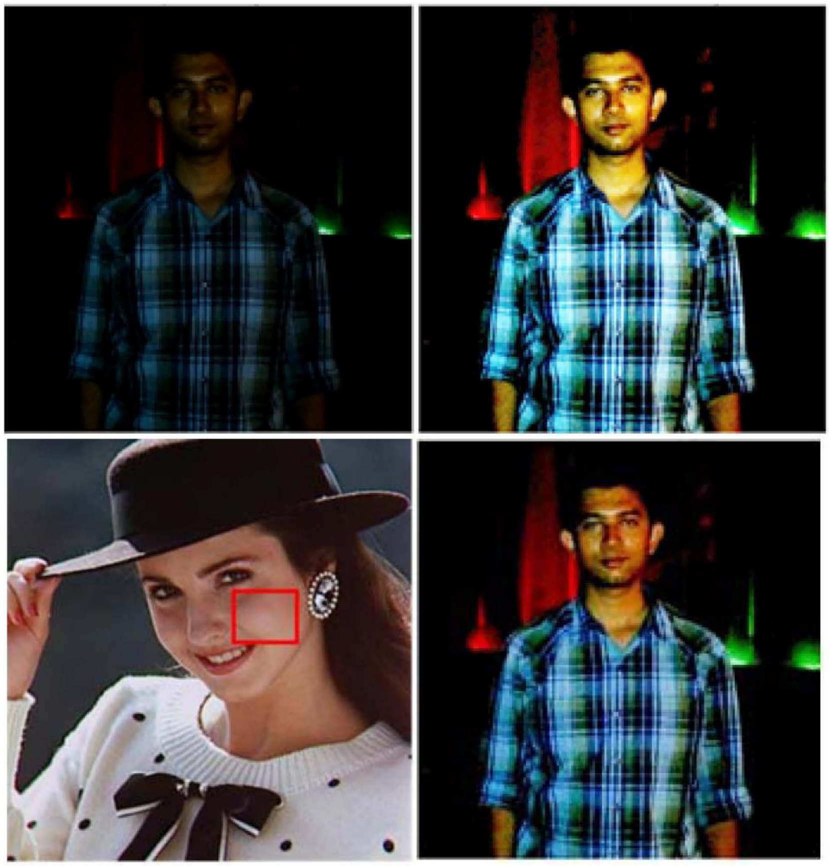

Figure 5 Result image taken from the paper.

The top left image is the original image. The top right image is enhance image with HE on saturation and intensity only. Bottom left is the reference image used. Bottom right is the final result of the system. As a result, although there were parameters needed to be varied, the system was able to present enhanced image with better skin color tone.

In paper [8], a real time vehicle monitoring and tracking system based on Raspberry Pi and Android application was developed. It was designed and developed to track school vehicles. The reason the authors proposed this system is because they found that some social problem are caused by auto theft and driver fatigue driving. Hardware used in the system were Raspberry Pi B+, GPS/GPRS/GSM SIM900A, DS18B20 temperature sensor, and MQ6 leakage sensor. The system can track the information of the vehicle such as vehicle ID, location, speed, time and data require from both sensor. The system will send all the data to server at real time which the data can be view in browser for tracking purpose. Besides, the system alert the user the driver if the driver vehicle does not travel along the designated path.

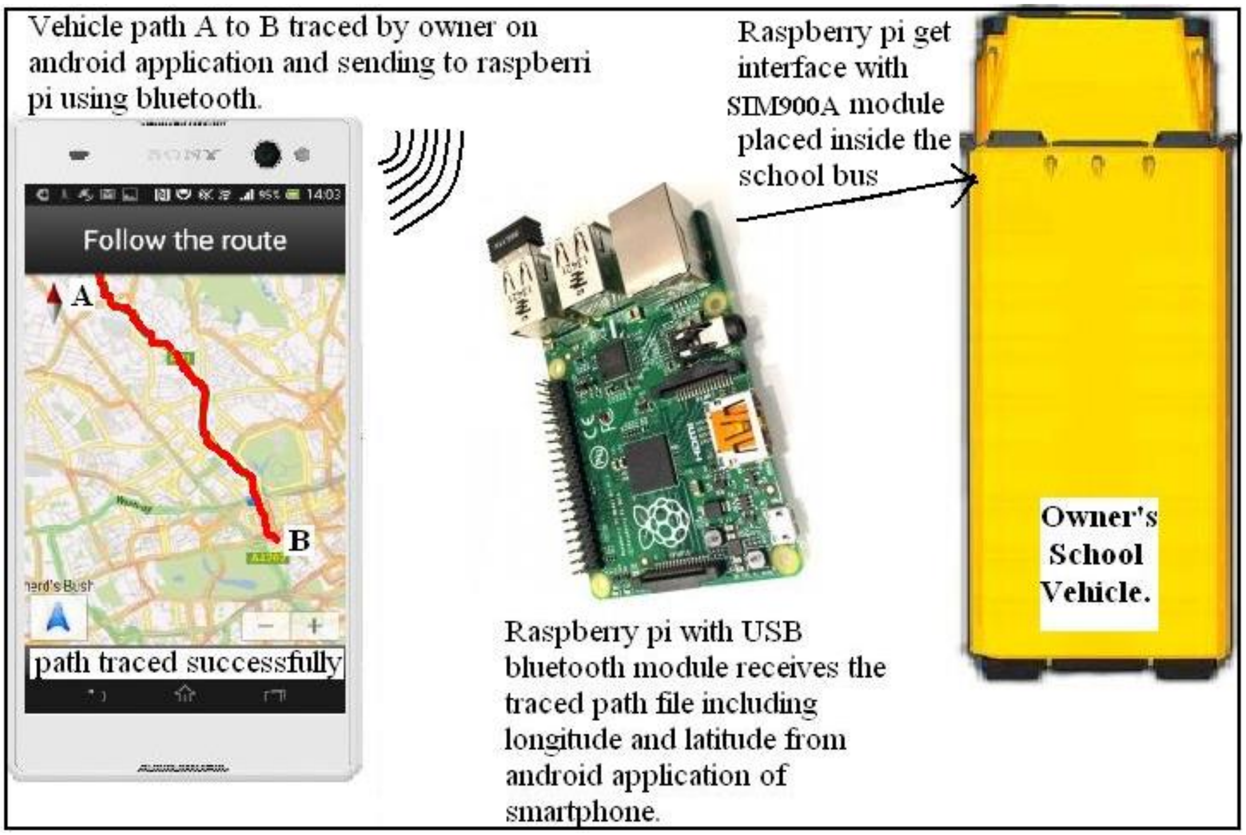

Figure 6. Illustration of how part of the system work.

As a conclusion, the system combine the technologies such as smartphone, GPRS and Raspberry Pi to provide a system that will ensure the vehicle safety and security. It keep tracks of the vehicle's location, speed and conditions.

2.2 Image Analysis and OpenCV

In [9], two types of edge detector techniques were compared. The author compared the performance of Sobel operator and Canny edge detector with their output image. The author also demonstrate the test using C++ and OpenCV library. As a result, the Sobel operator is sensitive to noise in the image. Using Gaussian smoothing before Sobel operator can reduce the noise but also affects the accuracy of the edge detection. The author conclude that Canny edge detector is immune to noise but it consume more times as it require more steps to complete and it would not be appropriate to use Canny in real time application.

In [10], various edge detection technique have been studied, compared and discussed. These techniques includes Sobel Operator, Robert's cross Operator, Prewitts's Operator, Laplacian of Gaussian and Canny Edge Detection Algorithm. This paper also used MATLAB 7.0 to develop the software. The authors of the paper had grouped most of the techniques into two categories which is gradient based edged detection and Laplacian based edge detection. In visual comparison of the output from different techniques, Canny Edge Detection Algorithm shows the best result among others. At the end of this paper, the authors conclude Sobel, Prewitt and Robert operator are simple to implements and able to detect orientations of the edges. However, these algorithms are highly sensitive to noise, rendering that edges detected in noisy environment are inaccurate. On the other hand, Canny Edge Detection Algorithm perform well in noisy environment but there are three parameter (lowest threshold, highest threshold and standard deviation for Gaussian filter) required to control. In order to detect lane, edge detection technique must be used as pre-processing steps for the video stream to reduce useless data. The result of this paper will be used as reference to find an optimal image edge detection techniques for my system.

2.3 Lane Detection and Lane Departure Warning System

In [11], a much advanced and efficient lane detection algorithm war proposed. Compared to current existing solution, the system used Region of Interest (ROI) segmentation as part of the algorithm. The author stated that there are problem in current lane detection algorithm. These problem include performance, efficiency, and various environment conditions. The authors stated that using ROI extraction can address these problems.

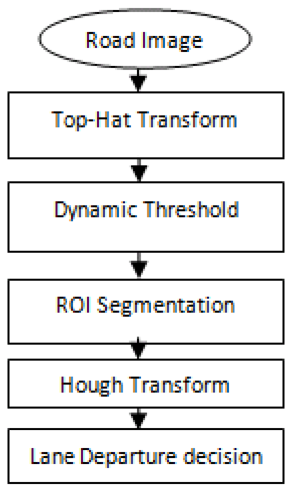

Figure 7, Flow of the proposed algorithm.

Figure 7 is the flow of the algorithm implemented in their system. The system is similar to existing solution in pre-processing the image before Hough Transform, however, ROI segmentation was implemented. This technique has two step, first, it split the image input into two major region, upper and lower. As roads will only appears in lower region, lower region will be focus and upper region will be ignored. Then, it will further split the lower region into left and right sub-regions. This can speed up the time to process because less area are required to compute and will reduce ambiguity line across the left and right region produce in Hough Transform. In conclusion, the system was performing as expected. It only took 0.053522 second to process each frame. However it stated that this algorithm still can be further enhanced. The approaches proposed in this paper will be studied and applied in my system if it is suitable

In [12], a real-time lane detection system was proposed and implemented. The paper has provide some of the key information such as algorithm and technique used in implementing this system According to the paper the system will first receive video input and then extract it into images. Then it convert the RGB image to Y'CbCr image. Thirdly it will use image filter and to detect edges. Fourthly binary image will be made by setting threshold. Hough transforms and Hough line were also used to detect the lane mark. However the system implemented did not mention any hardware specification of the system purpose. This is critical as a DAS should be compact, portable can be easily installed on a vehicle.

In [13], a real-time LDWS based on a field programmable gate array platform was purposed. Similar to our proposed project, it was developed on an embedded platform but the platform used in the paper had lower processing power and low performance for algorithms. According to this paper, the system use an enhanced vanishing point-based steerable filter to guide the orientation at each pixel. An enhanced vanishing point based Hough transform also used to reduce the hardware requirement. However, the system was not working in road with no lane marks and was too sensitive if the road were too curved.

In [14], a DAS based on Raspberry Pi was proposed. Compared to our system, both system are similar in development platform. The system was able to implement the Lane Departure Warning on a demo module based on Raspberry Pi platform. The system purposed also developed by using python and OpenCV library. According to this paper, the system received video input continuously from a USB webcam and then process the video using Hough Transform to detect the lanes. A green LED indicator will turn on if the system is in the correct lane. If the vehicle change lane, a red LED indicator will turn on and a buzzer will emit alert sound. However, this system was not tested in a real world but rather in a model road. Furthermore, the system only evaluate Hough Transform method to detect lane, this is not complete as in real world factors have to be considered.

3. SYSTEM DESIGN

3.1 System Flow

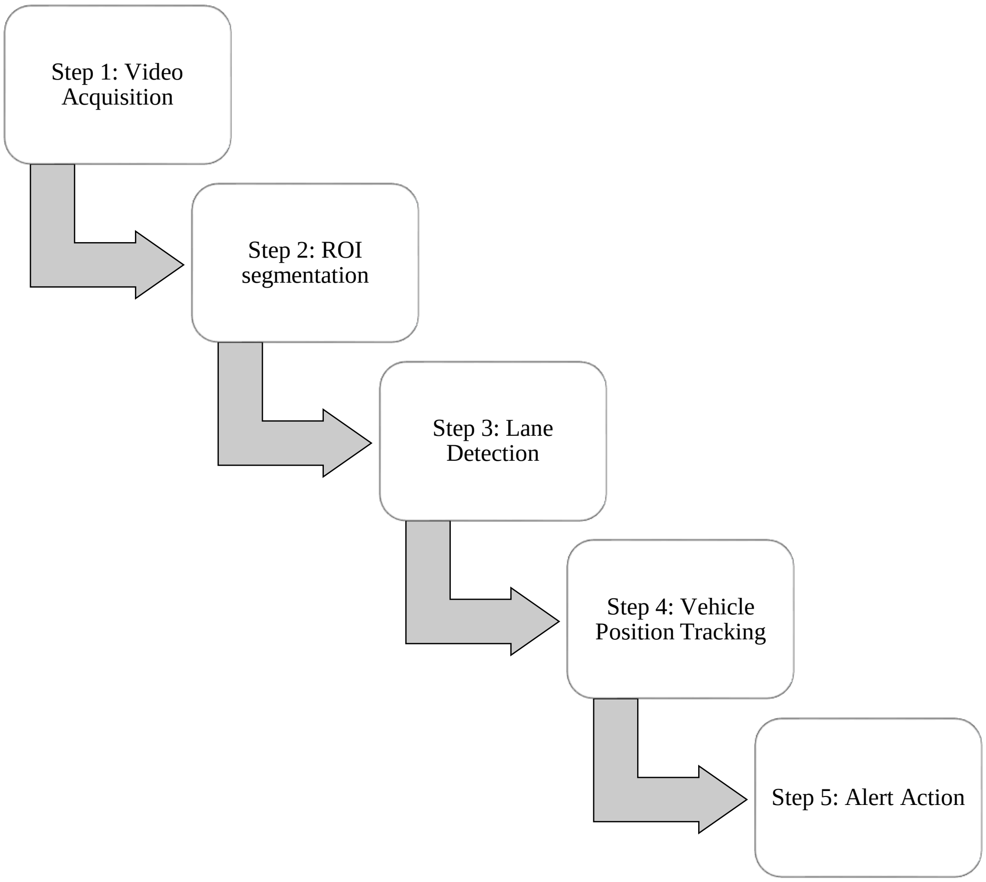

Figure 8 Methodology of the Real Time Raspberry Pi LDWS.

Real Time Raspberry Pi LDS

The following briefly describe the Real Time Raspberry Pi LDWS (RTRPL). It consists of all the steps done.

Step 1: Image Acquisition

Figure 9 Video Acquisition on North-South Expressway

Figures 9 is a screenshot of video taken on North-South Expressway. RTRPL is mounted firmly on the back of rearview mirror so that the data collected throughout the whole journey remain consistent. Each part of the video are taken and recorded for length of 30 seconds in raw H264 format.

Step 2: ROI Segmentation

ROI segmentation was done so that 2 sub-region, which is left region and right region, can be process concurrently using parallel programing technique. This is also done to reduce the size of image which is required to process.

Step 3: Lane Detection

For the lane to be detected, RTRPL is required to use digital image processing technique to extract the road lane mark. Sobel Operator is one of the edge detection technique. It is used to find the discontinuity and boundary between road and road lane mark.

Then thresholding will increased the contrast as the output of the process is binary image. Using thresholding after Sobel operator will less likely be affected by shadows and changes of lightning.

Hough Transform as a feature extraction technique, it will return lines that have be contrasted by thresholding. Then line filter is required as road lane do have specific patterns. This will eliminate most of unwanted line such as line extracted from others passing by vehicle and lamp post.

Step 4: Vehicle Positioning Tracking

This step will also be done concurrently as the system is processing the input, left region and right region. It will signal the output to the alert action module. The step has its own running threads. So that it can have maximum low responsive signal step in alert action.

Step 5: Alert Action

In this steps, 3 electronic component will be control if it gets the signals, which is 1 piezoelectric buzzer and 2 led where each of them represent 1 side of the ROI segment.

4 Description of Methodology

4.1 Methodology and Tools

At first, the project will be developed in Windows based OS. This is because a personal computer will have very sufficient resource and computation power compare to the Raspberry Pi. Visual Studio IDE and OpenCV library will be used to develop most of the functionality. Upon completion of the functionality, the whole system will be ported to the embedded system, where after this stage, optimization will be the main priority. In Raspberry Pi platform, lastest up to date version of GCC compiler will be used to compile the C++ program.

Figure 10 Software used in developing the system

The OpenCV library used in both Windows and Linux platform have been updated to 3.0.1. This is the major version up to date. There are few build-in functions provided by OpenCV such as Canny Edge Detector, Sobel Operator and Hough Lines Transform. Canny Edge Detector or Sobel Operator are edge detection pre-processing method that calculate the derivatives of an image. Comparing to Canny, Sobel are more simply in detection of edges but it would be inaccurate when noise present. However, Canny rectify this issues but it also consume computational cost. These 2 function will be discussed in the following section.

Video4Linux driver will be used as intermediate library between the C++ program and the camera module which connected to Raspberry Pi MIPI CSI interface. Video4Linux supports hardware acceleration allows the Raspberry Pi to utilize its VideoCore IV Graphic Processing Unit (GPU) for the video encoding as most of the Central Processing Unit (CPU) power will be reserved for digital image processing.

4.1 Raspberry Pi Development Board

Raspberry Pi development board is a low cost small size computer which has standard input/output such as USB port, HDMI port, and audio input/output port. It also build in with video input connector, a 15-pin MIPI camera interface connector which can connect to various different camera module design for Raspberry Pi. The general purpose input/output (GPIO) also allow the Raspberry Pi get input from sensor or signal from another.

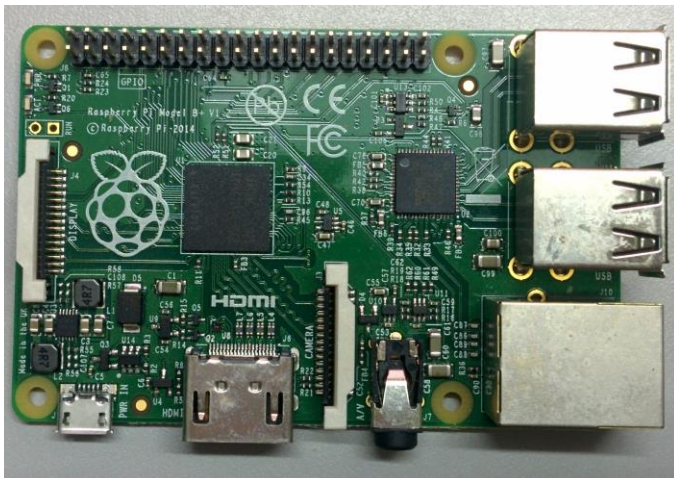

Figure 11. Layout of Raspberry Pi 2 Model B

Below are the specification of the Raspberry Pi model 2 Model B [15]:

- CPU: 900MHz quad-core ARM Cortex-A7 CPU
- GPU: Dual Core VideoCore IV® Multimedia Co-Processor
- Memory: 1GB SDRAM
- Power: Micro USB 5V, 2A
- Connectors
    - USB: 4 x USB 2.0 Connector
    - Ethernet: 10/100 BaseT Ethernet socket
    - Audio Output: 3.5mm jack, HDMI
    - Camera Connector: 15-pin MIPI Camera Serial Interface (CSI-2)
    - GPIO Connector: 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip

4.2 Raspberry Pi Camera Module

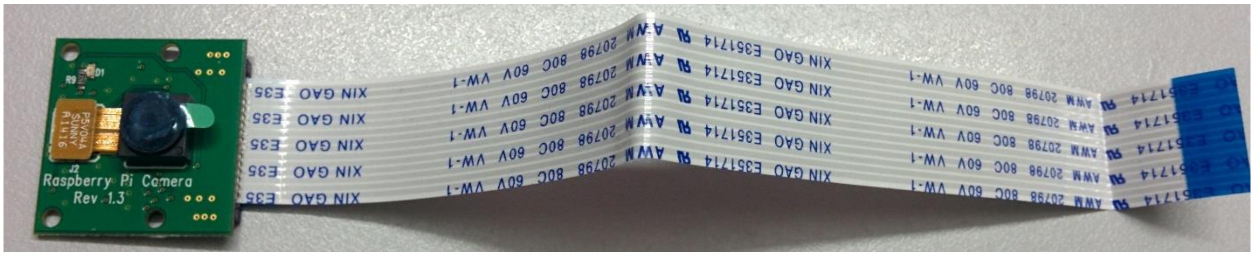

Figure 12 Raspberry Pi Camera module for Raspberry Pi

Raspberry Pi Camera Module is an image sensor module launched by Raspberry Pi foundation. The camera can be installed on the Raspberry Pi CSI connector.

The following are the specifications of the camera module [16]:

- OV5647 by OmniVision
- 5 megapixel
- Fixed focus
- 2592×1944 stills
- 1080p at 30 frames per second
- 720p at 60 frames per second
- 640x480p at 60/90 frames per second
- CSI interface with 150mm ribbon cable
- 2.4gram in weight

Figure 13 Electronics used in the system

Figure 163 shows that the installation of the system. It consists of 2 led and a piezoelectric buzzers which is keep with the casing. Then it will be mounted on a car identically to figure 14.

4.3 Real Time Raspberry Pi Lane Departure Warning System (RTRPL)

While developing the system, many image processing methods had been used to extract the lane mark of the road. Some of them will be combined so that the noise in most environment can be eliminated. Most of them will be addressed in the following steps.

Step 1: Video Acquisition

RTRPL will be mounted on back of the rearview mirror. This will allow the video recorded has a clear view on the road ahead. The suction cup mounting kit are adjustable, so that the viewing angle of the camera can be adjusted.

Figure 14 Mounting the system on the car

As the system will be implemented on a Raspberry Pi, using high resolution will cause the system to perform poorly. When the system started, it will record the video stream from the camera module which sampling at resolution of 640×480 pixels and 30 frames per second. The encoding of the video used is H264 because raspberry pi support acceleration oh H264 format using it graphic processing unit.

Figure 15 Mounting the system on the car

Figure 15 shows the screenshot in one of the video taken in North–South Expressway on the way from Kampar to KL. The image shows that the resolution size use is able to pick up the lane mark in decent quality compare even at longer distance.

Step 2: ROI Segmentation

The area capture between the hood and the fading road, which is shown in Figure 16, is the main region of interest (ROI) where it will be further processed. In this region, most of the time it will not be affected by incoming traffic.

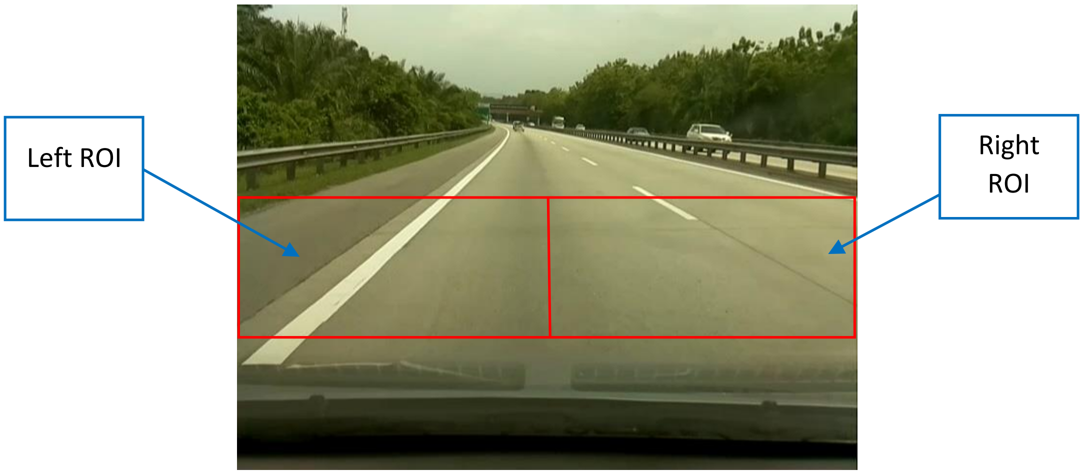

Figure 16 Region of interest

Then the region will be divided into two sub ROI which is left region and right region. This will help to reduce the noise line detected which will be discussed in following steps. Region will be set at 640 pixels wide and 100 pixels high. This is because setting the ROI with higher height will not help in the accuracy in detecting the lane as the lane mark are and will likely become noise. Besides, the higher the resolution, the more processing time the system need. This can affect the response time of the system.

Step 3: Lane Detection

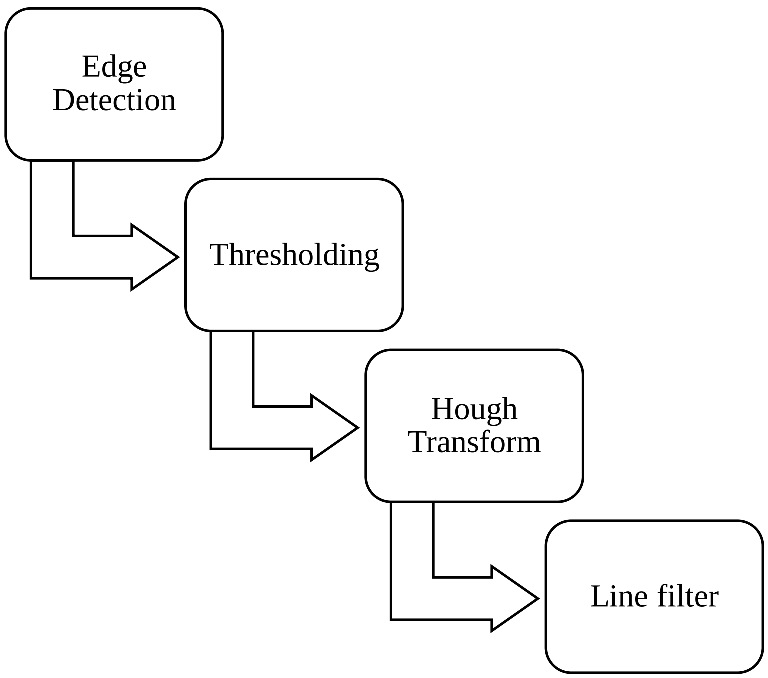

Figure 17 Lane detection process

Edge detection

In this step, there are several process required to proceed in order to obtain the correct lane mark. This will provide a filter for next step to filter out most unwanted pixels. Figure 18 shows that the original frame that is departing the lane.

Figure 18 Original input frame

The ROI will then be extrated from the frame for the first process of lane detection. The first processed is Sobel Operator. This process extract the boundary of the between the road lane and the line marks. However, since the candidate than will be extracted are only vertical line lie between 0° to 70° and 110° to 180°, only X axis of the ROI image are processed in Sobel Operator.

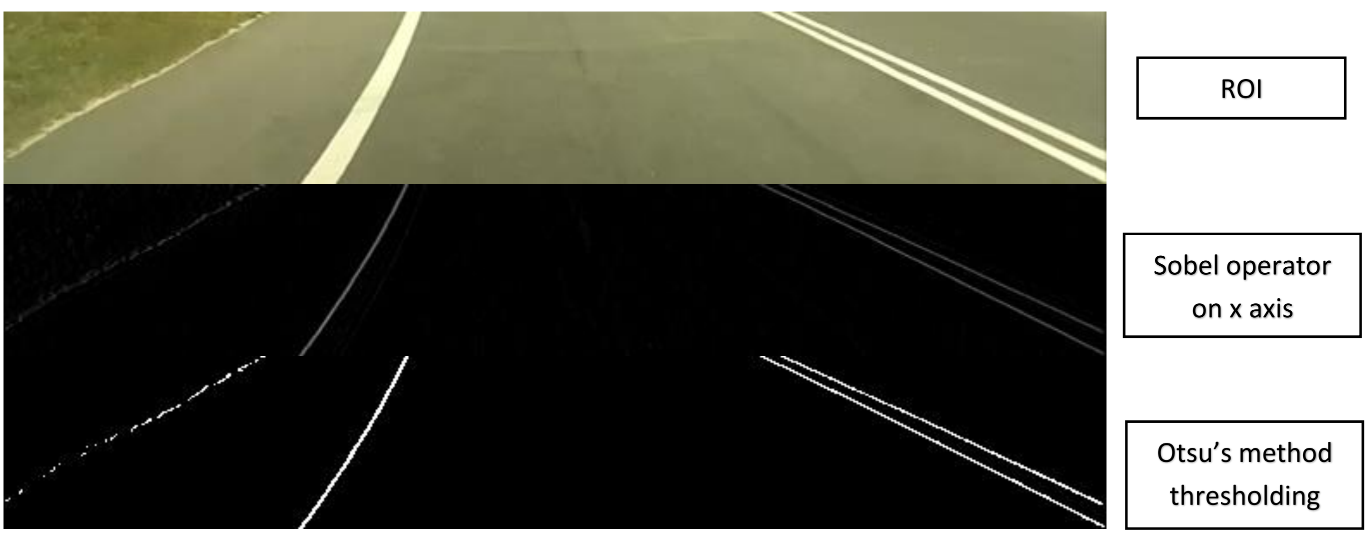

Figure 19 Output of Sobel operator and Otsu thresholding on ROI image

Image in Figure 19 show the procedure in extracting the lane mark. The last image is the output of Otsu's method thresholding. It partitioning the Sobel output image into background and foreground. The output of the Sobel is grayscale image and Otsu's method convert it into binary image.

Figure 20 Line detected in Hough Line Transform

The next process is to get the line of the boundary. The process can be done by using Hough Line Transform. The threshold value used for Hough Line Transform is 60. This is to ensure that the line detected will be more than 60 pixels. As of the height of the ROI, this can help to reduce any unwanted line detected by Hough Line Transform. Lines return by Hough Line transform will be determined by two value, which is theta and rho. Theta can be representing angle by using formula:

$$a = theta * \frac{180}{\pi}, \pi = 3.142$$

The line detected in Hough Line transform will by filter twice. First, for lines of left lane, the angle must be between 0° to 70°. For lines of right lane, the angle must be between 110° to 180°. Then among lines of left lane and lines of right lane, a single line that is nearest to the vehicle will be used. Rho is the distance between the coordinate origin (0, 0), the top left corner of the image, and the perpendicular angle of the line.

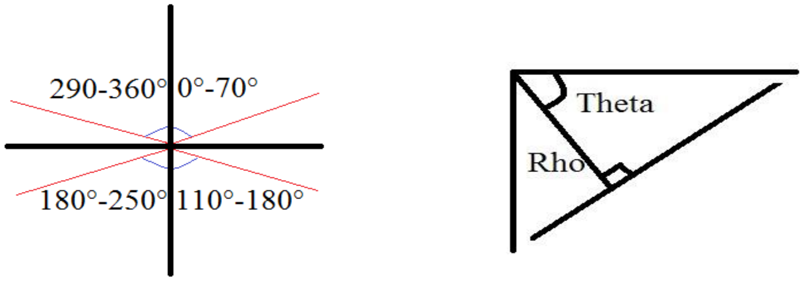

Figure 21 Illustration of the angle and distance of line from the origin used for filtering.

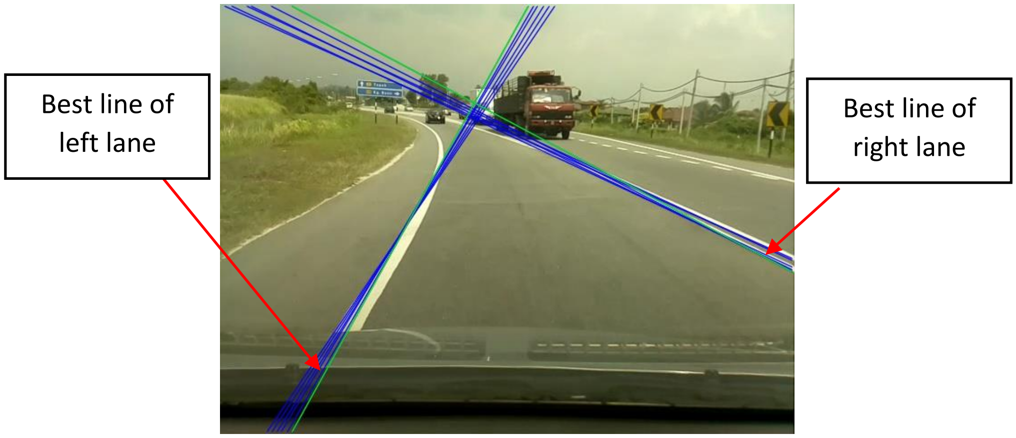

Figure 22 Result of filtering for the best line

Step 4: Vehicle Position Tracking

Once the lane is acquired, the system can now proceed to next step, to track the position of the vehicle. The length of horizontal red line in Figure 23 is calculated based on the car width, which is distance between two front tires.

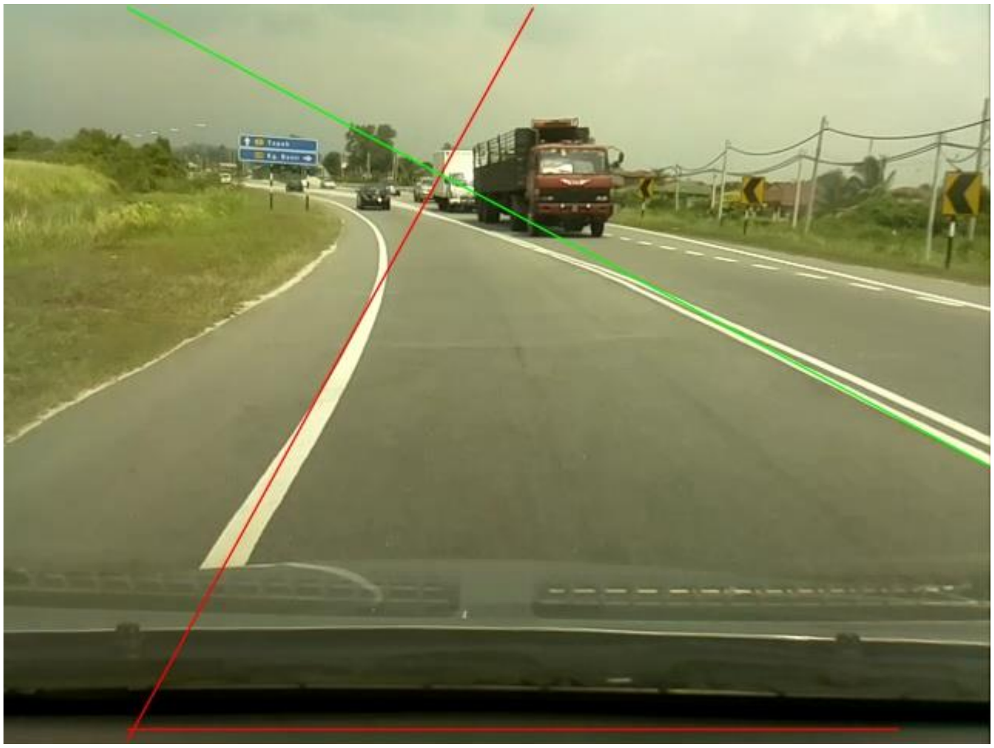

Figure 23 Vehicle position detection

This approach is easier for user to configure as they only need to know the width of the vehicle and same algorithm can be used for both left and right region. After calibration, the value will be used to detecting the interception point between the detected lane and the value calculated. If the detected lane intersect with blue red line, it will be shown as red line, instead of green line, then the driver will be alerted.

Step 5: Alert action

In this step, alert action will be performed if the vehicle is departing its lane. The buzzer and the LED is controlled via the GPIO of Raspberry Pi. There is one library to control the GPIO via in C++, which is wiringPi. WiringPi is third party library provided to support all version of Raspberry Pi to access to its GPIO with provided library. WiringPi also provide support for software and hardware Pulse Width Modulation (PWM). PWM is required to drive a piezoelectric buzzers.

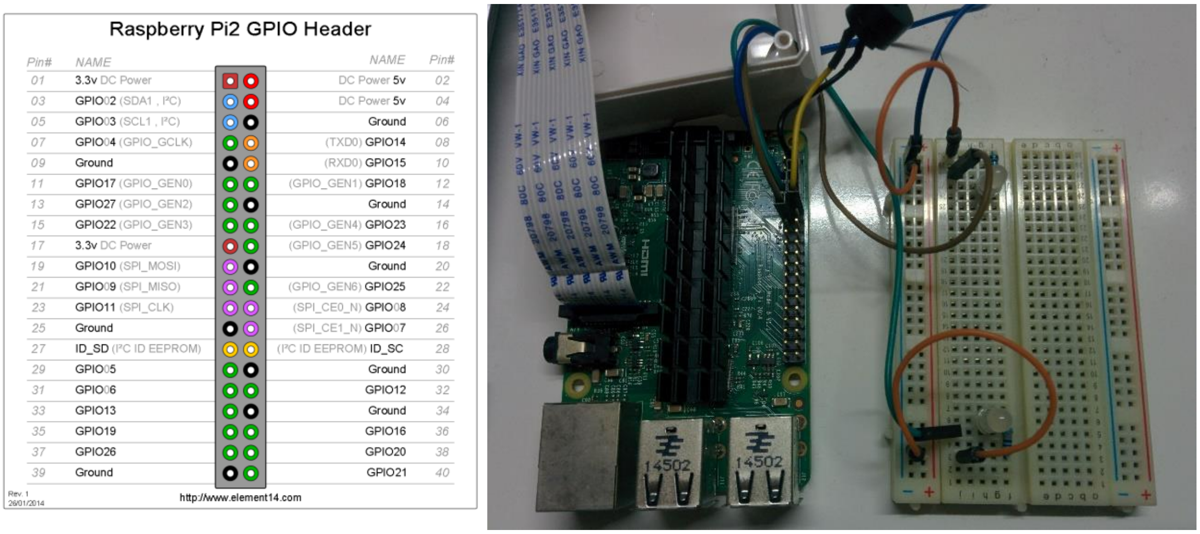

Figure 24. Raspberry Pi 2 GPIO header description and connections of electronics.

Figure 24 shows the header description of Raspberry Pi 2 Model B GPIO pin. The RTRPL will utilize only 3 pin which is GPIO17, GPIO 18 and GPIO27. Any ground pin will be able to complete the circuit of the electronics component.

4.4 RTRPL and Analysis

Beside the steps used in RTRTL, there are others methold

4.4.1 Thresholding

Figure 25 Input Image and standard thresholding output

Thresholding is easiest way to extract the lane mark from the road, however, in the observation while analyzing the result, it is found that shadow and the patches of the road lane will affect the result. This is due to the local threshold value used varied from time to time. Figure 25 shows that large area of lane mark could not be detected. The shadow cause the changing of illumination on the lane mark. As a result, basic thresholding is not suitable for first step of lane detection.

Variance of thresholding, local thresholding are able to overcome this situation. However, the algorithm used of this method, bring up many white noise. Although shadow will have minor effect on the image, where its boundary will show a visible line, the output image will be too noisy for further processing.

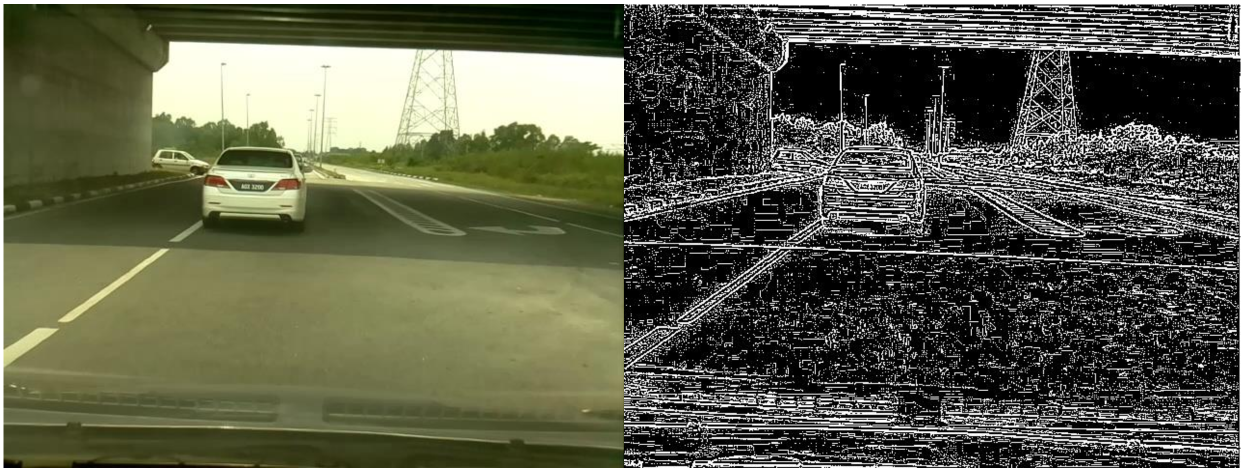

Figure 26 Input image and output of local thresholding

Figure 26 shows that the shadow of the bridge does not affect the detection of lane marks, but the image will not able be used for further processing.

4.4.2 Hue Saturation Value filter

In Malaysia, most of the road line marks are white in color and some critical road where yellow line marks will be used. A technique used to filter out all other colors is called Hue Saturation Value (HSV) filter. Figure 27 shows that how HSV model is different from Blue Green Red (BGR model).

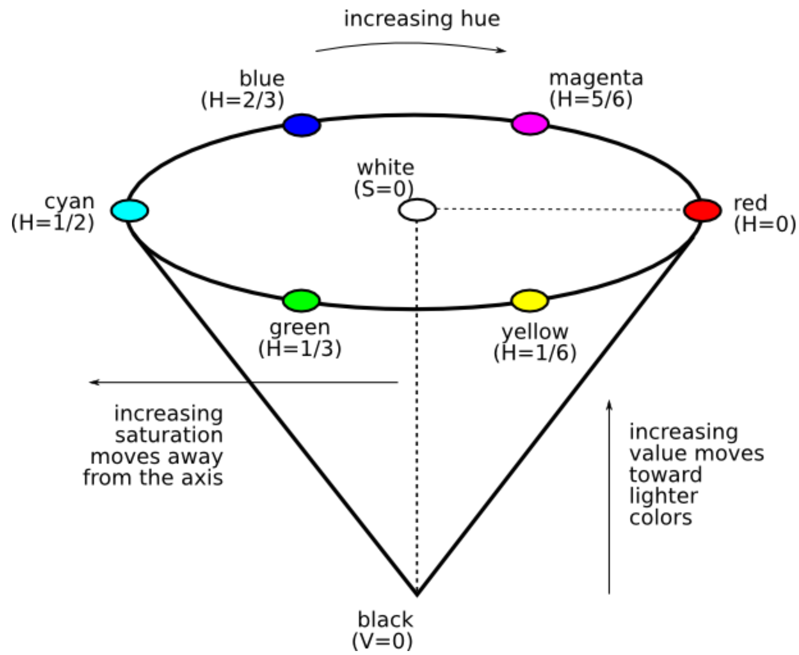

Figure 27 HSV cone.

In OpenCV, the total value of hue is only 180 degree, to filter out all other colors except yellow, the range of hue used is$180 \times \frac{1}{6} \pm 10 = 20 < h < 40$. This will help to filter all others color. Besides, value and saturation also will be filter to increase the accuracy of the system. Saturation of the lane marks is lower and value is higher. Figure 28 shows that a good filter that extracting lane. The only problem left is color of sky. In the figure, the lane mark and sky has almost identical HSV value. This can be solve by applying ROI segmentation, thus only the lane mark is detected.

Figure 28 Input image and filtered image

Figure 29 Input image and result image of imperfect filter

Figure 29 shows that all other color have been filtered out, however, as there are residue on other objects which created by reflection of lightning of sunlight, it creates a huge different in contrast. These contrasts will be pick up by the edge detection algorithm and will reduce the accuracy of lane detection. Furthermore, white color vehicle is not filter and the incoming white vehicle is likely to cause noise.

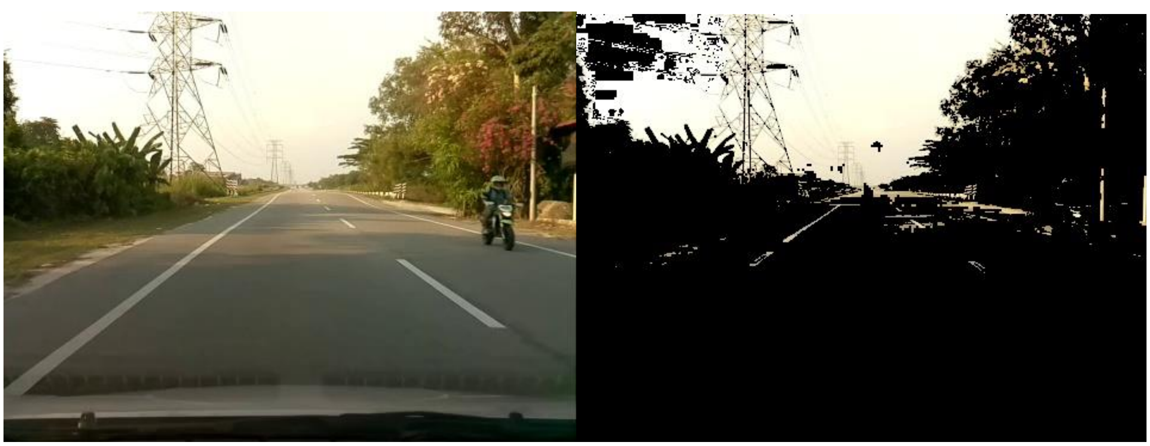

Figure 30 Input image and result image of imperfect filter

The inaccurate result of Figure 30 is more severe in this scenario. Due to the saturation and value changes, the filter is not able to detect the lane. Moreover, The residue of the created by reflection and lightning also persist.

Although this method is not usable as main process, the HSV filter can be added into RTRPL after the thresholding

5. CONCLUSIONS

When a driver is driving, he might be distracted or inattention drowsiness or mobile phone to keep his course. He might hit the lamp post or any pedestrian. In worst scenario, he might also collide heads on traffic. LDWS will alert the driver when the vehicle is departing from the lane without intention. This is useful if the vehicle's departure is due to inattention of the driver.

In this proposal, the system will be developed in Linux. Raspberry Pi is chosen as the platform. C++ and OpenCV will be used as the development language. Other than that, some image processing techniques will be discuss so that the system will be more optimized as the processing power is limited.

The 2 major issues that appear in the system are color and shadows. These 2 issues are addressed in RTRPL. However other methods mention can be combined with purposed method to enhance the system. This will compromise some of the CPU power while increase limited accuracy for the purposed system.

By learning more about image processing, the system is completed with a robust performance. From the development, a better understanding towards digital image processing, help to find out how the lane can be extracted. This project will be able to help future research to develop more advance ADAS as the project can be used as a foundation of others ADAS. In future, the Project will be focus more on the accuracy in different environment such as night.

Biblography

[1] “Don’t ask, tell” Available:http://drivingforceincentives.co.nz/dont-ask-tell/ [15 November 2014]

[2]United States Department of Transportation - Federal Highway Administration, “Interactive Highway Safety Design Model: Accident Predictive Module”, United States Department of Transportation - Federal Highway Administration, Harry Lum and Jerry A. Reagan. Public Roads Magazine, Winter 1995 Available:http://www.fhwa.dot.gov/publications/publicroads/95winter/p95wi14.cfm. [15 November 2014]

[3] The Royal Society for the Prevention of Accidents, “DRIVER FATIGUE AND ROAD ACCIDENTS” The Royal Society for the Prevention of Accidents, June 2011.
Available :http://www.rospa.com/roadsafety/info/driver_fatigue_2011.pdf [15 November 2014]

[4] E, Guizzo, “How Google's Self-Driving Car Works” IEEE Spectrum, 18 OCTOBER 2011 [20 July 2015]

[5] Google 2015 “Google Self-Driving Car Project Monthly Report May 2015”
Available:www.google.com/selfdrivingcar/files/reports/report-0515.pdf [20 July 2015]

[6] Google 2015 “Google Self-Driving Car Project Monthly Report June 2015”
Available:www.google.com/selfdrivingcar/files/reports/report-0615.pdf [20 July 2015]

[7] M. Sahani, S. K. Rout, A. K. Sharan, and S. Dutta “Real Time Color Image Enhancement with a high regard for restoration of skin color by using Raspberry Pi”, International Conference on Communication and Signal Processing, Available:http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6949857 [20 July 2015]

[8] P A. Shinde, Y.B.Mane and P. H. Tarange "Real Time Vehicle Monitoring and Tracking System based on Embedded Linux Board and Android Application", 2015 International Conference on Circuit, Power and Computing Technologies [ICCPCT] Available:http://ieeexplore.ieee.org.libezp.utar.edu.my/stamp/stamp.jsp?tp=&arnumber=7159414 [20 July 2015]

[9] D. Kim "Sobel Operator and Canny Edge Detector", ECE 480 Fall 2013 Available:http://www.egr.msu.edu/classes/ece480/capstone/fall13/group04/docs/danapp.pdf [20 July 2015]

[10] R. Maini and H. Aggarwal "Study and Comparison of Various Image Edge Detection Techniques", International Journal of Image Processing (IJIP), Volume (3) : Issue (1) Available:http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.927&rep=rep1&type=pdf [20 July 2015]

[11] S. Sharma and D. J. Shah "A Much Advanced and Efficient Lane Detection Algorithm for Intelligent Highway Safety" Available:http://www.researchgate.net/publication/257291768_A_Much_Advanced_and_Efficient_Lane_Detection_Algorithm_for_Intelligent_Highway_Safety [20 July 2015]

[12] M. Thittaporn, K. Mahasak and S. Sasipa, "REAL-TIME LANE DETECTION FOR DRIVING SYSTEM USING IMAGE PROCESSING BASED ON EDGE DETECTION AND HOUGH TRANSFORM" Available:http://sdiwc.net/digital-library/download.php?id=00000634.pdf [15 November 2014]

[13] X. An, E. Shang, J. Song, J. Li and H. He, "Real-time lane departure warning system based on a single FPGA" EURASIP Journals, vol. 38. Available:http://jivp.eurasipjournals.com/content/pdf/1687-5281-2013-38.pdf [15 November 2014]

[14] P.Narayan, "Driver Assistance System based on Raspberry Pi" International Journal of Computer Applications (0975 – 8887), vol. 95, no.16, June 2014 Available:http://research.ijcaonline.org/volume95/number16/pxc3896794.pdf [15 November 2014]

[15] "Raspberry Pi Specification Datasheets" AdaFruit Available: https://www.adafruit.com/datasheets/pi-specs.pdf [15 November 2014]

[16] Mate, "The Official Raspberry Pi Camera Module", May 2013[Online] Available:http://www.raspberrypi-spy.co.uk/2013/05/the-official-raspberry-pi-camera-module/ [15 November 2014]

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF Information and Communication Technology

Full Name(s) of Candidate(s)	Chee Winserng
ID Number(s)	1308318
Programme / Course	Communications and Networking
Title of Final Year Project	Implementing Lane Departure Warning System on Raspberry Pi

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: ______ % **Similarity by source** Internet Sources: ______ % Publications: ______ % Student Papers: ______ %	
Number of individual sources listed of more than 3% similarity: _________	
Parameters of originality required and limits approved by UTAR are as follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

____
Signature of Supervisor
Name: __________________________

Date: __________________________

____
Signature of Co-Supervisor
Name: __________________________

Date: __________________________