MOBILE ROBOT NAVIGATION SYSTEM VISION BASED THROUGH INDOOR CORRIDORS

LEE ZONG CHEN

BACHELOR OF MECHATRONICS ENGINEERING WITH HONOURS

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

MOBILE ROBOT NAVIGATION SYSTEM VISION BASED THROUGH INDOOR CORRIDORS

LEE ZONG CHEN

A report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Mechatronics Engineering with Honours

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled "MOBILE ROBOT NAVIGATION SYSTEM VISION BASED THROUGH INDOOR CORRIDORS is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ---

Name : ---

Date : ---

APPROVAL

I hereby declare that I have checked this report entitled “title of the project” and in my opinion, this thesis it complies the partial fulfillment for awarding the award of the degree of Bachelor of Mechatronics Engineering with Honours

Signature : ....

Supervisor Name : ....

Date : ....

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

In setting up this report, I was in contact with many people. They have contributed towards my understanding and thought. I wish to express my sincere appreciation to my main project supervisor, DR. Hairol Nizam Bin Mohd Shah, for encouragement, guidance, advices and motivation me. Next, I also need to thank UTeM for providing this opportunity for me and giving fund in completing this project.

My sincere appreciation also extends to all my colleagues and others who have aided at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family members for supporting me in morally and financially.

ABSTRACT

Nowadays, industry has been moving toward fourth industry revolution, but surveillance industry is still using human in patrol. This will put this industry in risk due to human nature instincts. By using a mobile robot with assist of vision sensor to patrol can bring this industry to a new level. However, the indoor corridor navigation will become a big challenge to this method. The objective of this project is to develop a navigation system using vision sensor and navigate the mobile robot in indoor corridor environment. To perform this operation, a control system though the WLAN communication develop to guide the movement of mobile robot. Besides that, corridor following system with vision sensor that using Sobel edge detection method and Hough transform to getting the vanish point is needed to help the robot to safely travel in the corridor. Both systems can be using MATLAB to be execute and link with the mobile robot through WLAN connection. This system can be analyse the corridor condition base on different feature and can decide to drive the mobile car in the direction that given. The image capture by mobile robot can be stream to MATLAB in real time and receive a feedback in short time.

ABSTRAK

Pada era globalisasi ini, kebanyakan sektor industi telah melakukan transformasi ke arah revolusi industi 4.0, tetapi industri keselamatan masih menggunakan manusia dalam rodaan. Keadaan ini akan meletakkan industri ini dalam bahaya disebabkan sifat manusia yang tamak. Dengan menggunakan robot dan dibantu oleh "vision sensor" dalam rodaan dapat meningkatkan industi ini ke aras baru. Namun, navigasi dalam bangunan merupakan cabaran dalam pelaksanaan kaedah ini. Objektif projek ini adalah untuk meghasilkan satu sistem navigasi menggunakan "vision sensor" dan menavigasi dalam koridor bangunan. Demi melaksanankan kaedah ini, satu sistem kawalan melalui penghubugan WLAN dicipta untuk mengawal pengerakkan robot. Selain itu, sistem mengikut koridor dengan bantuan "vision sensor" yang menggunakan "Sobel edge detection method" dan "Hough transform" dalam memperolehi titik lenyap adalah diperlukan dalam memastikan robot dapat bergerak dengan selamat dalam koridor. Sistem-sistem diatas dapat dilaksanakan dalam MATLAB melalui penghubungan WLAN. Sistem ini dapat menganalisikan keadaan koridor berdasarkan ciri-ciri berlainan dan membuat keputusan untuk bergerak dalam arah yang diberi. Gambar yang diambil oleh robot akan dihantar ke MATLAB dan menerima maklumbalas dalam masa singkat.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

CCTV	-	closed-circuit television
DoF	-	Degree of freedom
fps	-	frame per second
GPS	-	Global Positioning System
IMU	-	Inertia Measurement Unit
IoT	-	Internet of things
MATLAB	-	matrix laboratory
WLAN	-	wireless local area networking

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

In era of fourth industry revolution, most of the industry is transform into cyber-physical system. This system is control or monitor a mechanism of an industry through a computer-based algorithm. All the operation on that industry is digitalize, so all the output, input or problem can be analyses. From the analysis, a pattern can be obtained by the system, this pattern will be led to the main problem. By recognize and solve the main problem, it will produce a better result in term of the quantity and quality of the product.

In surveillance system, the system mostly used is closed-circuit television (CCTV) which also known as the video surveillance. In this method, the video recorded can be monitor in real time or as a reference in future in order to obtain some information from it. However, this method having some blind spot due to it required a large quantity of CCTVs to cover all place in a building.

To overcome that problem, human patrol is used to secure a place. Human is used to guard an area of place with the assist of the CCTV. Besides that, human also required to patrol the area in a random or fixed time and route frequently. This will give them permission to assess the area and it will be having some risk in it. This is due to the nature of human being that is greedy.

Hence, the human patrol task needs to be digitalized. To making this solution achieve, the advantage of the human patrol which is highly mobility must implant into the CCTV system. A robot with a camera which can move inside the path of patrol can carry out this task, but the storage of the data obtained will be an issue. This issue can be solved through the IoT solution in fourth industry revolution. This method will record the situation on the blind sport of CCTV system.

With a support of the mobile system, the human patrol can be replaced. At the same, the fixed CCTV only installed in importance location and the blind sport of the CCTV can be eliminated using the mobile system. The network of coverage of the surveillance system is expanded into larger area.

1.1 Motivation

In Malaysia, there is a massive amount of money that had stolen by the security guards who secure the building. This case is occurred on election night 2018 at Prime Minister's Office. Former Prime Minister claimed that a total of RM 3.5 billion is stolen by 17 security guards [1].

From the above incidence, there are a potential hazard in hiring a human with unknow personality to secure our estate. Therefore, it is required to replace the human patrol with machine. This machine can be a robot with vision sensor, which can analyses the area of patrol and record all the data obtained for future reference. This method not only overcome the problem above, but it also increases the quality of the patrol system. This system can provide a powerful evidence in form of image or video rather that a testimony of a security guard.

1.2 Problem Statement

The technology with vision is advance with rise of fourth industrial revolution, many industries used vision to replace human's eye in order to analyses an event. At the same time, the algorithm used for each industry is different from each other due to the field and method of analysis is different.

In this system, the most challenging part is how to make the robot tracking the corridor and moving in indoor corridor without collision. This can be real with some algorithm to process the image get by the robot in real time and return a respond to the robot in a short time. This required a high processing power to make this system effective.

Besides that, the robot also needs to be able recognize the corridors. In order to perform the navigation task, the robot required to identify the corridor condition. For example, the robot must identify the different type of junction and making the correct turning based on the navigation task.

1.3 Objective

This project has the following objectives:

1. To design and develop a navigation system using vision sensor for mobile robot.
2. To control movement of mobile robot based on the navigation system.
3. To analyze performance of mobile robot in term of accuracy of corridor tracking and recognizing.

1.4 Scope

The main scope of this project is the navigation system is design for mobile robot with camera as vision sensor. The system is use for the navigation at indoor corridors environment only. The information for the corridor condition is detect by a camera placed in front of the mobile robot, so this navigation system only can guide the mobile robot in forward direction only.

All the information of the corridor is detect using vision sensor and send to laptop as input data for algorithm. The laptop as the platform to run the algorithm. The output of the algorithm is the command that send to mobile robot as the control of movement. The communication between mobile robot and laptop is through WLAN connection. The WLAN module of the mobile robot is work as server while the laptop as its client.

Since the camera use is normal camera without the night vision, so the environment factor which is lighting condition must be fixed to visible condition. Most of the indoor corridors having limited light from the sun, so most of the corridor having the electric light. The electric light can light up the corridor to 500 and above. The navigation system only can work under this lighting condition.

CHAPTER 2

LITERATURE REVIEW

2.1 Type of Robot

Robot is an importance platform to test and stimulate an algorithm of navigation system in corridor. There are 3 type of robot used in previous work to perform this operation which are mobile robot [2] [3] [4] [5] [6] [7] [8], humanoid [9] and micro air vehicle (MAV) [10]. The mobile robot is used to perform this task due to the control of the mobile robot is easy. The combination of rotation of each motor in either clockwise (cw) or counterclockwise (ccw) will result in different motion. The Table 2.1 below will show the motion of mobile robot and its corresponding motor rotation combination.

Table 2.1: Relation of direction of motor and the motion of mobile robot

Motion of mobile robot	Rotation of left motor	Rotation of right motor
Go forward	cw	cw
Go reverse	ccw	ccw
Go left	ccw	cw
Go right	cw	ccw

However, the mobile robot only can move in flat surface. Humanoid robot can be overcome this kind of problem. Humanoid robot can move in the rough surface of corridor such as stair but hard to control it motion. The leg of the humanoid robot, NAO having 6 degree of freedom (DoF) [11]. In order to move one step, the control algorithm of humanoid robot required to calculate all angle for each of the motor. This will increase the complexity of the system. MAV is better that humanoid robot in overcome surface problem because it did not move at surface. At the same time, the factor influence it control also increase. The environment factor such as wind and the mass of sensors at MAV will affect it performance. To run a new algorithm, the robot with easy control is better because this is easy to troubleshoot the problem in the algorithm. Hence, the mobile robot is the better choice.

2.2 Type of Sensors

Sensor is the key to let a system or robot to become intelligence. The sensors to perform navigation task having many types. First type of sensor is combination of Global Positioning System (GPS) and Inertia Measurement Unit (IMU) [12] [7]. GPS is used to pinpoint location for a device, the accuracy of GPS is in range of few meter and ineffective in indoor environments. Function of IMU used to getting the direction of a device moving but it is having a cumulate error when it uses for a long time period. The combination of GPS and IMU can overcome the weakness of each other by increase the accuracy of the GPS and eliminate the cumulate error of IMU.

Next, the laser sensor such as lidar or laser scanner is another type sensor used in navigation system and obstacle avoiding system [4] [5] [6] [7]. Laser sensor is used to calculate the distance between object ahead and sensor. Laser sensor is very fast respond, easy to use and use less processing power. However, many laser sensors required to get enough information of surrounding because laser sensor only can detect the distance of a surface which is perpendicular to laser sensor. The angle of each sensors is an importance issue in order to create a 3D map.

Besides that, camera also can be use as vision sensor [2] [9] [13]. Camera is easy to setup by fixed it position on a robot. Vision sensor is can get large information about environment by real time analysis of image frame by frame. For example, in [10] was used stereo camera to build a 3D image, this can get the information in term of range data, 3D virtual scan and 3D occupancy map. This will be causing this sensor required a hardware with high processing power to analyze it, so the response time of this sensor depend on the specification of the hardware.

Last but not least, Kinect is used in [3] to perform the same operation. Kinect is a combination of vision sensor and laser sensor. From the combination of both sensors, more information can be obtained. From the library of Kinect, user can direct get the information needed with need to know or develop an algorithm. Other than that, the price of Kinect also higher than other sensors.

To getting more information to make a system to more intelligence, the more type of sensor used is better, but this also increase the mass and cost of the system. Camera is the better sensor in term of low mass and cost. Although it required a hardware to processes it data. This can be overcome by making a wireless communication between them.

2.3 Corridor Scanning Method

For a robot to move automatically in corridor environment, the method to analyze the information collected by sensors is importance. There are 2 method to analyze the data which are 3D map and image processing. For 3D map, laser sensors are required to obtain data at difference angle [3] [4] [5]. By getting the data of distance from each sensor at difference angles and this data is used to do alignment, filtering and bounding. Through this process, the distance between laser sensors at two adjacent angles is estimate, so if the number of sensors is less, this will be causing the 3D map to have many inaccuracies. Hence, large number of sensors required to create a detailed 3D map for better navigation system.

Another method is through image processing method to process the data of a camera [2] [3] [9] [13] [14]. Segmented Hough Transform and Canny's algorithm is the most common method used in the analyze corridor environment. Line segment detector (LSD) is an upgrade version of previous method by refer to methods proposed by Burn et al. method and Desolneux et al.. Both methods are using edge detection and line extraction to get the vanish point of the corridor. However, LSD is faster than pervious method in obtaining the line from an image, but LSD can't get short line from the image [8]. Therefore, it will be loss some of the data in the image. Besides that, block-based image processing method also used to obtain data about the environment [6]. This method using the pixel of the image to get the information. Grouping of the same pixel into a large pixel will increase the respond time but this method only effective in large color different such as road. Hence this method more convenience using in analyze outdoor environment.

2.4 Autonomous Navigation

Autonomous navigation is required for a robot to travel across a distance without the supervise by a human. Autonomous navigation in indoor and outdoor is difference. In outdoor environment, vehicle is control by a hardware which processing the data from the sensor. There are two way to carry this outdoor task which are combination of vision and sensors [12] [7]or vision and artificial neuron network (ANN) [13]. From the combination of vision and sensors such as GPS and IMU, the system can understand the item at surrounding of vehicle, the position of vehicle, and the direction of the vehicle move. For the vision and ANN method, the system is fully depending on the information collected by the camera only. Therefor the ANN will analyze and getting the data as much as possible to make sure the vehicle can move safely on the road.

In indoor corridors, the navigation methods are easier since the speed of mobile robot is not as fast as outdoor vehicle and did not endanger human life, so lesser safety issues need to consider. Few methods can carry out this task which are vision only or with sensor fusion, ANN and finite state machine (FSM). By using vision, the system can move I at center of the corridor by adjust it direction from the feedback of the vanish point [9]. Moreover, the vision also can help the robot to follow the path from the analysis of the image. For example, the path given is turn left at junction, then the robot will be move straight before it reach the junction until the image processing will a feedback of junction ahead and it will turn left. There is not distance measurement required in the navigation. By adding the sensors fusion, the system more correctly with error modal from the sensor feedback [2]. This will increase the accuracy of the system because compare data from two sources. Next, ANN method is used the analyze the data from image [3] or from sensor such as laser sensor [5] [6] and give the correspond command to control the robot. The few layers ANN system will analyses the data received and give the output to the motor to move from time to time until reach the destination. The last method is FSM, which is generate a route from a topological map in sequence of steps. This method is effective in navigation and did not need to know the actual position of the robot during navigation. On the other hand, this method required a topological map insert to the system and known initial position before any navigation start.

2.5 Obstacle Avoiding

In a navigation system, obstacles will appear in random along the route, so it is importance to avoid it to success travel to the destination. There are five way to avoid obstacle. First way is obtaining data from 3D map [3]. This 3D map create through the Kinect sensor is update from time to time. Thus, when an obstacle is appeared, the Kinect will update the new data into the map for the system to analyze and making the correction to planned route.

Secondly, bubble rebound algorithm apply in obstacle avoiding [6]. The robot will be rebound in the direction of low density of obstacle when an obstacle is detected at it “sensitive region”. The “sensitive region” is an area created around sensors within a range that defined by user. This method capable to avoid all type of obstacle and work in narrow corridors by using low cost sensors and microprocessor only. Even so, this method required a high level of path planning algorithm to bring the robot to destination and it motion also not smooth.

Next, another method can perform obstacle avoiding is “follow the gap method” [7]. This is using the angle of detected obstacle to calculate the gap array and find the maximum gap. Then, the gap center angle is calculated and produce the final heading angle for vehicle to move. This method is only having one turning parameter so turning process is easy, but it is limited by the speed of vehicle.

Fuzzy logic system also can use to avoid obstacles [8]. Few rules are set in the fuzzy logic system. The methods for fuzzification and reasoning are singleton and Mamdani’s method, respectively while defuzzification using center of gravity. The performance of this method is better than other in term of travel time but worse in travel distance.

Lastly, stereo vision which can create a 3D image from different angle of vision sensor is having fast obstacle mapping and path planning [10]. Not only that, it also can avoid obstacle in unknown environment. However, this method required external computer to process the data due to heavy processing power to run the algorithm and high-speed communication system for fast data transferring.

2.6 Research Gap

The type of robot chosen is mobile robot. This is due to the control of this robot is easy, so it is very suitable to testing a new algorithm. To validate a vision-based navigation system, the difficulty of the control needed to decrease to minimum so that the troubleshooting for the system can ignore the control issue.

For the sensor part, camera is used due to its ability to get many types of information and lesser weight compare to other. The information that can be extract from camera such as distance, position and direction of moving. The information gain by camera is depend on the analysis algorithm, which are not same as other sensors that only sense particular information only and making the system need to add on new sensor to get new information. In addition, the camera also can record all the route travelled for a record or future use.

To analyze image from the camera, the more information extracted is better, so Segmented Hough Transformation and Canny's algorithm are selected. Although these two algorithms are consuming some time to process the image, but it can get on more detail line from the image so increase the accuracy of the vanish point. To reduce the time taken, the analysis is limit to some area of the image which are the corridor lines lined on.

Since the camera is chosen as sensor, the autonomous navigation and obstacle avoiding are using vision method. In this method, some algorithms are used to obtaining the required data to perform these operations. Since the indoor moving speed and dangerous level is not as high as outdoor, so without help of other sensors also can move safety and follow the path planned. The Figure 2.1 shown the summary of the literature review.

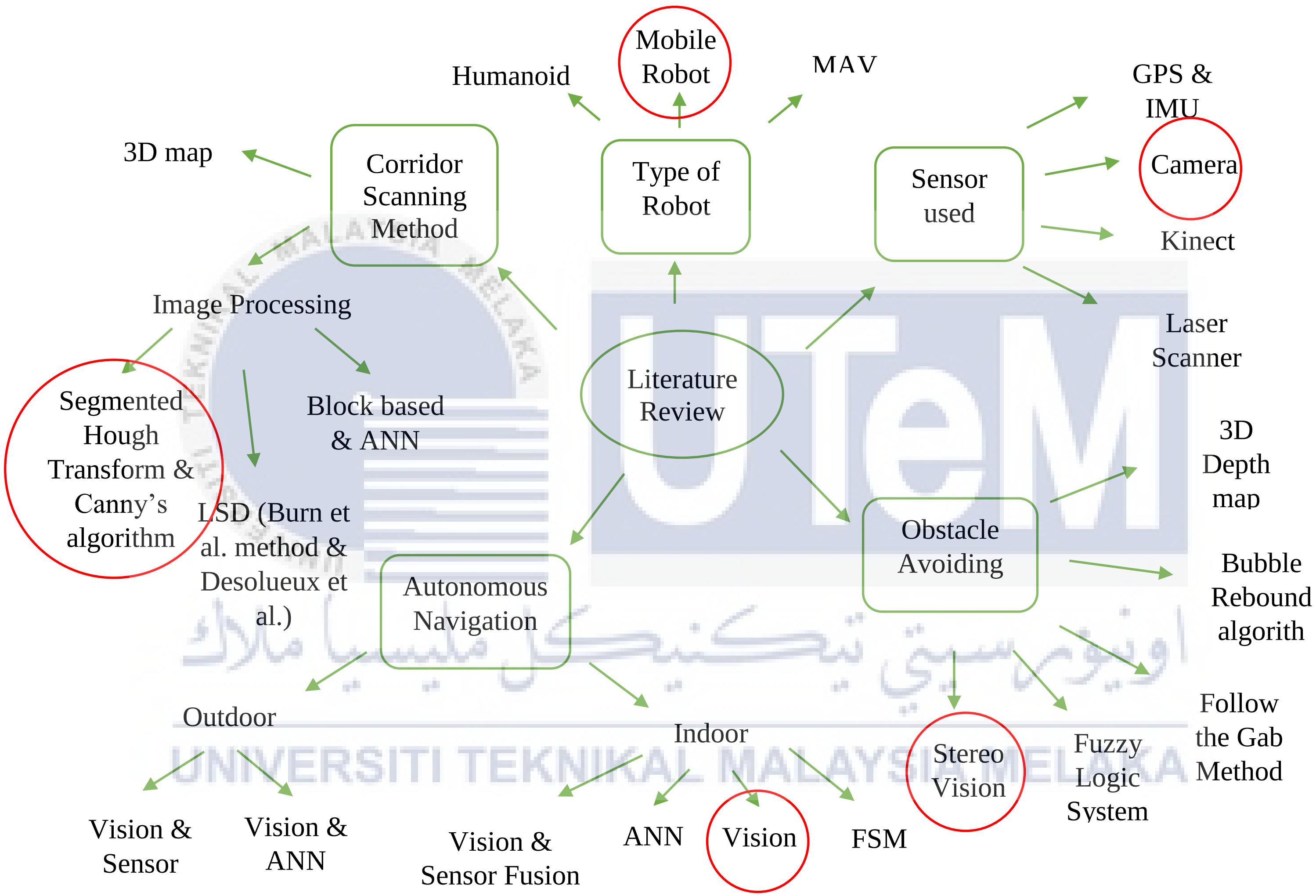


Figure 2.1: Summary of literature review

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the organization and methodology of the project is discussed. It describes and explains all the method, technique and procedure in order to achieve the objective of this research. A project flowchart is an overall process to complete this research. Project flowchart shown all the step include planning, develop software, and analysis. The project flowchart is show in Figure 3.1.

Before starting to do the project, the title of the project must be understanding first. This is to make sure actual requirement of the project is not deviated. Few researches on the concept and technique are carry out to understand and having a concept about what need to be done. After having the direction of the project, a literature review on past researches is needed to understand the technique used and the pros and cons of the technique. A suitable technique is selected based on research that carry out. After confirming the technique used, suitable hardware is chosen and a control system for the hardware is developed. On the other hand, an algorithm of vision is developed. The control of the system is combined with the vision algorithm to complete the project. This system is undergone adjustment to improve its performance. The performance of the system is analyzed.

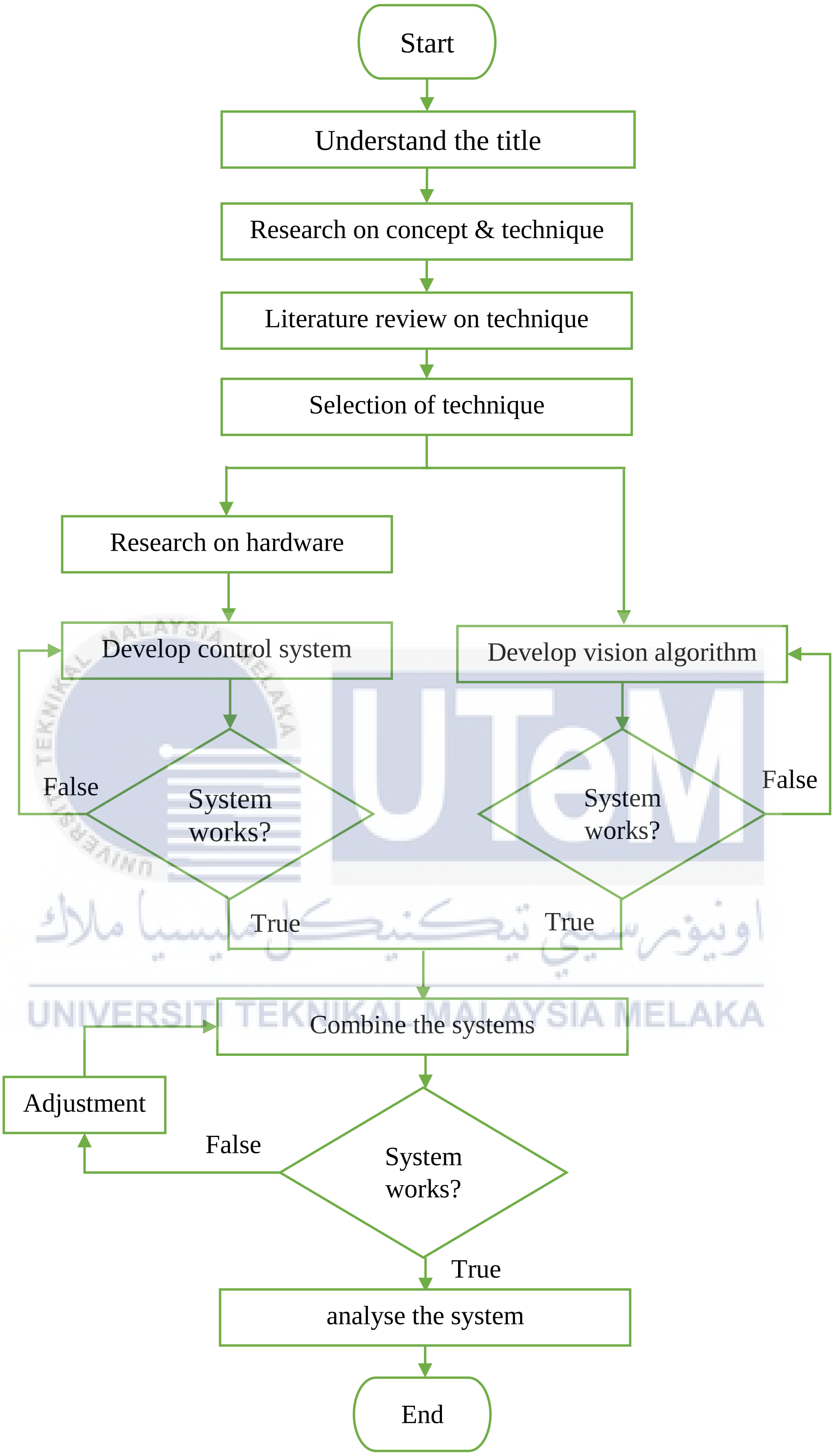


Figure 3.1: Project flowchart

3.2 Methodolygy Flowchart

In methodology flowchart, it will show the overall method use to complete this research. This flowchart will discuss the method use for hardware and software developing. The Figure 3.2 show the flowchart of the navigation system that developed in this project.

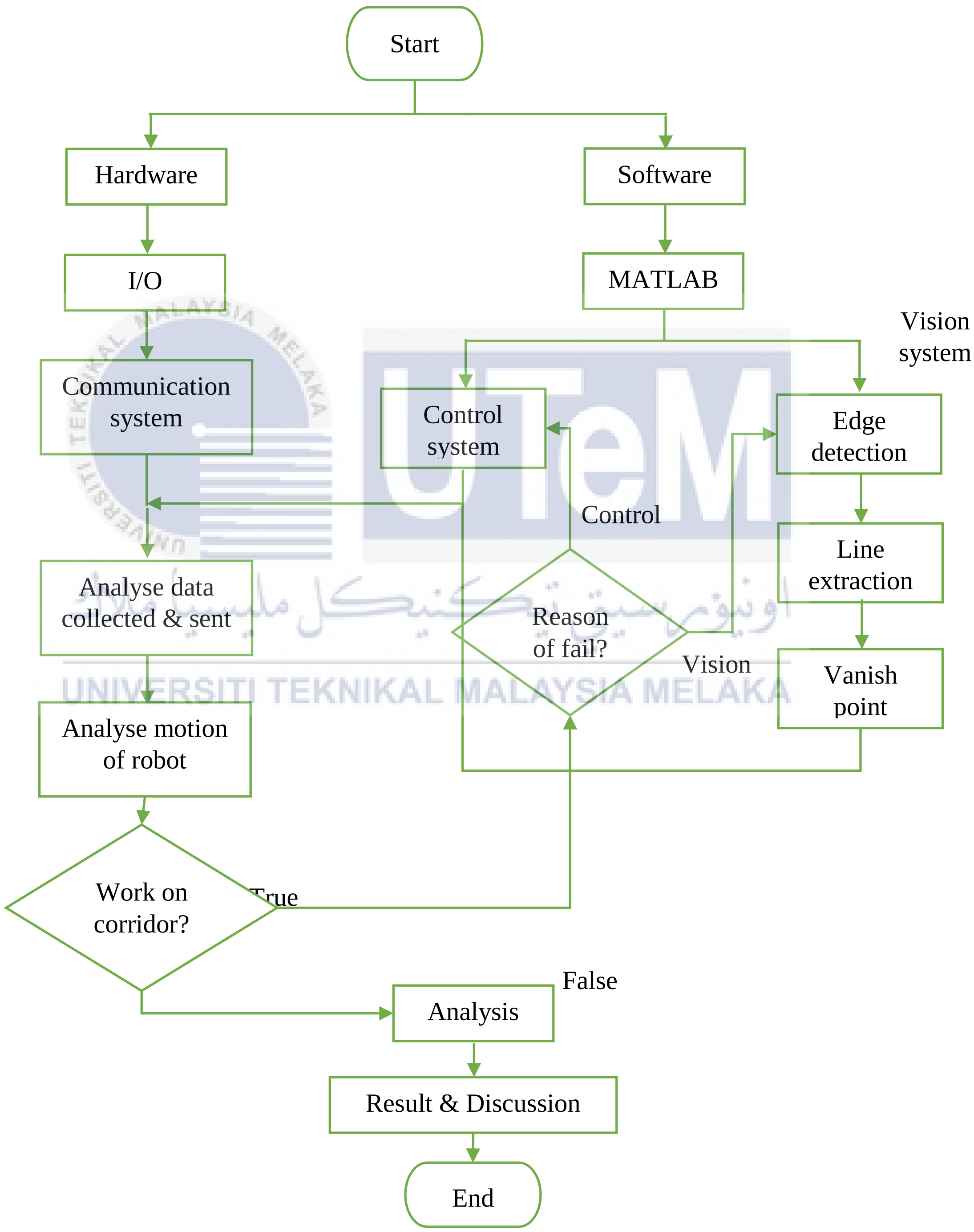


Figure 3.2: Flowchart of navigation system

The Figure 3.2 show the development of hardware and software. For hardware part, the input output (I/O) configuration is required to developing communication system. In the software part, there are two systems to be developed which are vision and control system. The vision system uses to collect the information and guide the system through it feedback. On the other hand, the control system needs the feedback of vision system to interface with communication system to navigate the system.

3.2.1 Edge detection

This vision system is collecting the information by capture the image in corridor and process it into useful data. Edge detection is the importance process to convert RGB image capture by the robot to an image of the edge in image. Edge detection is an image processing method used to find the boundaries of object in an image. This method works by detecting the discontinuities of the brightness within the image. It is used to extract the data in the image as an edge image. This edge image is a binary image. There a many type of edge detection methods. The different type of edge detection method is used to obtain the edge of the same image and the result of the different method is compared to choosing the suitable method for the vision system.

3.2.2 Line extraction

The Hough transformation is covert the edge image into data form. The data obtained from the calculation is the first point and end point of a line, the shortest distance of the line with the origin point, rho and the angle of the rho from the origin. From the first point and end point of the line, the gradients of the lines are obtained. These gradients are used to form linear line equations. These line equations are extended the line to the border of the image to making the intersect of line easier. The angle obtained from the Hough transformation is the importance characteristic in this system. These angles are used to classify the lines into different sections. There are 4 sections, which are vertical, horizontal, left and right section. The vertical section is from -30° to 30° while the horizontal is from 70° to 90° and -70° to -90°. The vertical line will be ignored in system while the horizontal line is used to determine the stop condition. The left and right section angle is the same but different in the sign only and these lines is the importance line which are used in vanish point calculation.

3.2.3 Vanish point

From the lines plot from the data, the line in positive angle (left section) is intersect with the negative angle (right section). Each of the lines is intersect will all the line individually. Two lines are used to calculated the intersect point at one time. The intersection point is obtain from the gradients and y-intersects of the line equations of both of the lines. The intersect points are recorded. The vanish point is calculate using the median of the intersect points. The horizontal distance between vanish point and the center of image is the feedback of vision system. The horizontal distance is used due to the vanish point is not at a constant y position (vertical line) and the robot only can adjust its position in the vertical direction. Hence the vertical distance can be ignored in the system.

3.3 Hardware description

The mobile robot use in this research is Arduino DS Robot [15] as shown in Figure 3.3. This robot is included a camera, WLAN module, Arduino Uno board, power supply and motor driver board and 4 motors.

Figure 3.3: Arduino DS Robot

The camera is support by a camera cradle which consists of two servo motor. These servo motor can turn the camera in four direction, which are left, right, up and down. However, in this project, the orientation of the camera is fixed at the center position only, without turning.

WLAN module is using to perform communication between mobile robot and laptop. This module support 802.11b/g/n at maximum 150Mbps with frequency of 2.4 GHz. The mobile robot is act as a server that sent image data to client, laptop and receive command of movement from laptop.

Arduino Uno board is act as the processor of the mobile robot. Arduino is preprogrammed with a coding to control the mobile robot movement. The command that receive by the WLAN module is process by the Arduino and send the signal to the motor drive.

Power supply and motor drive board distribute and regulate the power to the mobile robot. This board is supply with a Li-ion battery with 7.4V 1200mAh. When this board receive signal from Arduino, it will give power to the motor to rotate either in clockwise or counterclockwise.

The motor is the actuator for the mobile robot to move. The four motor is connected to two port of the motor driver, which mean that two motor is connected to one port. These two motors are on the same side so will be move in the same direction. Direction of rotation of the motor on each side will determine the movement of the mobile robot.

Since the Arduino is preprogrammed, so signal sent by the laptop for communication is importance in order to get the correct movement. The Table 3.1 shown the direction of movement and corresponding signal.

Table 3.1: The direction of movement and corresponding signal

	Head	Type	Cmd	data	Tail
Stop	FF	00	00	00	FF
Forward	FF	00	01	00	FF
Backward	FF	00	02	00	FF
Turn Left	FF	00	03	00	FF
Turn Right	FF	00	04	00	FF

From the Table 3.1, to tell the robot to move forward, signal that need to send is FF000100FF.

3.4 Research Design

In this research, there were 3 different experiments that carried out to achieve the objective. The experiments involve determine the better edge detection method , develop the vanish point algorithm and testing it performance in different environment of corridor.

3.4.1 Experiment 1 : Comparison between edge detection method

Edge detection was used to process the image to obtain the lines on every edge in the image. These lines obtained can be used to getting the information of the corridor, so suitable edge detection method is importance to better performance.

Few photos were captured in different corridor through the mobile robot camera to use in testing the edge detection methods. There are 2 methods which are Canny edge detection method and Sobel edge detection method. A set of photos were processed using these two edge detection methods, and the results is observed. From the observation, a suitable method was chosen to find the vanish point.

3.4.1.1 Parameters

The parameters used in equipment of this experiment are shown in the Table 3.2. These parameters are included all the version and specification used.

Table 3.2: Parameter of experiment 1

Item		Parameters
Version of MATLAB		MATLAB 9.6 (R2018b)
Laptop		MSI GP62 2QD Leopard
Specification	CPU	Intel(R) Core i7-5700HQ @ 2.70GHz
	GPU	Intel(R) HD Graphics 5600
	RAM	8.0 GB DDR3
Mobile robot		Arduino DS Robot
	WLAN	802.11b/g/n @ 2.4 GHz
	Camera resolution	640 x 480

3.4.1.2 Equipment

The equipment required to carry out this experiment are:

- MATLAB R2018b
- MSI GP62 2QD Leopard
- Arduino DS Robot

3.4.2 Experiment 2 : Development of vanish point algorithm

After the image was processed using Sobel method, the image is converted to a binary image with only the edge information only. This image was undergone Hough Transformation to transform the information into data that can be used. Hough Transformation formula is shown in equation 3.1 below.

$$r = x cos\theta + y sin\theta \tag{3.1}$$

Where r is the distance from the origin to the closest point on the straight line, and theta θ is the angle between the x axis and the line connecting the origin with that closest point.

From the data calculated through Hough Transformation, the lines which fulfil the Hough parameter will be stored as it first and last point as a variable. The points were used to obtain it line equation in equation 3.2.

$$y = mx + c \tag{3.2}$$

Where x and y are the coordinate of the point, m is the gradient of the line and c is the y-intersect of the line.

With the line equation, the lines were extended from one end to another end of image. This will allow the intersection of line to occur. The point of intersect of the lines is calculate by using these formulas in equation 3.3 and 3.4.

$$xintersect = \frac{c_2 - c_1}{m_1 - m_2} \tag{3.3}$$

$$yintersect = m_1 \times xintercept + c_1 \tag{3.4}$$

Where xintersect and yintersect in equation 3.3 and 3.4 are the coordinate for the intersection, m_1 is the gradient and c_1 is the y-intersect of line 1 while m_2 is the gradient and c_2 is the y-intersect of line 2.

When all intersection of points was calculated, the median of all the points is obtained as the vanish point. The mobile robot will be react based on the distance of the vanish point and the center of camera in x-axis.

All the lines and points were plotted in the image with different color for easy to recognize.

3.4.2.1 Parameters

The parameters used in equipment of this experiment are shown in the Table 3.3. These parameters are included all the version and specification used.

Table 3.3: Parameter of experiment 2

Item		Parameters
Version of MATLAB		MATLAB 9.6 (R2018b)
Laptop		MSI GP62 2QD Leopard
Specification	CPU	Intel(R) Core i7-5700HQ @ 2.70GHz
	GPU	Intel(R) HD Graphics 5600
	RAM	8.0 GB DDR3
Mobile robot		Arduino DS Robot
	WLAN	802.11b/g/n @ 2.4 GHz
	Camera resolution	640 x 480
Hough	Peak number	200
	Peak threshold	0.2 x maximum of Hough value
	Minimum line gab size	10 pixels
	Minimum line length	100 pixels

3.4.2.2 Equipment

The equipment required to carry out this experiment are:

- MATLAB R2018b
- MSI GP62 2QD Leopard
- Arduino DS Robot

3.4.3 Experiment 3 : Test run in corridor with different environment.

To analyze the performance of the system, 3 corridors were chosen based on its environment. The first corridor is one of the corridors in Sport Complex of UTeM as shown in Figure 3.4. This corridor was chosen due to it was clear of obstacle and the end of the corridor is a wall without anything at the end. Hence, the vanish point calculated is not influence by another factor.

Figure 3.4: Image of first test run location

The second corridor that test run occurred was in front lift in third floor of FKE as shown in Figure 3.5. The location chosen because of it don’t have any obstacle in its runway but the end of corridor is not flat. This will affect it calculation of vanish point.

Figure 3.5: Image of second test run location

The last location was in front of FKE lecture's office at first floor as shown in Figure 3.6. This location has a flat end, but it has many obstacles at side of the corridor. These obstacles will be count into the calculation of the vanish point and will affect the decision made by the mobile robot.

Figure 3.6: Image of third test run location

The mobile robot was run in with the navigation system developed in every corridor above. In each corridor, the mobile robot is run 15 times and the result of each run will be recorded. The accuracy of the system in each corridor was calculated using formula of equation 3.5 below.

$$accuracy = \frac{number\ of\ success\ runs\ or\ image\ detected}{total\ number\ of\ runs\ or\ image\ tested} \quad (3.5)$$

During each run, the image captured along the path was stored in laptop. The vanish point of the image were obtained by the same system and all related line and point were shown in the image. These processed images were observed, and the type of errors occurred was recorded in a table. The accuracy of the system is calculated based on the table using formula 3.5.

3.4.3.1 Parameters

The parameters used in equipment of this experiment are shown in the Table 3.4. These parameters are included all the version and specification used.

Table 3.4: Parameter of experiment 3

Item		Parameters
Version of MATLAB		MATLAB 9.6 (R2018b)
Laptop		MSI GP62 2QD Leopard
Specification	CPU	Intel(R) Core i7-5700HQ @ 2.70GHz
	GPU	Intel(R) HD Graphics 5600
	RAM	8.0 GB DDR3
Mobile robot		Arduino DS Robot
	WLAN	802.11b/g/n @ 2.4 GHz
	Camera esolution	640 x 480
Hough	Peak number	200
	Peak threshold	0.2 x maximum of Hough value
	Minimum line gab size	10 pixels
	Minimum line length	100 pixels

3.4.3.2 Equipment

The equipment required to carry out this experiment are:

- MATLAB R2018b
- MSI GP62 2QD Leopard
- Arduino DS Robot

CHAPTER 4

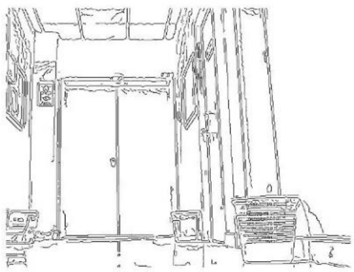
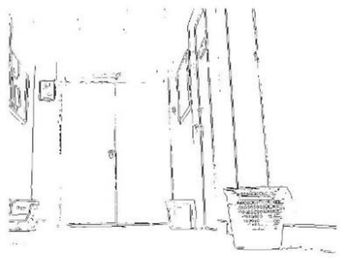
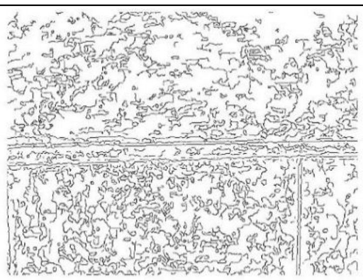
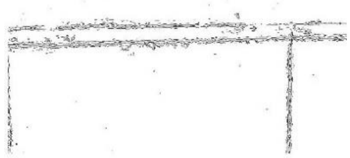
RESULTS AND DISCUSSIONS

4.1 Comparison between edge detection method

To identify the different of Canny and Sobel edge detection, two photos is taken in different corridor. these photos are captured using the camera of robot. The Figure 4.1shown the image used to compare the result of these edge detection method.

Figure 4.1: Image of corridor (left) and wall (left)

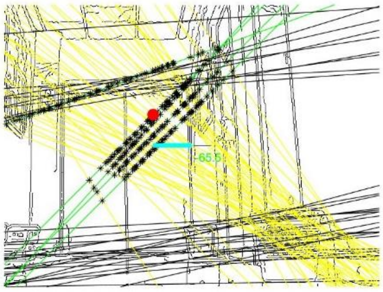
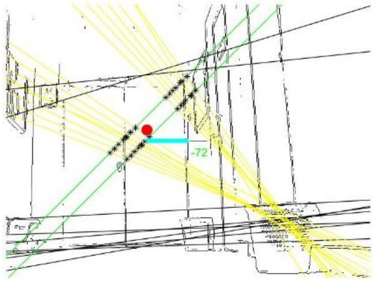
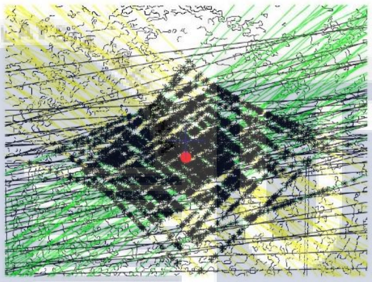
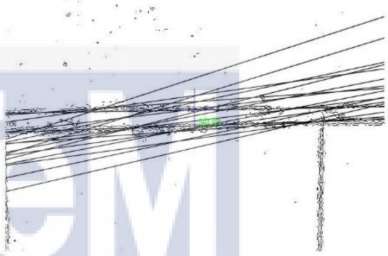
Table 4.1: Comparison of Canny and Sobel edge detection method

	Canny	Sobel
Corridor		
Wall		

From the Table 4.1, we can observe that the corridor is restore almost all the detail using the Canny edge detection method, but edge detected from the wall is too much noise. This noise will affect the calculation for the vanish point and lead the mobile robot to a wrong direction.

The edge image in the Table 4.1 are used in the vanish point algorithm and the result shown in Table 4.2 with all the importance data is plotted.

Table 4.2: Comparison of both method after processed

	Canny	Sobel
Corridor		
Wall		

From the Table 4.2, for image of corridor, Canny edge detection method is obtaining more line than Sobel edge detection method. The higher the number of lines, the more accurate the vanish point calculated. However, in the image of the wall, the noise detected by the Canny edge detection method is obtained the unwanted lines, this will lead the mobile robot to make wrong decision. Through this experiment, Sobel edge detection method is chosen to use in our system.

4.2 Development of vanish point algorithm

A photo of corridor as shown in Figure 4.2 is selected to processed using the vanish point algorithm. This photo is chosen due to the line of corridor is obvious and no obstacle in corridor.

Figure 4.2: Sample image for vanish point algorithm

This photo will undergo the edge detection and converted to binary image shown in Figure 4.3. This photo is shown the edge contain in the sample image.

Figure 4.3: Edge of sample image

From the edge image, Hough Transformation was converting all the edge into a data that can used for the algorithm to calculate. All data of the edge is recorded in the APPENDIX D.

In Table 4.3, the rho and theta are representing the r and θ in equation 3.1. While the point 1 and point 2 were used in the equation 3.2 to obtain the line equation and extended to the frame of image as shown in Figure 4.4. The different color of lines used in the figure 4.4 are used to differentiate the lines according to it θ value. Yellow lines are representing lines with θ values from -30 to -70, which also same angle with the line separate the floor and the right wall. The green lines are same as the yellow lines but is in positive angle, so it is shown the angle of the line separate the floor and the left wall. The blue lines are representing the horizontal line which is the angle between 70 to 90. The vertical line was not used in the algorithm, so it was not plotted in the image.

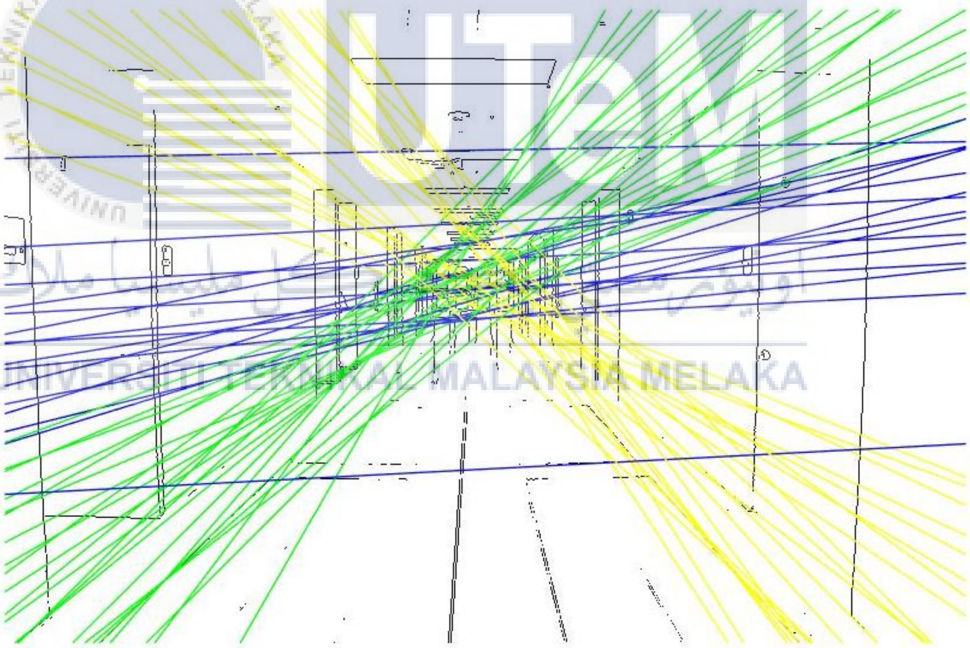


Figure 4.4: Lines from Hough Transformation ploted in edge image

The intersection point was only calculated from the intersect of yellow and green lines only. All points were plotted in the Figure 4.5 with black color. The intersects are between yellow and green lines only to reduce the error .

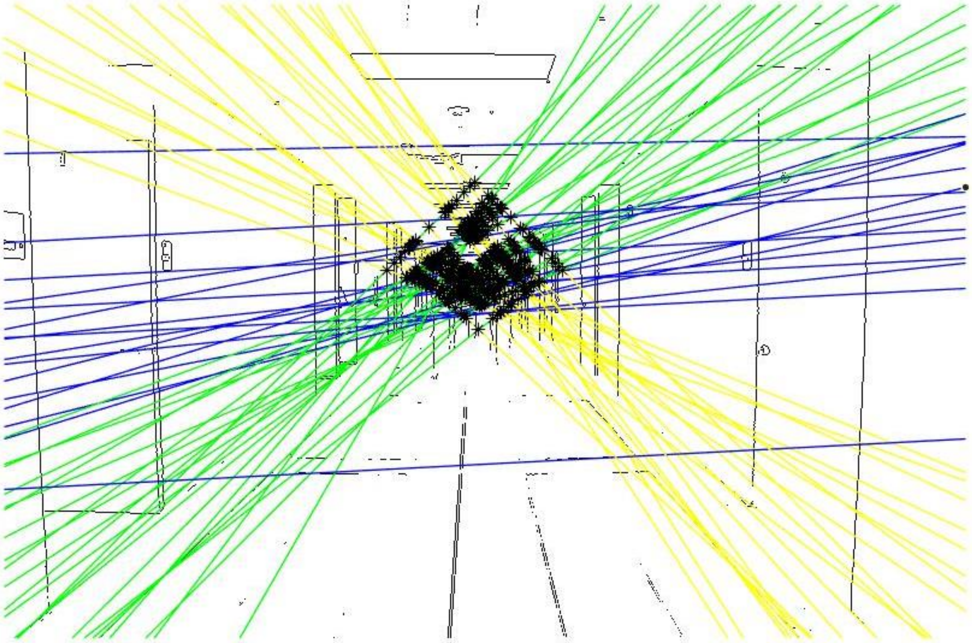


Figure 4.5: Intersection point of yellow and green lines

From the point of intersection, the vanish point was obtained by calculated it median. Through this median calculation, the exclude of the intersect with the vertical line, let the deviation of the vanish point to be more consistent. The vanish point was represented as a red dot on the Figure 4.6.

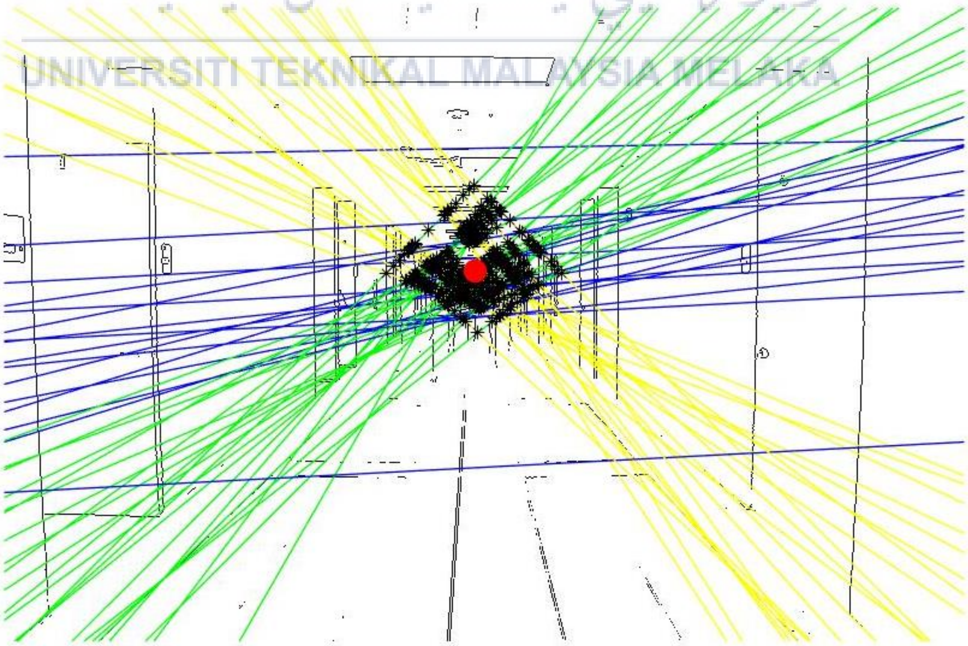


Figure 4.6: Vanish point (red dot)

This vanish point is used to compare with the center of the image. The distance of these 2 points in x-axis is the guideline for the mobile robot to move in center of corridor. In Figure 4.7, the lines are removed, and all the importance information were plot in the original photo. The distance of vanish point and the center of image (blue cross) in y-axis is shown in cyan color line and its value is shown with green color. The negative value means the vanish point is on the left-hand side while positive is in right-hand side.

Figure 4.7: The output of vanish point algorithm

The sample image is a photo of corridor with vanish point inside the image, but in some condition, the algorithm can’t detect the vanish point. In this kind of situation, this algorithm needs an alternative command for the mobile robot to move until find the vanish point.

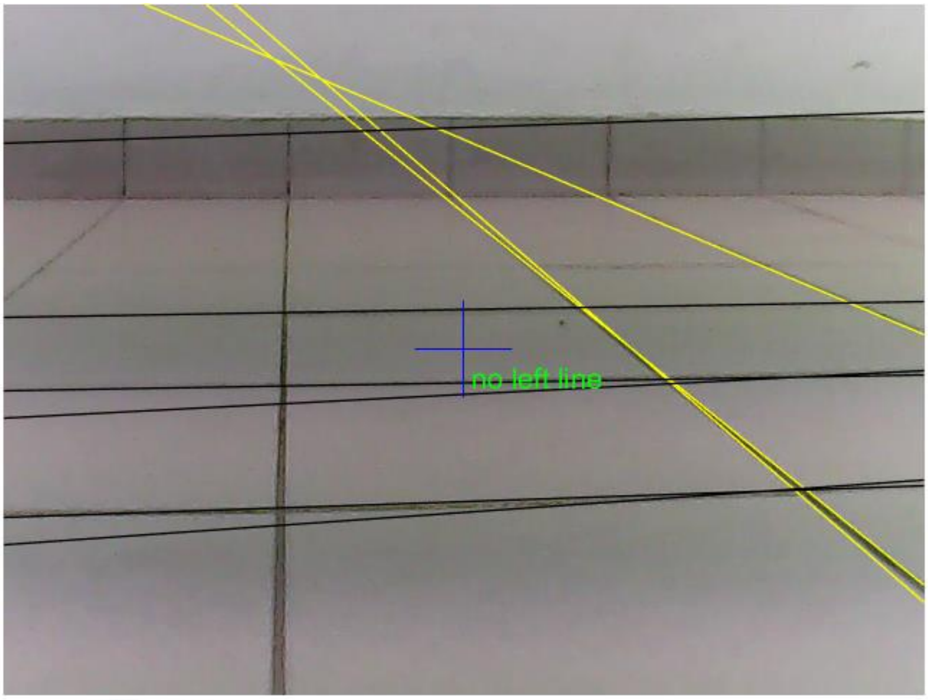


Figure 4.8: Image of corridor without the left line

For example, in the situation of Figure 4.8, the system cannot find vanish point inside the image, so the system will give an output of no left line. In this output, the robot control system will turn left in a small degree and capture a new image to calculate. This process will be continuing until it finds the vanish point. This process also can apply to situation of no right line.

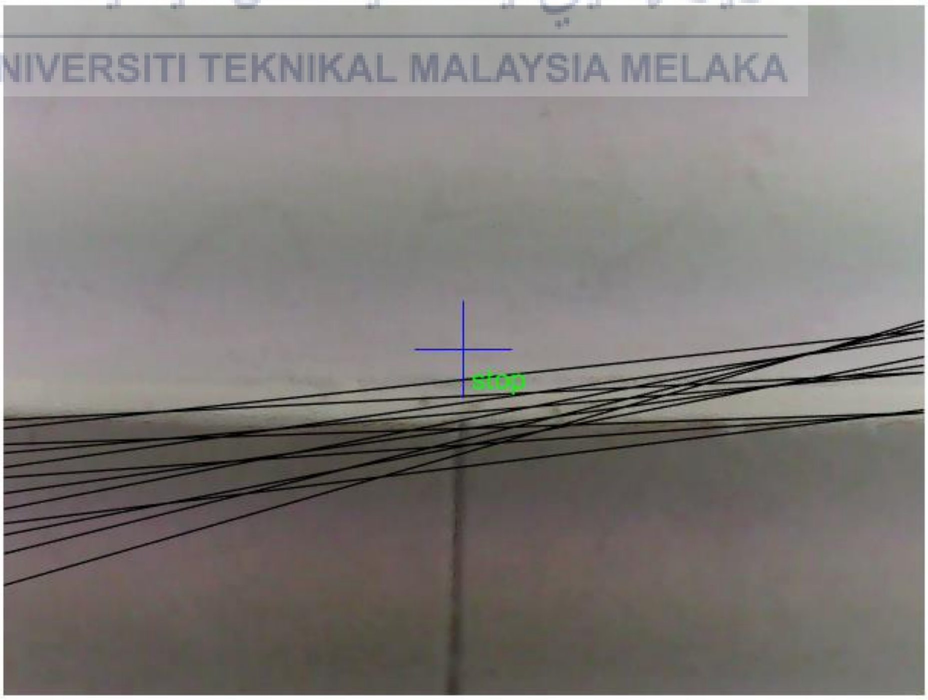


Figure 4.9: Image captured when reach a corridor end

When the robot reaches the end as in Figure 4.9, the left line and right line are disappeared and only left the horizontal line. In this situation, the algorithm will stop the movement of the robot and the navigation section will be end. The result display in figure 4.4 to 4.9 is used the coding in APPENDIX A. This coding is used to plot the line for a better observation only, the coding used during ran the robot is excluded plot the line and information into image. This is due to this process was time consume time and decrease the frame rate.

4.3 Test run in corridor with different environment.

In this experiment, the mobile robot with the system is tested in 3 location. These location are Sport Complex of UTeM, in front lift in third floor of FKE and in front of FKE lecture's office at first floor.

4.3.1 Test run in location 1

This experiment is repeated 15 times. All the data and observations are recorded in the Table 4.4.

Table 4.3: Number of error detected and success run in location 1

Number of trials	Frame capture per trial	Number of detection errors				Success in completing the task
		Vanish point	Left line	Right line	Unwanted line	
1	21	1	3		1	Fail
2	14		2			Fail
3	5			1		Fail
4	5			1		Fail
5	20		3	1		Fail
6	19		1			Success
7	16		2			Success
8	11					Success
9	17				1	Success
10	19					Success
11	16		1			Success
12	22					Success
13	9			1		Success
14	13		1			Success
15	15				1	Success
Total	222	1	13	4	3	10 success
		21				

$$Accuracy\ of\ navigation = \frac{10}{15} = 66.67\%$$

$$Accuracy\ of\ image\ detection = \frac{222 - 21}{222} = 90.54\%$$

From the Table 4.4 and calculation, the accuracy of the mobile robot move in this location is 67%. Although this success rate is low, but the accuracy of the image detection is high. This situation is occurring due to the system making the wrong calculation in crucial moment.

4.3.2 Test run in location 2

This experiment is repeated 15 times. All the data and observations are recorded in the Table 4.5.

Table 4.4: Number of error detected and success run in location 2

Number of trials	Frame capture per trial	Number of detection errors				Success in completing the task
		Vanish point	Left line	Right line	Unwanted line	
1	6					success
2	5					success
3	10			1		success
4	6				1	fail
5	17			1	2	fail
6	9					success
7	15				2	success
8	26		6	2		success
9	16				2	fail
10	9					fail
11	9				1	fail
12	17				1	success
13	14				4	fail
14	9					success
15	5					fail
Total	173	0	6	4	13	8 success
		23				

$$Accuracy\ of\ navigation = \frac{8}{15} = 53.33\%$$

$$Accuracy\ of\ image\ detection = \frac{173 - 23}{173} = 86.71\%$$

In this location, the success rate is low because of the surface of corridor end is not flat and causing the robot to recognize the wrong edge. In Figure 4.10, the green lines below the vanish point which from the corner will confuse the robot.

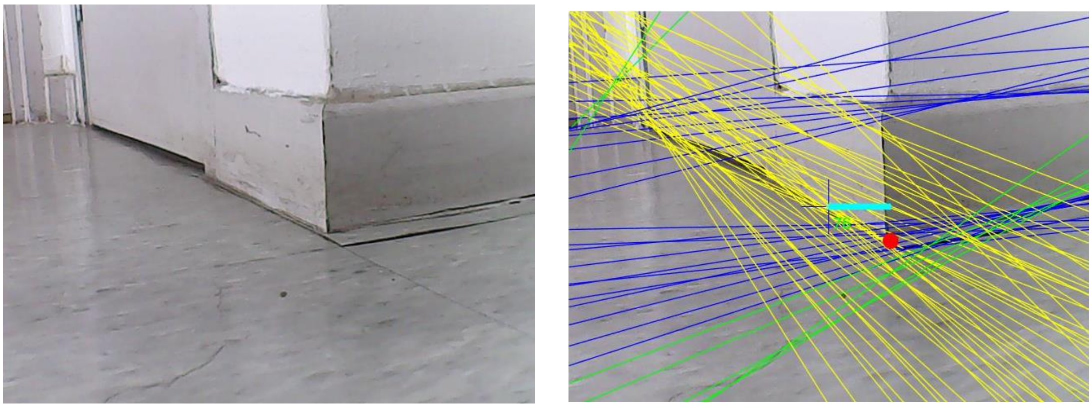


Figure 4.10: Before and after

Other than that, the robot also fails to enter the turn before it is making left turn. This cause the camera is obtain the image too far, and the corridor side too wide, so it can’t detect the corridor line.

4.3.3 Test run in location 3

This experiment is repeated 15 times. All the data and observations are recorded in the Table 4.6.

Table 4.5: Number of error detected and success run in location 3

Number of trials	Frame capture per trial	Number of detection errors				Success in completing the task
		Vanish point	Left line	Right line	Unwanted line	
1	4					Success
2	3					Success
3	6			2		fail
4	6				1	fail
5	7					Success
6	4					Success
7	6		1	1		fail
8	6				1	fail
9	11				3	fail
10	6			1	1	fail
11	1					fail
12	7		1	1		fail
13	4					Success
14	9			2	2	fail
15	9				1	Success
Total	89	0	2	7	9	6 success
		18				

$$Accuracy\ of\ navigation = \frac{6}{15} = 40\%$$

$$Accuracy\ of\ image\ detection = \frac{89-18}{89} = 79.78\%$$

The accuracy of navigation in this location is too low due to the obstacles (dustbin) in the side of corridor is blocking the corridor lines and create unnecessary lines. This line will affect the whole vanish point calculation.

4.3.4 Overall performance of the algorithm

To obtain the overall performance of the algorithm, the total errors and total frame capture I obtain from the experiment 4.4.1 to 4.4.3. This data is shown in the table 4.7.

Table 4.6: Total frames and errors

Location	Total frames capture	Total errors
1	222	21
2	173	23
3	89	18
Total	484	62

$$Accuracy\ of\ system = \frac{484-62}{484} = 87.19\%$$

The overall accuracy of system is acceptable, however the success rate in different location is vary depend on the environment. Besides that, the frame rate processed by the mobile robot is 1 frame per second. This will lead the collide or missed the importance data. Therefore, to overcome this problem, the processing speed need to be increased by run this algorithm in more powerful laptop.

4.3.5 Effect of environment

As the result shown in experiment 4.4.1 to 4.4.3, the lesser the obstacles in the corridor, the lesser the noise created in system. Hence the better the performance of the system. Besides that, the clearer the line of corridor, the better the performance.

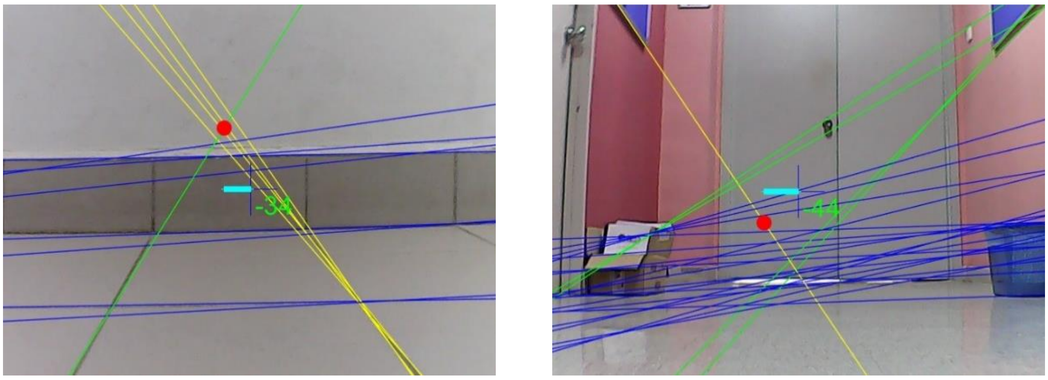


Figure 4.11: Image captured before reach end

In Figure 4.11, the image on the left which have the clearer line than image on right, so the system gets the vanish point more accurate. Other than that, the smaller the wide of the line detected, the closer the robot reach the end.

Figure 4.12: Image of the distance robot detect an end

This situation also depends on the angle of view of the camera. The angle of the camera used in this robot is not wide, so it only can get part of information in front it. In Figure 4.12, the distance of the robot and the end of the corridor (door) is longer, but the view of the robot is too far in front of its actual position. Figure 4.13 shown the robot view in situation in figure 4.12.

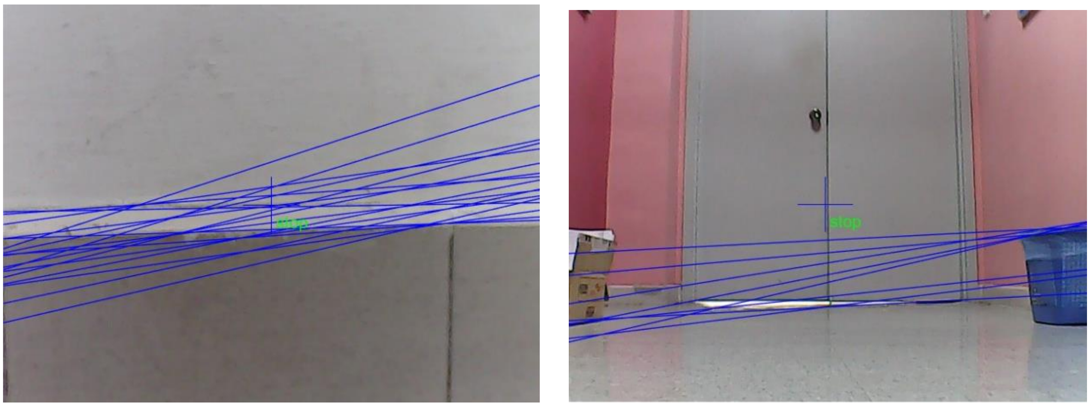


Figure 4.13: View of robot

Both view in the robot also detect the condition of the end of corridor but the distance of the robot stop is varied.

4.4 Summary

In conclusion, Sobel edge detection method is more suitable for this project than Canny edge detection. This is due to the requirement of the system is needed the importance line, not the detail on the image. By having the right edge detection method, an algorithm of vanish point is successfully developed. The system is success to calculate the vanish point present on the image.

After the system is developed, the performance of system is tested. The accuracy of calculation of the system is consistent and having high accuracy which is 87.19%. However, the accuracy of the navigation is not consistent due to the environment and limitation of the hardware and software used.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In the nutshell, the objectives of this project are achieved. In experiment 1 and 2, a navigation system was developed, this system is fulfilling the requirement of the objective 1. In these 2 experiments, a suitable edge detection method, Sobel is chosen for the algorithm of vanish point calculation. The algorithm developed can recognize the characteristic of different corridor condition and give corresponding command to robot to move. For the objective 2 and 3, it was achieved through the experiment 3. In this experiment, the algorithm is used to control the robot to move according to the calculation which fulfil the objective 2. At the same time, the performance (objective 3) is obtain by observe the result of the system and real condition of the robot.

The highest success rate in navigation is at location 1, however rate is 66.67% only. Overall accuracy of the system is 87.19% which is varied too much with the success rate. This situation is due to the frame rate processed by the system. The frame rate is 1 fps, so the robot having time gap of 1 second before getting next command. In this time gap, the robot might be missed an importance information or collide with the wall before the robot is able to respond. Other than that, the angle of view also affects the performance of the robot. The camera used in this robot is limited the information that can be obtain.

In conclusion, the performance of the algorithm is acceptable, but the overall performance of the robot is limited by the hardware use. By improve the hardware used, the robot performance can be increased.

5.2 Future Works

In future, the overall performance of this project can be improved through the upgrade of the hardware used. There are 3 majority part of hardware need to be improve.

First, the device used to run the algorithm need to be upgrade in term of it processing power. The higher the processing power, the more the frame rate can be process. When the frame rate increased, the respond time of the robot will be shortened. Hence the robot will be more sensitive to the environment. Besides that, the frame rate improved also can be cover the mistake calculate in the last frame. This situation can avoid the robot going the wrong path or collide. Therefore, the success rate of navigation will be increase rapidly.

The second part can be improving is the camera using in robot. The angle of view of camera will be effect to information that can be capture. The more information obtained, the better the performance of robot. This will also improve the result into more consistent.

The motor of the mobile robot is also an importance part in improve the performance. The motor used in this project is just a simple DC motor. By changing the motor with a motor with encoder and adjustable speed. The control system of the robot can be design in close circuit instead of open circuit used. This system can used to control the speed of motor and result in a smoother motion of robot.

Other that hardware part, the algorithm also can be improved. The algorithm ran in this project is ran in CPU. If the algorithm is rewrite into the format that can be support by the GPU, the frame will can be improved until almost closed to the frame rate of the camera which are 30 fps.

REFERENCES

[1] “PM’s Office guards stole Umno election funds: Najib, SE Asia News & Top Stories - The Straits Times.” [Online]. Available: https://www.straitstimes.com/asia/se-asia/pms-office-guards-stole-umno-election-funds-najib. [Accessed: 12-Nov-2018].

[2] E. Bayramoglu, N. A. Andersen, N. K. Poulsen, J. C. Andersen, and O. Ravn, “Mobile robot navigation in a corridor using visual odometry,” *Adv. Robot. 2009. ICAR 2009. Int. Conf.*, pp. 1–6, 2009.

[3] D. S. O. Correa, D. F. Sciotti, M. G. Prado, D. O. Sales, D. F. Wolf, and F. S. Osório, “Mobile robots navigation in indoor environments using Kinect sensor,” *Proc. - 2012 2nd Brazilian Conf. Crit. Embed. Syst. CBSEC 2012*, pp. 36–41, 2012.

[4] A. Adán, B. Quintana, A. S. Vázquez, A. Olivares, E. Parra, and S. Prieto, “Towards the automatic scanning of indoors with robots,” *Sensors (Switzerland)*, vol. 15, no. 5, pp. 11551–11574, 2015.

[5] D. O. Sales, F. S. Osório, and D. F. Wolf, “Topological Autonomous Navigation for Mobile Robots in Indoor Environments using ANN and FSM,” *1^a Conferência Bras. Sist. Embarcados Críticos*, 2011.

[6] I. Susnea, V. Minzu, and G. Vasiliu, “Simple, real-time obstacle avoidance algorithm for mobile robots,” *Recent Adv. Comput. Intell. Man-Machine Syst. Cybern.*, no. figure 2, pp. 24–29, 2009.

[7] V. Sezer and M. Gokasan, “A novel obstacle avoidance algorithm: ‘Follow the gap method,’” *Rob. Auton. Syst.*, vol. 60, no. 9, pp. 1123–1134, 2012.

[8] X. Li and B. J. Choi, “Design of obstacle avoidance system for mobile robot using fuzzy logic systems,” *Int. J. Smart Home*, vol. 7, no. 3, pp. 321–328, 2013.

[9] A. Paolillo, A. Faragasso, G. Oriolo, and M. Vendittelli, “Vision-based maze navigation for humanoid robots,” *Auton. Robots*, vol. 41, no. 2, pp. 293–309, 2017.

[10] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “Autonomous Obstacle Avoidance and Maneuvering on a Vision-Guided MAV Using On-Board Processing,” pp. 2472–2477, 2011.

[11] “Robot App Store | Apps For Every Robot!” [Online]. Available: http://www.robotappstore.com/Robopedia/Degrees-of-Freedom. [Accessed: 11-Nov-2018].

[12] M. Goebl *et al.*, “Design and Capabilities of the Munich Cognitive Automobile,” *Proc. IEEE Intell. Veh. Symp.*, pp. 1101–1107, 2008.

[13] P. Y. Shinzato and D. F. Wolf, *Features image analysis for road following algorithm using neural networks*, vol. 7, no. PART 1. IFAC, 2010.

[14] R. Grompone Von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, “LSD: A fast line segment detector with a false detection control,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, 2010.

[15] “小R讲堂-Arduino DS视频智能小车.” [Online]. Available: http://www.xiao-r.com/index.php/Study/catalog/cid/1. [Accessed: 11-Nov-2018].

APPENDICES

APPENDIX A GANTT CHART OF THE PROJECT

Task	2018				2019					
	Sept	Oct	Nov	Dec	Jan	Feb	Mar	April	May	June
Understand the title										
Research on the concept & technique										
Literature review on technique										
Research on hardware										
Design methodology										
Present proposal										
Conduct experiment										
Experiment 1										
Experiment 2										
Experiment 3										
Record all the observation and result										
Making discussion and conclusion										
Complete report										
Presentation on project										

APPENDIX B CODING TO DISPLAY THE RESULT

```
%obtaining the image capture and save by the robot
i = imageDatastore('D:\Users\MSI\Desktop\sampleimage\straight_testing\6.sucess_bestshot'); % location of image saved
iread = readall(i);
size(iread,1)

% main process : processed the image one by one
for a=1:size(iread,1)
    img =  readimage(i,a);
    [line , columns,H,theta,rho] = processimage(img);
    [leftline, rightline] = drawline(line, columns);
    [xin yin] = interseptpoint(line);
    [x,y] = vanishpoint(xin,yin);
    xc = size(img,2)/2;
    yc = size(img,1)/2;
    final = length (img);
    center = final / 2;

    % insert importance data into image
     if leftline == 0 &&  rightline == 0
         diff = 0;
        text((xc+5),(yc+20),'stop','FontSize',15,'Color','green');
    elseif leftline == 0
        diff = 0;
        text((xc+5),(yc+20),'no left line','FontSize',15,'Color','green');
    elseif rightline == 0
        diff = 0;
        text((xc+5),(yc+20),'no right line','FontSize',15,'Color','green');
     else
         diff = x - center;
     end
    plot(xc,yc,'+','markersize',50, 'Color', 'blue');
    if diff < 0
        plot((x:xc),yc,'+','markersize',5, 'Color', 'cyan');
        text((xc+5),(yc+20),num2str(diff),'FontSize',25,'Color','green');
    elseif diff > 0
        plot((xc:x),yc,'+','markersize',5, 'Color', 'cyan');
        text((xc+5),(yc+20),num2str(diff),'FontSize',15,'Color','green');
    end
end
% Process the image into the edge data
function [lines , columns,H,theta,rho] = processimage(Img)
    I=rgb2gray(Img);
    Ie=edge(I,'sobel');
    [H,theta,rho] = hough(Ie);
    P = houghpeaks(H,200,'threshold',ceil(0.2*max(H(:))));
    x = theta(P(:,2));
    y = rho(P(:,1));
    lines = houghlines(I,theta,rho,P,'FillGap', 5,'MinLength',100);
    [~, columns] = size(Ie);
    figure, imshow(Img)
    hold on
end
```

```
% Drawing all the line abtained from Hough Transformation
function [leftline, rightline] = drawline(lines, columns)
     leftline = 0;
     rightline = 0;
    for k = 1:length(lines)
        theta = lines(k).theta;
        xy = [lines(k).point1; lines(k).point2];
        x1 = xy(1,1);
        y1 = xy(1,2);
        x2 = xy(2,1);
        y2 = xy(2,2);
        slope = (y2-y1)/(x2-x1);
        xLeft = 1;
        yLeft = slope * (xLeft - x1) + y1;
        xRight = columns;
        yRight = slope * (xRight - x1) + y1;
        if (theta >= 30 & theta <= 70 )
            leftline = leftline + 1;
            plot([xLeft, xRight], [yLeft, yRight], 'LineWidth',1,'Color','green');
        elseif (theta >= -70 & theta <= -30 )
            rightline = rightline + 1;
            plot([xLeft, xRight], [yLeft, yRight], 'LineWidth',1,'Color','yellow');
        elseif ((theta >= 70 && theta <= 90 ) || (theta <= -90 && theta >= -70))
            plot([xLeft, xRight], [yLeft, yRight], 'LineWidth',1,'Color','blue');
        end
    end
end
% calculate the intercept point and plot on image
function [xin , yin] = interseptpoint(lines)
     a=1;
     xin = 0;
     yin = 0;
     %leftline = 0;
     %rightline = 0;
     for k = 1:length(lines)
         theta = lines(k).theta;
        if (theta >= -70 & theta <= -30 )
            xy_1 = [lines(k).point1; lines(k).point2];
            for k = 1:length(lines)
                theta = lines(k).theta;
                if (theta >= 30 & theta <= 70 )
                    xy = [lines(k).point1; lines(k).point2];
                    slopee = @(line) (line(2,2) - line(1,2))/(line(2,1) - line(1,1));
                    m1 = slopee(xy_1);
                    m2 = slopee(xy);
                    intercept = @(line,m) line(1,2) - m*line(1,1);
                    b1 = intercept(xy_1,m1);
                    b2 = intercept(xy,m2);
                    xintersect = (b2-b1)/(m1-m2);
                    yintersect = m1*xintersect + b1;
                     if  ~isnan(xintersect) && ~isnan(yintersect) && yintersect~=-Inf && xintersect~=-Inf
                         && xintersect<800 && yintersect<526 && xintersect>0 && yintersect >0
                         xin(a,1)=int16(xintersect);
                         yin(a,1)=int16(yintersect);
                         a=a+1;
                         plot(xintersect,yintersect,'m*','markersize',8, 'Color', 'black')
                     end
                 end
             end
         end
     end
end

%calculate the vanish poinnt and plot on image
function [x,y] = vanishpoint(xin,yin)
x = median(xin,'all');
y = median(yin,'all');
plot(x,y,'.','markersize',50, 'Color', 'red')
end
```

APPENDIX C ALGORITHM USED IN RUN THE ROBOT

```
clear all
count = 1;

% set the path need to be run by the robot
% s = going straight until reach end and stop
% l = going straight until reach end and turn left
% r = going straight until reach end and turn right
count = control('r',count)
count = control('s',count)
```

```
function count = control(con,count)
while true
    cam = ipcam('http://192.168.1.1:8080/?action=stream');
    cam = snapshot (cam);
    name = sprintf('%s_%d%s','D:\Users\MSI\Desktop\fyp program\corridorrun7\image',count,'.png');
    imwrite (cam,name);
    final = length (cam);
    center = final / 2;
    [line , columns] = processimage(cam);
    xy_1 = zeros([2,2]);
    [xin, yin, leftline, rightline] = interseptpoint(line);
    [x,y] = vanishpoint(xin,yin);

    % control of the robot
    if leftline == 0 &&  rightline == 0
        if con == 's'
            stop
            diff = 0;
            data = 2;                              % 2 = end of a command
            break
        elseif con == 'l'
            left
            pause(0.4)
            stop
            diff = 0;
            data = 2;
            break
        elseif con == 'r'
            right
            pause(0.4)
            stop
            diff = 0;
            data = 2;
            break
        end
    elseif leftline == 0
        left
        stop
        diff = 0;
        data = 3;
    elseif rightline == 0
        right
        stop
        diff = 0;
        data = 4;
     else
        [diff, data] = move(x,center, final)
    end
    z(count,1) = diff;
    z(count,3) = data;
count = count + 1
end
end
```

```
% process the image into the edge data
function [lines , columns] = processimage(Img)
    I=rgb2gray(Img);
    Ie=edge(I,'sobel');
    [H,theta,rho] = hough(Ie);
    P = houghpeaks(H,200,'threshold',ceil(0.2*max(H(:))));
    x = theta(P(:,2));
    y = rho(P(:,1));
    lines = houghlines(I,theta,rho,P,'FillGap', 5 ,'MinLength',100);
    [~, columns] = size(Ie);
    figure, imshow(Img)
    hold on
end

% calculate the intercept point
function [xin , yin, leftline, rightline] = interseptpoint(lines)
     a=1;
     xin = 0;
     yin = 0;
     leftline = 0;
     rightline = 0;
     for k = 1:length(lines)
         theta = lines(k).theta;
         if (theta >= 30 & theta <= 70 )
            leftline = leftline + 1;
        elseif (theta >= -70 & theta <= -30 )
            rightline = rightline + 1;
         end
        if (theta >= -70 & theta <= -30 )
            xy_1 = [lines(k).point1; lines(k).point2];
            for k = 1:length(lines)
                theta = lines(k).theta;
                if (theta >= 30 & theta <= 70 )
                    xy = [lines(k).point1; lines(k).point2];
                    slopee = @(line) (line(2,2) - line(1,2))/(line(2,1) - line(1,1));
                    m1 = slopee(xy_1);
                    m2 = slopee(xy);
                    intercept = @(line,m) line(1,2) - m*line(1,1);
                    b1 = intercept(xy_1,m1);
                    b2 = intercept(xy,m2);
                    xintersect = (b2-b1)/(m1-m2);
                    yintersect = m1*xintersect + b1;
                    if  ~isnan(xintersect) && ~isnan(yintersect) && yintersect~=-Inf && xintersect~=-Inf &&
                        xintersect<800 && yintersect<526 && xintersect>0 && yintersect >0
                        xin(a,1)=int16(xintersect);
                        yin(a,1)=int16(yintersect);
                        a=a+1;
                    end
                end
            end
        end
     end
 end

% calculate the vanish poinnt
function [x,y] = vanishpoint(xin,yin)
    x = median(xin,'all');
    y = median(yin,'all');
end
```

```
% move the robot based on the data obtained
function [diff, data] = move(x,center, final)
    offset = 10;
    diff = x - center;
    if x > center - offset && x <center + offset
        forward;
        data = 1;                                 % 1 = forward
    elseif x > 0 && x < center - offset
        left;
        data = 3;
        forward;                                  % 3 = left
    elseif x > center + offset && x < final
        right;
        data = 4;                                % 4 = right
        forward;
    else
        stop;
        data = 0;                                % 0 = stop
    end
end
```

```
% data sent to robot
function forward
    t = tcpip('192.168.1.1',2001);
    fopen(t);
    fwrite(t, [255, 0, 1, 0, 255])
    fclose(t);
end

function right
    t = tcpip('192.168.1.1',2001);
    fopen(t);
    fwrite(t, [255, 0,4, 0, 255])
    fclose(t);
    pause (0.05);
end

function left
    t = tcpip('192.168.1.1',2001);
    fopen(t);
    fwrite(t, [255, 0, 3, 0, 255])
    fclose(t);
    pause (0.05);
end

function stop
    t = tcpip('192.168.1.1',2001);
    fopen(t);
    fwrite(t, [255, 0, 0, 0, 255])
    fclose(t);
end
```

APPENDIX D DATA CONVERTED BY HOUGH TRANSFORMATION

point 1	point 2	theta	rho
[17,1]	[35,526]	-2	16
[731,1]	[703,526]	3	729
[1,62]	[800,62]	-90	-61
[124,1]	[133,526]	-1	123
[492,1]	[492,526]	0	491
[255,1]	[264,526]	-1	254
[631,1]	[622,526]	1	630
[401,1]	[373,526]	3	399
[80,526]	[625,1]	46	433
[188,1]	[661,526]	-42	139
[1,41]	[800,41]	-90	-40
[279,1]	[279,526]	0	278
[1,208]	[800,208]	-90	-207
[516,1]	[507,526]	1	515
[421,1]	[366,526]	6	418
[1,124]	[800,110]	89	123
[719,1]	[719,526]	0	718
[327,1]	[327,526]	0	326
[337,1]	[468,526]	-14	326
[376,1]	[376,526]	0	375
[450,1]	[450,526]	0	449
[410,1]	[410,526]	0	409
[1,256]	[800,256]	-90	-255
[353,1]	[362,526]	-1	352
[1,232]	[800,232]	-90	-231
[1,153]	[800,153]	-90	-152
[200,1]	[726,526]	-45	141
[1,420]	[800,434]	-89	-419
[318,1]	[469,526]	-16	305
[1,229]	[800,271]	-87	-228
[1,229]	[800,187]	87	228
[475,1]	[284,526]	20	445
[1,188]	[800,230]	-87	-187
[1,196]	[800,280]	-84	-194
[1,443]	[800,36]	63	394
[1,190]	[800,317]	-81	-187
[107,1]	[144,526]	-4	106

point 1	point 2	theta	rho
[1,278]	[800,194]	84	275
[1,253]	[800,211]	87	252
[363,1]	[465,526]	-11	355
[1,484]	[745,1]	57	405
[1,328]	[800,173]	79	321
[1,187]	[800,187]	-90	-186
[1,172]	[800,342]	-78	-167
[1,509]	[701,1]	54	411
[1,172]	[800,256]	-84	-170
[146,1]	[672,526]	-45	103
[333,1]	[379,526]	-5	331
[1,462]	[799,1]	60	399
[255,1]	[651,526]	-37	203
[1,295]	[800,168]	81	290
[1,377]	[800,377]	-90	-376
[1,121]	[800,335]	-75	-116
[1,98]	[800,343]	-73	-93
[1,382]	[800,91]	70	358
[1,64]	[800,437]	-65	-57
[54,1]	[800,504]	-56	30
[545,1]	[89,526]	41	411
[75,526]	[680,1]	49	445
[1,336]	[800,91]	73	320
[1,313]	[800,114]	76	303
[1,151]	[800,277]	-81	-148
[1,145]	[800,329]	-77	-140
[1,418]	[800,79]	67	384
[1,66]	[800,389]	-68	-60
[70,1]	[768,526]	-53	42
[261,1]	[602,526]	-33	218
[369,1]	[415,526]	-5	367
[1,105]	[800,396]	-70	-98
[1,40]	[800,412]	-65	-35
[230,1]	[655,526]	-39	178
[131,1]	[113,526]	2	130
[12,526]	[596,1]	48	398
[1,361]	[800,116]	73	344

point 1	point 2	theta	rho
[58,526]	[584,1]	45	412
[1,107]	[800,149]	-87	-106
[266,1]	[557,526]	-29	232
[313,1]	[558,526]	-25	283
[13,526]	[539,1]	45	380
[1,384]	[800,45]	67	353
[1,360]	[800,69]	70	337
[1,277]	[800,236]	87	276
[1,144]	[800,373]	-74	-137
[1,28]	[800,453]	-62	-24
[10,1]	[800,475]	-59	5
[245,1]	[771,526]	-45	173
[286,1]	[615,526]	-32	242
[1,469]	[646,1]	54	379
[1,420]	[727,1]	60	363
[1,270]	[800,115]	79	264
[1,299]	[800,215]	84	296
[392,1]	[419,526]	-3	390
[299,1]	[469,526]	-18	283
[385,1]	[459,526]	-8	380
[452,1]	[359,526]	10	444
[522,1]	[126,526]	37	416
[119,526]	[645,1]	45	455
[1,503]	[621,1]	51	390
[1,407]	[800,17]	64	365
[1,217]	[800,301]	-84	-215
[1,126]	[800,267]	-80	-123
[1,73]	[800,348]	-71	-68

point 1	point 2	theta	rho
[1,351]	[800,152]	76	340
[182,1]	[765,526]	-48	121
[290,1]	[569,526]	-28	255
[355,1]	[429,526]	-8	351
[1,1]	[47,526]	-5	0
[224,1]	[697,526]	-42	166
[373,1]	[345,526]	3	371
[500,1]	[197,526]	30	432
[10,526]	[659,1]	51	414
[68,526]	[740,1]	52	455
[1,248]	[800,136]	82	245
[1,198]	[800,156]	87	197
[1,403]	[800,361]	87	401
[1,146]	[800,188]	-87	-145
[130,1]	[800,488]	-54	76
[106,1]	[670,526]	-47	72
[279,1]	[661,526]	-36	225
[332,1]	[534,526]	-21	309
[302,1]	[404,526]	-11	295
[317,1]	[391,526]	-8	313
[292,1]	[265,526]	3	291
[395,1]	[321,526]	8	390
[429,1]	[346,526]	9	423