



HAL
open science

Development of new algorithm for improving accuracy of pole detection to the supporting system of mobility aid for visually impaired person

Muhammad Yusro

► **To cite this version:**

Muhammad Yusro. Development of new algorithm for improving accuracy of pole detection to the supporting system of mobility aid for visually impaired person. Other [cs.OH]. Université Clermont Auvergne [2017-2020]; Universitas Indonesia, 2017. English. NNT : 2017CLFAC107 . tel-02004329

HAL Id: tel-02004329

<https://theses.hal.science/tel-02004329v1>

Submitted on 1 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**DEVELOPMENT OF NEW ALGORITHM FOR IMPROVING
ACCURACY OF POLE DETECTION TO THE SUPPORTING
SYSTEM OF MOBILITY AID FOR VISUALLY IMPAIRED
PERSON**

DISSERTATION

MUHAMMAD YUSRO

Thèse soutenue le 18/10/2017

sous la direction de Monsieur Kun-Mean HOU

**DOCTORATE PROGRAM
UNIVERSITAS INDONESIA AND UNIVERSITY OF AUVERGNE
CLERMONT-FERRAND
2017**

DECLARATION ON NOT COMMITTING PLAGIARISM

I, the undersigned, do hereby declare that in accordance with the existing regulation of University Indonesia, that I have written this dissertation without any plagiarism.

If, in the future, I was found to have committed plagiarism, I will take full responsibility and will accept any suspension imposed against me by Universitas Indonesia and University of Auvergne, Clermont Ferrand.

Indonesia, June 2017

Muhammad Yusro

DECLARATION ON ORIGINALITY

This dissertation is the result of my own work,
and I have declared all sources quoted or referred.

Name: Muhammad Yusro

Signature

ATTESTATION PAGE

This Dissertation is submitted by:

Name : Muhammad Yusro
Study Program : Electrical Engineering
Disertation Title : Development of New Algorithm for Improving Accuracy of Pole Detection to the Supporting System of Mobility Aid for Visually Impaired Person

COUNCIL OF EXAMINERS

Supervisor-1 : Prof. Dr. Ing. Kalamullah Ramli, M.Eng
(Universitas Indonesia)

Supervisor -2 : Prof. Kun Mean HOU
(University of Auvergne Clermont-Ferrand, France)

Co-Supervisor-1 : Dr. Dodi Sudiana, M.Eng
(Universitas Indonesia)

Co-Supervisor-2 : Prof. Edwige PISSALOUX
(University of Rouen, France)

Examiner-1 : Prof. Bagio Budiardjo, M.Sc
(Universitas Indonesia)

Examiner-2 : Prof. Nadine PIAT
(Engineering school in Mechanics and Microtechniques, France)

THANK YOU NOTE

Praise to God, because with His blessings, I have successfully completed my dissertation. I wrote this dissertation as one of the requirements to attain the Doctorate of Engineering Degree from University of Indonesia (UI) and Computer Engineering from University of Auvergne Clermont-Ferrand (*Université Clermont d'Auvergne*), France.

I am aware that without the assistance and guidance of various parties, from the 1st day of my study here to the writing of my dissertation, it will be very difficult for me to complete this dissertation. Therefore, I would like to express my appreciation to:

- 1) Prof. Dr.-Ing Kalamullah Ramli and Prof. Kun Mean Hou as Supervisors and Dr. Dodi Sudiana and Prof. Edwige Pissaloux as Co-Supervisors.
- 2) Prof. Dr. Ir. Bagio Budiardjo, M.Sc. (UI) and Prof. Nadine Piat (ENSMM) as Examiners.
- 3) Dr. Christophe de Vault, Dr. Hong Ling Shi, Dr. Khalid El Gholami and all members of the team at *Système Multisensoriel Intelligent intégré Réparti (SMIR) Laboratoire d'Informatique, Modélisation et Optimisation des Systèmes (LIMOS)* University of Auvergne Clermont-Ferrand.

Finally, I pray to God could repay to everyone who have bestowed all their assistance to me. I hope this dissertation would be beneficial to the development of Science.

Depok, June 2017

Writer

CONTENTS

DECLARATION ON NOT COMMITTING PLAGIARISM	ii
DECLARATION ON ORIGINALITY	iii
ATTESTATION PAGE	iv
THANK YOU NOTE	v
CONTENTS.....	vi
LIST OF FIGURES	ix
LIST OF TABLES	xii
LIST OF ANNEX	xiii
LIST OF ABBREVIATION	xiv
LIST OF TERMINOLOGY	xvi
LIST OF PUBLICATIONS	xvii
CHAPTER 1 INTRODUCTION.....	1
1.1. Background	1
1.2. Identifying the Problems	4
1.3. Formulation of Problems.....	4
1.4. Goals of the Research.....	5
1.5. Contribution of the Research.....	5
1.6. Writing System.....	5
CHAPTER 2 LITERARY REVIEW.....	7
2.1. Definition and VIP needs	7
2.2. Assistive Technology for VIP	9
2.3. Smart Cane for VIP	12
2.4. VIP's assistive tool based on <i>Computer Vision</i>	20
2.5. Concept on Awareness on Active Multi-Sensor Context.....	24
2.6. Image Processing for Object Detection	26
2.6.1. Edge Detection	26
2.6.2. <i>Roberts</i> Operation	27
2.6.3. <i>Prewitt</i> Operation	28
2.6.4. <i>Sobel</i> Operation	29
2.6.5. Canny Edge detection.....	29
2.6.6. <i>Laplacian</i> Operation.....	31
2.6.7. <i>Laplacian of Gaussian (LoG)</i> Operation	32
2.6.8. Hough Transformation	33

2.6.9 OpenCV for Image Processing	35
CHAPTER 3 DEVELOPMENT AND IMPLEMENTATION OF SEES SYSTEM	37
3.1 Concept and Architecture on SEES	37
3.2 SEES for VIP's Mobility Function.....	39
3.3 SEES Activation Model	41
3.4 SEE-Stick Concept and Architecture.....	42
3.4.1. SEE-Stick Architecture	42
3.4.2. Multi-sensor Data Fusion for Navigation	44
3.4.3 Definition of Navigation Process.....	45
3.4.4 Definition of Navigation Route	46
3.5 SEE-phone Concept and Architecture	46
3.5.2. SEE-phone for Space awarenss function	48
3.6 Implementation and Test on RRT System	50
3.7 Implementation of SEES System.....	54
3.8 Data Communication model in SEES system.....	56
3.9 Implementation and Testing of Traffic Light Detection System.....	58
CHAPTER 4 DEVELOPMENT OF THE NEW POLE DETECTION ALGORITHM ...	62
4.1 Research Related to Object Detection	62
4.2 The Basis on the Development of YuRHos Pole Detection Algorithm	64
4.3 Optimising Canny and use of Houghlines function	66
CHAPTER 5 ANALYSIS ON THE PERFORMANCE OF YuRHoS POLE DETECTION ALGORITHM	77
5.1 YuRHoS Pole Detection Algorithm.....	77
5.1.1. Connect_line_segment function	78
5.1.2. Detect_pole Function	82
5.2 Evaluation of YuRHoS algorithm	84
5.3 Analysis Comparison on Pole Detection Algorithm	87
5.3.1. Testing for single pole and multi poles.....	88
5.4 SEE-Stick Real-Time Test	115
5.4.1. Specification on Hardware and Criteria of the Test.....	115
5.5 Result of <i>Real-Time</i> Test on Pole Detection.....	117
5.6 Analysis on Accuracy And Statistical Tests	120
5.6.1. Analysis on Accuracy Test	120
5.6.2. The Analysis on Time Response on Pole Detection	124
5.6.3. Analysis on Statistical Test.....	127

CHAPTER 6	129
CONCLUSION AND FUTURE RESEARCH	129
6.1 Conclusion of the Research	130
6.2 Future Research.....	130
LIST OF REFERENCE	132

LIST OF FIGURES

Figure 2.1 Bramblings model mobility.....	9
Figure 2.2 Travel Activity towards mobility and surrounding access [21]	9
Figure 2.3 Figure on assistive technology block	12
Figure 2.4 Complex procedure to detect obstacles [34].....	14
Figure 2.5 Teletact cane, University of Paris 11 [35]	15
Figure 2.6 Basic Scheme of Telemeter [35]	15
Figure 2.7 The UltraCane, University Leeds [36]	16
Figure 2.8 The 'K' Sonar, Canterbury University [37].....	16
Figure 2.9 The GuideCane, Michigan University [38]	17
Figure 2.10 The MiniGuide [39]	17
Figure 2.11 The “Smart Cane”, Indian Institute of Technology [31].....	18
Figure 2.12 The “iCane”, National Taiwan University [41]	18
Figure 2.13 SVETA experimental system [42]	21
Figure 2.14 Architectural system of NAVIG [45].....	22
Figure 2.15 AudioGuide Architecture [46]	22
Figure 2.16 System with PDA block diagram	23
Figure 2.17 Control of Visual Process by Supervisor.....	25
Figure 2.18 <i>Roberts</i> Operation	27
Figure 2.19 <i>Prewitt</i> Operation	28
Figure 2.20 <i>Sobel</i> Operation.....	29
Figure 2.21 Example of Gaussian mask with $\theta = 1.4$	29
Figure 2.22 Matrix of pixel measuring 5x5	30
Figure 2.23 Area for edge direction conversion	30
Figure 2.24 Test to change value 128 to 255.....	31
Figure 2.26 LoG(y, x) function as Mexican hat [63].....	32
Figure 2.27 LoG Operation	33
Figure 2.28 (a) xy coordinates (b) space parameter	34
Figure 2.29 Illustration of geometrical interpretation from parameter ρ and θ	35
Figure 3.1 Concept of SEES system.....	37
Figure 3.2 Traffic light model	39
Figure 3.3 Surface of the path specially made for VIP.....	40
Figure 3.4 Sensory Connection Model to the SEES system	41
Figure 3.5 SEE-stick system Architecture.....	43

Figure 3.6 Integrasi Linearized Kalman filter berbasis RISS/GPS	44
Figure 3.7 Engine condition during navigation process	45
Figure 3.8 Vector Navigation Route	46
Figure 3.9 Access Model from GPS receiver	48
Figure 3.11 Image Processing in android	48
Figure 3.12 TTS System	49
Figure 3.13 GPS Receiver Communication Model to the database server	50
Figure 3.14 The stages in building the RRT application	51
Figure 3.16 Example of the user's route.....	53
Figure 3.17 Display on the monitoring of route passed by the user.....	53
Figure 3.18 SEE-phone architecture.....	54
Figure 3.19 Chronology detection on pole and traffic light.....	55
Figure 3.20 Hardware Raspberry Pi 2 model B.....	56
Figure 3.21 Pseudo code of communication model between SEE-stick and SEE-phone .	57
Figure 3.23 Function of cvInRangeS.....	59
Figure 3.24 Thresholding Process	60
Figure 3.25 Simulation Test on Traffic Light Detection	61
Figure 4.1 (a) Process in detecting pole (b) Canny Edge detection Procedure	65
Figure 4.2 Algorithm of lines pairing for pole detection	66
Figure 4.3 Gaussian distribution with deviation standard of $\sigma=1$	67
Figure 4.4 Matrix 5x5	68
Figure 4.5 Matrix multiplication of 265,6	68
Figure 4.6 Result of matrix rounding off.....	68
Figure 4.7 Form of kernel matriks.....	68
Figure 4.8 (a) Input of grayscale photo (b) Outcome from Gaussian filter.....	69
Figure 4.9 Sobel operation Matrix	69
Figure 4.10 Gradient product from Sobel operation.....	70
Figure 4.11 Product of NMS method	72
Figure 4.12 Value of θ 0° group	72
Figure 4.13 Result of the group of $\theta=0^\circ$ value.....	73
Figure 4.14 3x3 matrix as mask	74
Figure 4.15 Graph of the hysteresis process	74
Figure 4.16 Result of the hysteresis process.....	75
Figure 4.17 Result of Houghlines process	75
Figure 4.18 Optimising Result in straightening the lines.....	76

Figure 5.1 YuRHoS pole detection Algorithm	78
Figure 5.2 Connect_line_segment algorithm.....	79
Figure 5.4 Algoritm Test and pole detection program of picture 1	85
Figure 5.6 Algorithm Test and pole detection program to photo 3.....	86
Figure 5.7 Working procedure of non-YuRHoS algorithm	87
Figure 5.8 Confusion Matrix	89
Figure 5.9 Labeling Process (Ground-Truth)	92
Figure 5.10 Images for One-Pole Simulation	93
Figure 5.11 The Result of Image Simulation of Single Poles.....	100
Figure 5.12 Image of Multi-Pole Simulation.....	103
Figure 5.13 The Result of Image Simulation of Multi-Poles.....	110
Figure 5.14 The Result of Single-Pole Simulation from a different Expert Judgment ...	113
Figure 5.15 The Result of Multi-Pole Simulation from Several Different Expert Judgments	115
Figure 5.17 Physical form of the SEE-stick	116
Figure 5.18 Coloured Testing Poles	116
Figure 5.19 Example of poles detected by SEE-stick.....	119
Figure 5.20 Example of poles undetected by SEE-stick.....	119
Figure 5.21 Test of Accuracy based on distance	120
Figure 5.22 Accuracy based on distance in the indoor and outdoor test.....	121
Figure 5.23 Accuracy based on distance in the pole coloured test - Black and Silver	121
Figure 5.24 Accuracy based on distance to the day and night time test.....	122
Figure 5.25 Accuracy based on camera resolution	122
Figure 5.26 Accuracy based on camera resolution in the indoor and outdoor test.....	123
Figure 5.27 Accuracy based on type of camera on the colour of pole (black and silver)	123
Figure 5.28 Accuracy based on type of camera to the day and night time test	124
Figure 5.29 The difference in the response time by resolution of the image black and silver poles	126

LIST OF TABLES

Table 1.1 Prevalency of visual impairment on elderlies- research in France [4]	1
Table 2.1 Visual Impairment Categories based on ICD [16].....	7
Table 2.2 Summary of various VIP's assistive tools in the form of cane	19
Table 2.3 Summary on various VIP's assistive tools based on computer vision.....	23
Table 2.4 1 st and 2 nd orders series in continuous and discrete form [54].....	26
Table 3.1 Type dan function of sensors in SEES	41
Table 3.2 SEES Mode.....	41
Table 3.3 Data on the position of the user in the database server	53
Table 3.4 Criteria in RGB value.....	59
Table 3.5 Example of data from the HSV level.....	60
Table 4.1 Summary of related studies and latest innovation in this research.....	64
Table 4.2 Grouping of the gradient direction	71
Table 5.1 The simulation procedure on poles.....	88
Table 5.2 Conclusions derived from the Simulation of Single Poles.....	100
Table 5.3 Conclusions derived from the Simulation of Multi- Poles.....	110
Table 5.4 Conclusions from The Result of a Single-Pole Detection of Several Expert Judgments	113
Table 5.5 Conclusions from The Result of a Multi-Pole Detection of Several Expert Judgments	115
Table 5.6 Criteria on the shape of the pole.....	117
Tabel 5.7 Criteria of the tested Camera.....	117
Table 5.8 Result of the test based on the criteria on pole testing	117
Table 5.9 Response time program algorithms for black pole image.....	125
Table 5.10 Response time program algorithms for silver pole image.....	125
Table 5.11 Relation between camera resolution, time , location, colour of object detected up to a distance of ≤ 300 cm	128
Table 5.12 Variable of the multivariate modelling which are connected to the detected object	129

LIST OF ANNEX

- Annex 1 : DDIP Research Activities
Annex 2 : Scheme Figure on Raspberry Pi 2 Model B

LIST OF ABBREVIATION

EBU	: European Blind Union
EGNOS	: European Geostationary Navigation Overlay Service
EOAs	: Electronic Orientation Aids
ETAs	: Electronic Travel Aids
GIMP	: GNU Image Manipulation Program
GIS	: Geographic Information Systems
GPS	: Global Position System
HCI	: Human Computer Interaction
HSV	: Hue, Saturation and Value
ICD	: International Classification of Diseases
ICT	: Information Communication and Technology
INS	: Inertial Navigation System
ITS	: Intelligent Transportation System
JNI	: Java Native Interface
JVM	: Java Virtual Machine
LiDAR	: Light Detection and Ranging
LKF	: Linearized Kalman Filter
LoG	: Laplacian of Gaussian
NAVIG	: Navigation Assisted by artificial Visual and GNSS
MLS	: Mobile Laser Scanning
NMS	: Non-Maximum Suppression
OR	: Odd Ratio
OpenCV	: Open Source Computer Visual
PCA	: Principal Component Analysis
PHP	: Hypertext Preprocessor
PLDs	: Position Locator Devices
PLOs	: Pole Like Objects
PoI	: Point of Interest
RFID	: Radio Frequency Identification
RGB	: Red, Green and Blue
RISS	: Reduced Inertial Sensor System
RPs	: Route Points
RRT	: Route Remote Tracking
SEES	: Smart Environment Explorer Stick
SRM	: Structural Risk Minimization
STA	: Speech to Action

STT	:	Speech to Text Conversion
SVM	:	Support Vector Machine
TIK	:	Teknologi Informasi dan Komunikasi
TOF	:	Time of Fly
TTS	:	Text to Speech Conversion
VIP	:	Visually Impaired Person
WGS84	:	World Geodetic System 1984
WHO	:	World Health Organization
YuRHoS	:	Yusro, Ramli, Hou, dan Sudiana

LIST OF TERMINOLOGY

Assistive Technology	: Technologies used by individuals with disabilities performing any work that is difficult or impossible.
Bivariate Analysis	: Analysis of two variables to determine whether there is a significant relations between the two variables.
Blindness	: Permanent blindness/visual impairment.
Chi-Square Test	: Statistical test to determine the relations between independent and dependent variables.
Computer Visual	: Technology reading information by using computer.
Context Awareness	: Important aspect for safe navigation for VIP in unfamiliar surrounding.
Houghlines Function	: Function to produce group of vertical lines data that are basic input for pole detection algorithm.
Hough Transform	: Technic to determine location of something in the form of image.
Hysteresis	: Process to eliminate broken lines to the edge object.
Low Visual (LV)	: Visual acuity reduced.
Mobilitas	: The ability to walk safely, comfortably, elegantly without any assistance.
Multivariate Analysis	: Analysing relationship of more than two variables. Used to determine which independent variable has an effect on dependent variables.
Navigation	: Planned controlled process from a movement renewing a position or an orientation.
Orientation	: Awareness of an individual on his/her physical position with the surrounding objects/the location of the object.
Regresi Logistic	: Producing <i>odds ratios/OR</i> between success or failure of an analysis.
Smart Stick	: Cane to assist VIP in walking which has been modified with sensors.
Thresholding Function	: Taking photos and returning them into binary image.
Visually Impaired Person (VIP)	: People suffering from visual impairment (<i>low visual dan blindness</i>).
Voxelization	: A process to add depth to images by using a set of cross sectional images known as data set volumetric.
White Cane / Stick	: Traditional equipment used to assist VIP to walk.
Wereable Computer	: Computer technology which can be used like a clothing or attached to the body of the user.

LIST OF PUBLICATIONS

1. Paper International Conference iN Bali, Indonesia. Authors: **M. Yusro** dan K. Ramli. Title: *GPS-based Navigation Devices for Visually Impaired People: Comparative Survey and Future Challenges* (QIR-‘Quality in Research’ Conference, July 2011, ISSN: 114-1284).
2. Paper International Conference in Gdansk, Poland (*The Best Paper Award*): Authors: **M. Yusro**, K.M. Hou, E. Pissaloux, H.L. Shi, K. Ramli dan D. Sudiana. Title: *SEES: Concept and Design of a Smart Environment Explorer Stick* (HSI-‘Human System Interaction’ Conference, June 6-8, 2013, Gdansk, Poland).
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6577804> (Scopus)
3. Paper International Conference in NSICST ‘*New and Smart Information Communication Science and Technology*’, September 18-20, 2013, Clermont Ferrand, France. Authors: **M. Yusro**, K.M. Hou, E. Pissaloux, K. Ramli, D. Sudiana, Z. Lizhong dan H.L. Shi. Title: *Design and Implementation of SEE-Phone in SEES (Smart Environment Explorer Stick)*.
http://edss.isima.fr/workshop/wp-content/uploads/paper_10.pdf
4. Chapter Book in Springer Verlag. Authors: **M. Yusro**, K.M. Hou, E. Pissaloux, K. Ramli, D. Sudiana, L.Z. Zhang dan H.L. Shi. Title: *Concept and Design of SEES (Smart Environment Explorer Stick) for Visually Impaired Person Mobility Assistance*.
http://link.springer.com/chapter/10.1007/978-3-319-08491-6_21# (Scopus)
5. Paper The 6th FTRA International Symposium on Advances in Computing, Communications, Security, and Applications (ACSA-14) di Busan, Korea. Authors: **M. Yusro**, K. Ramli, K.M. Hou, E. Pissaloux, D. Sudiana, L.Z. Zhang dan H.L. Shi. Title: *SEES System (Smart Environment Explorer Stick): Implementation of Data Communication between SEE-Stick and SEE-Phone on Mobility Assistance for Visually Impaired Person*.
http://www.ftrai.org/acsa2014/download/ACSA-14_registered_papers.pdf
6. Paper Journal of Engineering and Applied Sciences (accepted). Authors: **M. Yusro**, K. Ramli, K.M. Hou, dan D. Sudiana. Judul: *Development and Performance Analysis of Canny and Hough Transform based Pole Detection Algorithm as Mobility Aid for Visually Impaired Person on SEE-S (Smart Environment Explorer-Stick)*. (Scopus)

CHAPTER 1

INTRODUCTION

1.1. Background

Visual impairment is the main disability cause in the world and it is presumed that there is a continued increase of people suffering from visual impairment in the last 20 years [1]. The *World Health Organization (WHO)* data mentioned that in the world there are 285,000,000 people suffering from impaired vision (*Visually Impaired Person/VIP*), 39,000,000 of them suffer from total blindness and 246,000,000 suffer from low vision [2]. This data also mentioned that 90 % of the VIPs live in developing countries. Less than 65% of the people suffering from impaired vision are more than 50 years old which is 20% of the total world's population.

European Blind Union (EBU) in 2012 estimated that more than 30,000,000 of blind and partially blind people live in Europe with a ratio of 1:30 of Europeans suffering from *sight loss*, that is 4 times more than people who are totally blind [3]. A survey in France indicated that with the increase in people suffering from visual impairment exponentially based on age as indicated in Table 1.1.

Table 1.1 Prevalency of visual impairment on elderlies- research in France [4]

Type of Visual Impairment	Age (Year)				
	60-69	70-79	80-89	90-99	100+
Low Vision	3.06	5.92	14.10	23.13	33.71
Total Blindness	0.21	0.09	0.91	4.73	3.27
Total	3.27	6.01	15.01	27.86	36.98

Based on the national survey in Indonesia in 1993-1996, the rate of people suffering from blindness had reached 1.5% (approximately 3,600,000 people). Indonesia is in the top of the list of people suffering from visual impairment in Asia and 2nd in the world. From the national health survey in 2012, 2.45% of the Indonesian population suffers from handicaps. The proportion of people suffering from handicap is 39.97% suffers from multiple handicaps, 29.63% suffers visual

impairment, 10.26% suffers from walking disabilities and the remainder suffers from other handicaps [5].

From the above statistic data, the writer feels that there is a need to find a solution (in the form of aid tool) to assist VIPs to perform their activities independently and interact with the modern society. This issue concludes that sight is very important for every human being because information receives through the eyes and sight has a big effect on the perception of individual. Sight combine with other sensors, especially hearing, will allow a person to have global perception and perform certain action [6]. Blindness and visual impairment is main impedement towards daily life in accessing information, mobility, searching for roads, and interaction with the surrounding and other people as well as other activities. Blindness and visual impairment has become a very serious health and social issue.

Besides that for the VIPs, being able to move independently outdoor is very important [7]. Generally they are able to walk only along routes where they are used to walking or learn from a guide. This fact significantly restricted their daily life. On the other hand, they have a strong desire to walk on their own without any assistance from other people even in new place or when they lose direction.

Mobility allows an individual to interact with the outside world. Human autonomous mobility is an ability to reach point "B" from starting point "A" without the aid of other people [8]. However, independent mobility creates various problems, [9], two of them are orientation and space awareness. Orientation and space awareness are key elements to actual independent mobility. The present mobility system to assist in navigating VIPs in familiar/unfamiliar surroundings and indoor/outdoor space does not always support orientation and space awareness concept.

To acquire high mobility, the VIPs need to possess ability to detect obstacles in front of them. The main obstacle found by VIPs in their journey are obstruction in the form of poles such as street lamp post, traffic lights, street sign posts and trees which are generally standing vertically. Therefore, there is a need for an ability to detect and recognise obstacles in the form of poles/posts by a new algorithm developed from image processing system.

There are many researches related to obstacle detection directed both for public interest as well as personal interest. One of the many studies on obstacle detection are *Pole Like Objects*/PLOs performed on various facilities such as traffic light posts, lamp posts and street sign posts [10][11]. Many developed related research to this PLOs detection used *Mobile Laser Scanning* (MLS) which is a derivative from *Light Detection and Ranging* (LiDAR) technology as a distance sensory technology using distributed light property in search for distance and information of an object from a target that is aimed [12] [13]. However these researches emphasised on the need of urban management and environmental mapping rather than for people with handicap or suffering from physical disabilities [11][12][13].

A study on object detection based on image processing is also made to recognise the location of traffic light posts in the city [14]. This method implements image processing and pattern recognition in 3 stages, that is area where there are traffic light posts taken using segmentation method of colour threshold, deminishing *noise* with 2 types of filters resulting focus on traffic light posts and matching the template to validate the presence of traffic light posts. This system is tested on real-time on smart vehicle using camera installed on these vehicle.

From the literary studies made by the writer, it can be concluded that the average existing smart stick only gives assistance for running function in detecting obstacle objects and avoiding the obstruction. The existing *smart sticks* are very restricted in supporting orientation function. We need to observe all present *smart sticks* that are not equipped with *multi-sensor* connected to wireless access media such as access to web services (e.g., Google Map) for remote guiding aid and hints (*ubiquitous smart stick*).

This research develops an assistive device for VIP called *Smart Environment Explorer Stick* (SEES). The SEES System is designed to provide solution in overcoming restrictions in VIP navigation system, especially in optimizing the orientation and space awareness functions for their mobility.

1.2. Identifying the Problems

There are several problems related to aid tools for VIP and therefore in this research there is a need to perform the following:

1. Finding the right solutions in the form of aid tools for VIPs so that they are able to integrate in a modern society and assist them in their independence.
2. Mobility system to assist VIP navigation in familiar/unfamiliar surrounding, indoor and outdoor spaces which are not always supported by orientation and space awareness functions based on independent mobility.
3. Developing an aid tool for VIP to support orientation and space awareness functions.
4. Designing object detection algorithm which is able assist VIP in avoiding obstacles especially obstacles in the shape of poles.
5. Developing detection pole detection algorithm by using optimization of Canny edge detection and Hough Transformation.
6. Integrating two peripherals that are smart phone and smart stick into SEES system.
7. Adopting active context-aware multi-sensor concept to be applied to SEES system so as to assist VIP move safely and easily at any places.
8. Applying SEES system in *Intelligent Transportation System* (ITS) concept which means VIP is able to use this system to access urban transportation system such as cars, buses and trains.

1.3. Formulation of Problems

From identification of problem, this research will focus on development of SEES system as aid tool for VIP mobility and improve the accuracy of object detection by developing pole detection algorithm based on optimization of Canny edge detection and Hough Transformation.

1.4. Goals of the Research

The goals of this research are:

1. To develop SEES system as VIP's mobilisation aid tool by integrating SEE-stick and SEE-phone technologies.
2. To develop pole detection algorithm through distance measurement and search of a pair of vertical lines acquired from optimization of Canny Edge detection and Hough transformation.
3. Comparing the performance of YuRHos pole detection algorithm with similar pole detection algorithm so as to acquire a better pole detection accuracy.
4. Analysing the performance of pole detection algorithm so as to acquire accuracy in pole detection by testing real-time algorithm to different camera resolution and various condition of the field, that is indoor and outdoor, day time and night time.

1.5. Contribution of the Research

This research will give the following contribution:

1. The allocation of SEES system as mobility aid tool for people suffering from visual impairment (VIP). The SEES System is equipped with *global remote server* (iSEE), *embedded local server* (SEE-phone) and *smart stick* (SEE-stick). With a modular SEES system, the operation of this system can be made in accordance with the need and could be easily used at any place by VIP.
2. The improvement in object detection through developing YuRHos pole detection algorithm through distance measurement method and searching for pairs of vertical lines acquired from optimization of Canny edge detection and Hough Transformation.

1.6. Writing System

This research is divided into 6 parts that is Chapter 1 which explains about the introductory of the research. Chapter 2 describes about the literary review on

VIP and its problems, development in aid tools for VIP and object detection technology based on image processing. Chapter 3 describes the development and application SEES system as VIP mobility aid tools. Chapter 4 elaborates about the development of pole detection algorithm. Chapter 5 describes the analysis on the performance of YuRHoS algorithm, comparing with other algorithms, real-time tests and statistical analysis. Chapter 5 and Chapter 6 is the conclusion and further research.

CHAPTER 2

LITERARY REVIEW

2.1. Definition and VIP needs

Based on WHO standard *The International Classification of Diseases (ICD)* had divided visual impairment into 5 categories with a maximum vision of less than 6/18 *Snellen*, category 1 and 2 are in low vision category while 3, 4 and 5 categories are categorised to be blind [15][16]. The following Table 2.1 explained visual impairment categories according to ICD.

Table 2.1 Visual Impairment Categories based on ICD [16]

Categories	Visual Acuity (Snellen Chart)	WHO Category
0	6/6 – 6/18	Normal
1	< 6/18 – 6/60	Medium visual impoairment
2	< 6/60 – 3/60	Serius visual impairment
3	< 3/60 – 1/60	Blind
4	< 1/60	Blind
5	<i>No Light Perception</i>	Blind

Table 2.1 illustrated that in general, people with visual impairment (VIP) is divided into groups that is low vision (LV) and blindness. LV is defined as visual acuity less than 6/18 (people sufferring from LV can only see maximum distance of 6 meter while people with normal eyesight can see a maximum distance of 18 meter) to less than 3/60 (people sufferring from LV can only see a maximum distance of 3 meter while those with normal eyesight can see a maximum distance of 60 meter), LV has a broad view of a maximum of 20° (degree). Blindness is defined as sight acuity of less than 3/60 (people sufferring from impaired vision can only a maximum distance of 3 meter, but people with normal eyesight can see a maximum distance of 60 meter) sampai kurang dari 1/60 (penyandang kebutaan hanya dapat melihat pada jarak maksimal 1 meter but people with normal eyesight is able to see a maxumum distance 60 meter) where as those who cannot see, suffering from blindness, has a broad eyesight of a maximum 10° [17].

Visual impairment is also defined as functional restriction from the eyes or visual system due to congenital disease, aging and trauma [18] which cannot be corrected with spectacle, contact lens, medical treatment or surgery. This loss of sight will result in these people being incapable of performing routine daily activities such as reading, cooking, cleaning, taking care of personal hygiene, working, and driving a car. Visual impairment also; presents a challenge for independent mobility indoor and outdoor [19].

Generally VIP has special needs that is they want to walk safely in a complex road and obtain as well as learn specific information from topography of the local district [20]. The survey executed by Technical University of Lodz, Poland gave several definitions relating to the needs of visual impairment people (VIP) [21]:

- a. Space perception, understands geometrical structure of the surrounding environment, aware of himself in a room and knowing location of object around him.
- b. Orientation, to be aware of one's physical position in relation to object around him, direction of position following existing path as well as location of the destination. Orientation can be divided into 2 categories:
 - 1) Spacial Orientation, orientation in around close by, including ability to maintain his position on the track (for example on a side walk) without turning away from where the destination is.
 - 2) *Geographic Orientation*; orientasi in a space far away, which is interpreted as a knowledge needed to obtain present position to a destination that had been determined.
- c. Cognitive mapping is a mental ability in finding a way/road, frequently known as an aid to find a direction/road internally.
- d. Wayfinding, the ability to choose correct route in a *network of routes* from the starting point to the destination. This is a dominant cognitive ability in understanding *path layout* and continuously is aware of the location on the road.
- e. Navigation is planned control process of a renewed movement of a position and orientation (indoors near and far) along the chosen route towards the location of the destination. Navigation is frequently related to questions: where am I, where am I going and how I can get there.

- f. Obstacle detection and avoidance, ability to safely detect obstacles and finding a way to avoid it.
- g. Mobility, ability to walk safely, comfortably, elegantly without any aid. Mobility for VIP can also be interpreted as an activity which combines abilities such as space perception, orientation, finding their way, navigation, detection and avoiding obstacles. For people who suffers from visual impaired and blindness, mobility is the most important issue of their life [22]. Figure 2.1 shows *Bramblings* model from a mobility function which combines perception and orientation tasks [21].

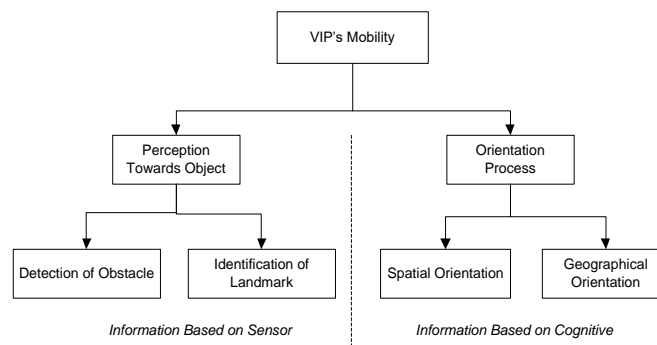


Figure 2.1 Bramblings model mobility

- h. Travel, a broader concept, not only covers mobility but access towards environmental surrounding as well as related to accessibility to the public space in urban surrounding that is buildings, public transportation, and marks Figure 2.2 describes sub section of travel activity for mobility and surrounding access.

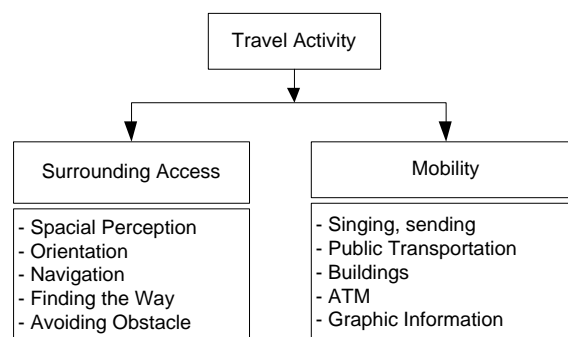


Figure 2.2 Travel Activity towards mobility and surrounding access [21]

2.2.Assistive Technology for VIP

The legal formal definition of assistive technology was first publicised in The Technology-Related Assistance for Individuals with Disabilities Act of 1988 (The

Tech Act). This definition was changed in 1994 and in 1998 was eliminated and changed to Assistive Technology Act of 1998 ("AT Act") [23][24]. The definition of assistive technology found in AT Act of 1998 is "...product, devices or equipment acquired commercially, modified or adjusted, used to maintain, upgrade or develop functional capability of handicapped individuals...."

Other than that other definitions on assistive technology are:

1. WHO [25] had defined assistive technology as a general term to every device or system which makes possible for individuals to perform their task when they are declared unable to perform it or increase the ease and safety in performing their task.
2. The National Center on Accessible Information Technology in Education, University of Washington [26] mentioned that assistive technology is the technology used by individual with disabilities to perform difficult or impossible task.
3. The King's Fund-United Kingdom [27] defined assistive technology as a kind of product or service which are designed to assist the independence of handicapped people or their parent.

Various assistive technologies have provided opportunities to everyone to access communication information technology in all human life sectors. The assistive technology still uses to indicate certain part from various tools in different sectors. This means that assistive technology is partially defined in accordance to the place or function when using it for example, an individual working in education sector inclines to define assistive technology as a product or service which assists learning process: access to computer (*hardware* and *software*), assistive communication device and digital learning assistive device.

Currently various types of assistive technologies for navigation system for VIP have been developed from simple ones to those with sophisticated and complex function. Since 1960, researchers have developed assistive technology, especially electronic equipment for navigation [28]. Especially for navigation system allocated for VIP categories were made which include: 1) increase in vision, 2)

replacement of vision and 3) substitution of vision. For devices in substitution of vision, it is divided in various sections, that is:

1. *Electronic Travel Aids (ETAs)* is a device with change information on surrounding which usually transmitted to the vision of a form which could be transmitted by other device (sensor).
2. *Electronic Orientation Aids (EOAs)*, is a device which provides orientation before and during travelling. The shape is an external device and/or can be carried by the users.
3. *Position Locator Devices (PLDs)*, is technology which includes GPS, European Geostationary Navigation Overlay Service (EGNOS) ...

The National Research Council's of ETAs provides performance guide for ETAs device [28][29], which are as follows:

1. Detection of obstacles in the path of travel from the ground to the head of the entire width of the body.
2. Information on the surface of the ground including texture and discontinuity.
3. Object detection bordering the travel path for the outline and projections.
4. Distance between object and cardinal direction information for straight line projection.
5. *Landmark* location and identification of information.
6. Information that enables self-socialization and mental environment mapping.
7. Ergonomic, operate with minimal interface with the natural sensor channels, unified unit, reliable, user selects sound / auditory or tactile modality, durable, easy to fix, robust, low power and cosmetically acceptable.

In answer to assistive technology challenge for VIP, research is being focussed on development of ETAs to increase confidence of users to travel independently. Therefore, the approach made is based on the on principles of electronic sonar/ultrasound, laser and vision devices [19].

In general, an electronic device as assistive technology for VIP is illustrated as in Figure 2.3 [21]. On the 1stfirst part there is environmental detection module,

built with detection method (active/passive) used to receive information on the condition of environment/physical phenomena. The next part is a process module which in a simple design, the transformation method used directly moves interface images. Other form is a system which process data acquired to select only important information. Finally, the interface implements an encoding scheme that converts some or all of the information into the form of output such as audio or vibration that can be heard / felt by VIP.

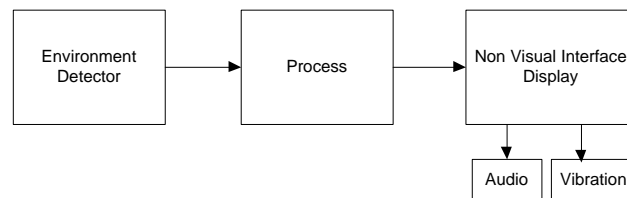


Figure 2.3 Figure on assistive technology block

More than 4 decades, several assistive devices had been developed to improve VIP's mobility. The development of information technology has opened an opportunity in the development of various assistive devices for VIP by utilising technology as well as from various disciplines such as GPS, computer vision, virtual reality, SMS speech, Text to Speech Conversion (TTS) as well as various softwares and hardwares so as to provide more complete information.

Several system impelmentations had integrated computer vision technology with GPS, GIS, and auditory display. However, with the present technological development, change in the urban surrounding as well as the increase in VIP's need, the existing system has flaws. Therefore, there is a need to improve it and improvement on the feature and function must be executed.

2.3.Smart Cane for VIP

Traditional assistive tool used to assist to walk/travel is white walking cane (*walking cane/white stick/blind cane*) and guide dog. Most VIP used walking cane/white cane to assist them to travel. A cane is a simple mechanic tool dedicated to detect static obstacles on uneven surface, holes and in steps, through the tip of the cane. According to [30] cane is one of VIP assistive tools and its function is very important. The most significant disadvantage from this tool is the need for skill

and training as well as the lack of information on the condition of the surrounding which the user is able to acquire.

The white cane used by VIP is relatively cheap, portable, and simple to use. Therefore, it is a popular tool as navigational assistive tool. However, the cane has two significant flaws [31], that is:

- 1) For practical control, it can only detect obstacle that are at knee level. Therefore cane is unable to detect obstacles which are potentially dangerous, such as protruding window panel, and surface that rises from the horizontal boundary.
- 2) The detection area of the cane has limited distance, 1-2 feet from the user. Certain obstacle (for example moving vehicle) cannot be detected and therefore, could endanger the cane user.

There are already several high tech assistive tools for VIP that have been developed. The working principle of this method [32] are: 1) The equipment in land surface detect a magnet inside the cane and then sent signal to the speaker; 2) the sensor inside the cane detects a magnet on the ground and then sent information to the user through vibration; 3) A special system in the ground detects signal from a user's transmitter and send a guiding message to the speaker; 4) The radio brought the message broadcast via an antenna mounted on the ground; 5) portable system which sent information on the position of the user using GPS signal.

This new guidance system utilises latest technologies such as data carrier, mobile communication, and portable computer as experimental test. Data code are recorded in a data carrier and then transferred to the user via a reader on the cane. The data is interpreted by the portable computer which transmits a guidance message to the speaker. For the user's safety, several system offer guidance message which uses a speaker which is installed on a fixed place and can be easily used. However this message is fixed and not adaptive. As the technology and the user's needs progress, various white cane innovation were made by modifying it into smart cane by installing several sensors on the cane.

The first white cane was found by an English photographer, *James Briggs*, who became blind in 1921 due to an accident so as to assist him in his mobility.

The white cane assists VIP to walk by giving a feedback in the form of vibration on their hand when there is an obstacle on the ground 1 meter away in front of him/her. When the obstacle is detected, the VIP must decide how to avoid the obstacle with cognitive integration (synthesis) from discrete vibration feedback.

However, there are many flaws in the usage of this cane [33]. This cane did not support the orientation and space awareness functions, because these functions greatly rely on the memory of the VIP concerned formed during the prior travel experience. The cane can only detect obstacle on the ground surface and the area of detection is less than 2 meter from the user. Recognition of the shape requires scanning of the obstacle with the cane and cognitive integration from discrete tactile feedback. This integration is the starting point to elaborate the procedure to avoid obstacles. Figure 2.4 describes how complex the integration of global information with the whole recognition of the form of obstacle when using the white cane [34].



Figure 2.4 Complex procedure to detect obstacles [34]

Several technological solutions have been proposed to assist the mobility of the VIPs. In this study, the writer will demonstrate the result of investigations on several assistive tools using cane that is white cane, smart stick, robotized smart cane and intelligent cane.

The white cane was changed into smart cane by adding and installing several sensors. The function of this sensor is usually to detect several obstacles which cannot be detected by the white cane such obstacles that protrudes or stairs to the top or to extend the reach of the stick. There is a feedback (vibration/audi) for the VIP to utilise the intensity reflection of the obstacle produced by the sensor and received by the cane. The whole form of the obstacle has to be reconstructed by

(integrated cognitive) by the VIP. The distance to the obstacle is estimated with the aid of Time-of-Flight (TOF) between the sensor and the nearest obstacle. The smart cane should be equipped orientation function as well as space awareness aid.

As shown in Figure 2.5, Teletact is hand held telemetre laser equipment which detects protruding obstacle with 1% error from distance of 10 cm to 10 m at a rate of 40 measurements per second [35]. The distance allocated is signaling through vibration feedback on the palm of the user. However, teletact does not have orientation function.

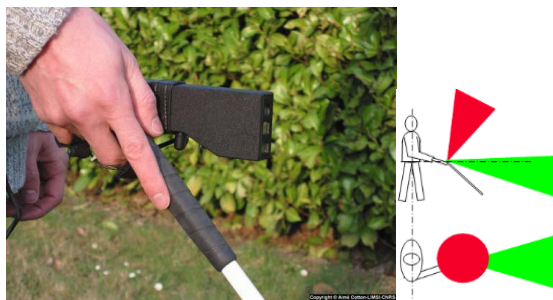


Figure 2.5 Teletact cane, University of Paris 11 [35]

Figure 2.6 demonstrates distance estimation principle based on TOF from a case on Teletact laser sensor designed at Paris University 11 [35]. The simplified basic principle of telemeter is a laser diode that emits a laser beam 670 nm collimated laser beam power 1mW. Collimation means that the laser beam has a minute divergency angle (in the order of milli radian) or the beam is focussed at a small angle. The laser beam strikes the distance D and produce laser point A . The image of laser point A that passes through the lense at CCD is A' . A' position on the line provide distance D .

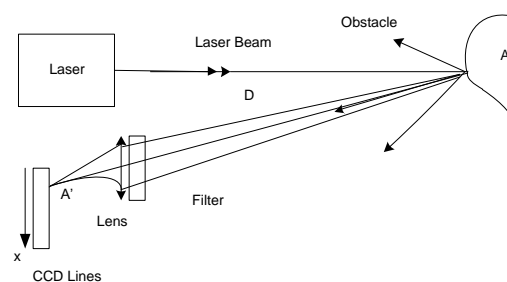


Figure 2.6 Basic Scheme of Telemeter [35]

Ultracane [36] which is a smart cane designed in *University of Leeds* as shown in Figure 2.7, offers to assist walking by detecting protruded obstacles. Ultracane uses ultrasonic sensor to detect obstacle that is in front of the user, including obstacles on the road and on the head of the user. *Ultracane* changes information in the form of vibration button on the handle which informs the user the position of an object and how far is the obstacle. The ultracane is able to adapt to the height of the user, but, it is unable to assist in the orientation function.



Figure 2.7 The UltraCane, University Leeds [36]

K-Sonar Smart Cane as in Figure 2.8 produced in *Canterbury University, New Zealand* also uses ultrasonic sensor to detect obstacles points. *K-Sonar* is mobility aid which emits sound which can be heard by user while walking. All obstacles are re-constructed (cognitive integration) by the user from these sonified points [37]. The users are able to determine the distance and location of the object and several other object features. Like *Ultracane*, *K-Sonar* is unable to assist in the orientation function.



Figure 2.8 The 'K' Sonar, Canterbury University [37]

GuideCane as can be seen in Figure 2.9 is a robotic smart cane designed to assist VIP in detecting specific obstacle and lies on the surface of the ground such as stairs going up. A group of ultrasonic sensors placed at the bottom tip of the *GuideCane* will change the data obtained from the sensor into vibration on the palm of the user [38]. This *GuideCane* is very pro-active in assisting the user in avoiding obstacle. *GuideCane* is also unable to assist in the orientation function.



Figure 2.9 The GuideCane, Michigan University [38]

The *Miniguide* shown in Figure 2.10 uses ultrasonic sensor to detect objects from various distances [39][40]. This tool is a hand unit tool by pressing the button to change the mode of operation. Detecting obstacles in 5 different distances is called mode of operation. This unit will vibrate to inform the user that there is an obstacle on the path. The nearer the user is from the obstacle, then the "miniguide" will intensify its vibration.

This *MiniGuide* has flaws such as it is unable to detect border line such as side walk and stairs. In addition, it does not have navigational assistance to guide the user from one place to another place. *Miniguide* does not give assistance for orientation function.



Figure 2.10 The MiniGuide [39]

SmartCane from *Indian Institute of Technology* [31] is shown in Figure 2.11. This project is aimed to develop 2 systems: 1) to detect obstacle above the knee and the alarm system uses ultrasound signal so as to increase horizontal and vertical detection scope; 2) to identify which is controlled by the user and housing system using radio frequency communication (RF). As like the previous tools, the Smart cane does not give assistance to orientation.



Figure 2.11 The “Smart Cane”, Indian Institute of Technology [31]

Intelligent Cane or iCane produced by *Taiwan National University* as shown in Figure 2.12 is to assist VIP to walk and to have an orientation. The operation principle of *iCane* is supported by exchange of data between standard cane, equipped with RFID, and RFID tag which is implanted in the surrounding [41]. RFID reader from *iCane* gets all the information data from RFID tag. The data is transmitted to the user via PDA headphone connected to the *Bluetooth* to the *iCane*. The information given is useful for walking in relation to urban space organisation such as crossroad, lift, stair and near shop/market.



Figure 2.12 The “iCane”, National Taiwan University [41]

The orientation function of the iCane, is assisted to providing optimal path (audio feedback) to reach the location targetted in relation to user's last location and database map. The main flaws of this system is that *iCane* needs high RFID component in the surrounding and maintenance cost. Other than that, the development/change in travelling route requires putting the right RFID tag on a new route.

Table 2.2 summarises the advantages and disadvantages of various VIP's assistive tools using canes as described above.

Table 2.2 Summary of various VIP's assistive tools in the form of cane

No	Name of Tools	Founder/ Developer	Advantages	Disadvantages
1	White Cane	James Briggs in 1921	Simple, cheap, manual, easy to get and use.	<ol style="list-style-type: none"> 1. Can only detect obstacles that is below the knees of the user. 2. The detection distance is very restricted, approximately 1 to 2 ft (30-60 cm) from the user
2	Teletact Cane	University of Paris 11	<ol style="list-style-type: none"> 1. Using laser technology, 2. Able to detect obstacle from 10cm to 10m. 3. There is vibration feedback on the palm of the user. 	There is no orientation function
3	Ultracane	University of Leeds	<ol style="list-style-type: none"> 1. Using ultrasonic sensor 2. Able to detect protruded obstacle which is infront of the user. 3. Change information to vibration form. 	There is no orientation function
4	K-Sonar	Canterbury University, New Zealand	<ol style="list-style-type: none"> 1. Using ultrasonic sensor 2. Emitted sound when finding obstacle. 	There is no orientation function
5	GuideCane	Michigan University	<ol style="list-style-type: none"> 1. Using many ultrasonic sensors 2. Detect specific obstacle and placed 	There is no orientation function

			on the surface of the ground.	
			3. Emitted vibration output to the palm of the user.	
6	MiniGuide	Many developers	1. Using ultrasonic sensor. 2. Change information to vibration form.	There is no orientation function
7	SmartCane	Indian Institute of Technology	1. Using ultrasound sensor to expand the horizontal and vertical detection scope. 2. Able to detect obstacle that are above the knee of the user. 3. Long distance control by using Radio Frequency.	There is no orientation function
8	Intelligent Cane	Taiwan National University	1. Using RFID sensor. 2. Using bluetooth technology for data communication 3. Using headphone transmitter to the user	1. Require many RFID components in areas where the users passes by 2. RFID tags must be replaced in accordance with the change of route where the user travel.

2.4.VIP's assistive tool based on *Computer Vision*

An important issue in independent mobility is to have the ability to detect physical obstacles. The ability to detect the presence of physical object can seriously be made by *vision system*. Several researches on navigation system for VIP had developed new tools by combining the function of GPS and GIS orientation and vision system. The mobility system based on vision, photos are taken and then sent to VIP via different sensoric channels such as sound and vibration.

The concept on using video camera as vision sensor is developed through portable system which uses the head where mono video camera is installed and translating the pictures taken by the camera and turn it into sound [42]. The

experimental system which changes picture into sound pattern had already been developed in 1992 [43]. This system is designed to give representation of hearing image from the human's hearing system as a step towards developing vision substitute tool for the blind. *Scenes* in front of the user is scanned through the moving his head up and down, and all the hearing data cognitively integrated and audio??? interpreted by the last user.

The SVETA system is a assistive tool system for the VIP which adopts change of image to sound technology [42]. Stereo camera is placed in front and *wearable computer* is placed on top of a helmet as illustrated in Figure 2.13. The stereo camera captures the visual information in front of the user. The photo taken is then processed using the methodology proposed in the *wearable computer*. Information on obstacles is then conveyed to the user using musical tone through stereo earphone.



Figure 2.13 SVETA experimental system [42]

The *Navigation Assisted by artificial VIsion and GNSS* (NAVIG) system [44] tries to combine vision and visual geographical location sign which will perfect the estimation of position by the GPS receiver. Figure 2.14 illustrates NAVIG system which has two main modules of the location of the object to allocate object needed by the user and the positioning function needed by the user to detect the visual object which were not projected to the user, but used to improve the GPS position. This system uses *Spike-Net NAVIG library* which provides fast algorithm based on human visual research.

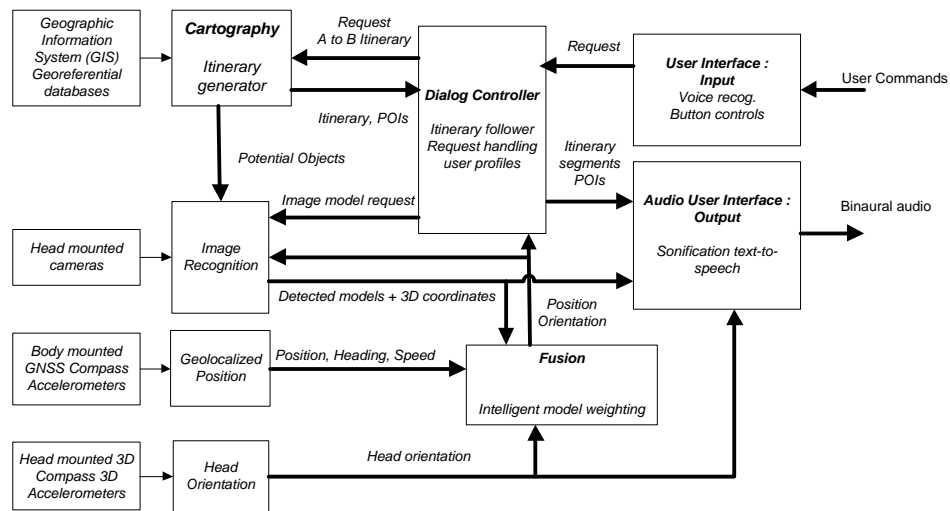


Figure 2.14 Architectural system of NAVIG [45]

Other studies proposed an Audio Guide system which integrated GPS and vision to the *Personal Data Assistant (PDA)* [46]. This kind of model has actually been developed by many researchers by using smartphone. This system utilises technological development in *computer vision*, GPS, GIS and *auditory display*. This system is to give the VIP context signs so as he/she navigates safely in a dynamic situation. The PDA camera catches dynamic environmental information and landmark information along the route from the mapping database using *Visual Studio eSuperMap*. Figure 2.15 illustrates architecture of Audio Guide system.

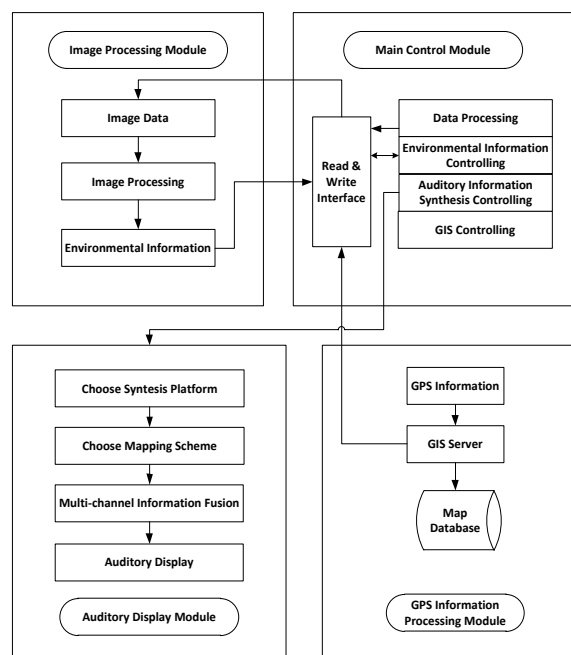


Figure 2.15 AudioGuide Architecture [46]

The Audio Guide system is able to monitor quickly the condition of the surrounding by using camera which is installed on the PDA. This tool can also detect road and turning corner information by image processing technic which is later conveyed to the user in the form of sound information.

The navigation system for VIP is further developed by combining GPS, camera and ultrasonic sensor [47]. This system receives information on the surrounding by using GPS and ultrasonic sensor is used to detect distance data, and then the user is informed as in Figure 2.16. The camera is used to evaluate the situation in front of the user like colour. In the case of short distance obstacle, the vibrator will sent voice and vibration signal to the user. This system uses *Bluetooth wireless access medium* to transmit data.

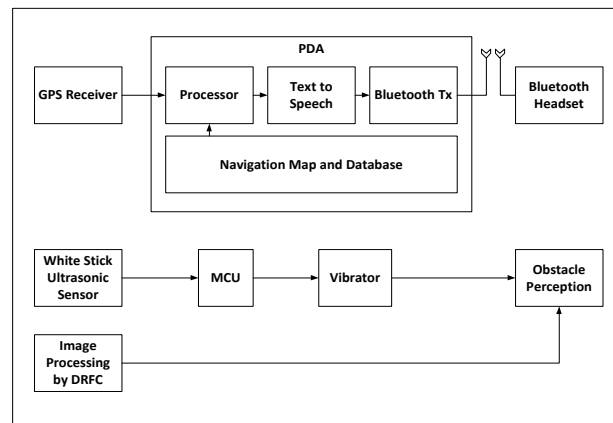


Figure 2.16 System with PDA block diagram

The conclusion from several VIP's assistive tools based on *computer vision* is that the system built has combined orientation and visual functions, the system uses camera and ultrasonic to detect obstacles. However, this system does not have long distance route monitor to track the user's location. Table 2.3 summarises the advantages and disadvantages of various VIP assistive tools based on *computer vision* as described above.

Table 2.3 Summary on various VIP's assistive tools based on *computer vision*

No	Name of Tools	Founder/ Developer	Advantages	Disadvantages
1	SVETA	G. Balakrishnan and team	1. Uses camera which is installed on the helmet on top of the user's head	This system is not equipped with long distance monitor to

			2. Change information to voice	track the location of the user.
2	NAVIG	Brian F G Katz and team	1. Combine vision principle with GPS receiver. 2. Change information to voice.	Generally, the use of this system is complicated for the user in executing travelling activity.
3	AudioGuide	Jie Xu and team	1. Integrating GPS and vision to the PDA equipment. 2. Camera Sensor takes picture and synchronises it with the database with database. 3. Change information to voice.	This system is not equipped with long distance monitor to track the location of the user.
4	Navigation System with PDA	Do-Hoon Kim, Heung-Gyoon Ryu	1. Combines GPS, camera and ultrasonic functions. 2. Change obstacle information into vibration and voice. 3. Uses Bluetooth communication to transmit data.	This system is not equipped with long distance monitor to track the location of the user.

2.5. Concept on Awareness on Active Multi-Sensor Context

The first and one of the most basic task now in building Human-Computer interaction (HCI) system is sensory perception and gathering of surrounding information. The use of multi-sensor is the key to building smart and reliable system. It also provides necessary information in detail. Besides that, integrating the output from multi-sensor frequently maybe the only way to acquire information that are needed when a single sensoric asset is not sufficient [48].

The concept on active multi-sensor adopted from smart system allows the system to obtain fast and accurate information with its context. Active context-aware multi-sensor means, according to the context a subset of sensor is activated to acquire environmental data to provide complementary and reliable information. Any external information can be used to read situation of anyone, place or object in a certain surrounding [49].

The *multi-sensor data fusion* technique is taken from variety of traditional disciplines including processing digital signal, statistical estimation, control theory, artificial intelligence, and classical numeric method [48]. Figure 2.17 illustrates the interaction of multi-sensor introduced as coordinate concept in visual process controlled by a supervisor [50].

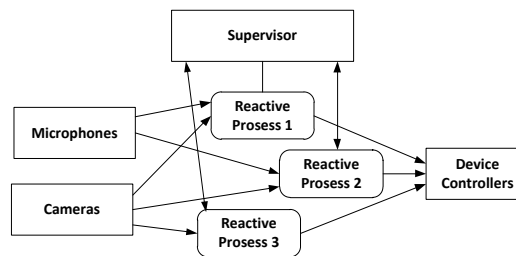


Figure 2.17 Control of Visual Process by Supervisor

The existing navigation system are able to provide some degree of assistance but are still unable to solve important aspects on context awareness, safety and usefulness. In the study on navigation, context awareness is an important aspect for a safe navigation for VIP in unfamiliar surrounding [22]. By providing maximum awareness on the surrounding and context, without requiring modification to the existing infrastructure, navigation experience significantly upgrade the VIP's navigation capability in any type of surroundings.

The concept on active multi-sensor awareness context for VIP means that systemic behavioral is controlled as a set of condition, a transition from one condition to another driven by the data provided by the sensor after going through integration of data. One feedback to the user is a guidance message following every condition transition. For example, when VIP walks along a straight pavement, the main function needed are obstacle detection and avoidance. In this case, ultrasonic sensor is considered as the most important active sensor than other sensors. Similarly, when the VIP wants to cross the street, the camera sensor used to see the status of the traffic light becomes the main sensor to be used compared to other sensors.

2.6. Image Processing for Object Detection

2.6.1. Edge Detection

Edge Detection is an approach frequently used in segmenting pictures based on the change in sharp intensity of the picture. The definition of edge in this context is a set of connected pixel located in two areas [51]. The edges are a number of places in the image with strong intensity contrast. Edge Detection is made by increasing the visibility of the lines to the edge image. Edge usually appears at the image location which represents object boundary.

Edge detection is divided into 2 groups, that is first, 1st order edge detection which uses 1st Order differential or series [52]. Several edge detection for the 1st order is the one using *Prewit*, *Roberts* and *Sobel* operation [53]. Second is edge detection of the 2nd Order which uses the 2nd Order differential or series. Example of the 2nd group is *Laplacian of Gaussian* (LoG). The Laplace edge detection operation is based on 2nd order series of edge detection operation which is sensitive towards disturbances.

Table 2.4 gives a picture on the definition on the series 1st and 2nd orders both in the continuous as well as discrete form. [54]. The discrete form is very useful in producing edge detection.

Table 2.4 1st and 2nd orders series in continuous and discrete form [54]

Series	Continuous Form	Discrete Form
$\frac{df}{dx}$	$\lim_{\Delta x \rightarrow 0} \frac{f(y, x + \Delta x) - f(y, x)}{\Delta x}$	$f(y, x + 1) - f(y, x)$
$\frac{df}{dy}$	$\lim_{\Delta y \rightarrow 0} \frac{f(y + \Delta y, x) - f(y, x)}{\Delta y}$	$f(y + 1, x) - f(y, x)$
$\nabla f(y, x)$	$\left[\frac{df}{dy}, \frac{df}{dx} \right]$	$[f(y, x + 1) - f(y, x), f(y + 1, x) - f(y, x)]$
$\frac{d^2f}{dx^2}$	$\lim_{\Delta x \rightarrow 0} \frac{\left(\frac{df}{dx}\right)(y, x + \Delta x) - \left(\frac{df}{dx}\right)(y, x)}{\Delta x}$	$f(y, x + 1) - 2f(y, x) + f(y, x - 1)$
$\frac{d^2f}{dy^2}$	$\lim_{\Delta y \rightarrow 0} \frac{\left(\frac{df}{dy}\right)(y + \Delta y, x) - \left(\frac{df}{dy}\right)(y, x)}{\Delta y}$	$f(y + 1, x) - 2f(y, x) + f(y - 1, x)$
$\nabla^2 f(y, x)$	$\frac{d^2f}{dx^2} + \frac{d^2f}{dy^2}$	$f(y, x + 1) + f(y, x - 1) - 4f(y, x) + f(y + 1, x) + f(y - 1, x)$

To look for a strong edge and direction of the location (x,y) from picture f is a gradient with a notation ∇f defined as vector to the Equation 2.1 [55].

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix} \quad (2.1)$$

The amount of vektor is calculated using Equation 2.2.

$$\nabla f = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} = \left[\left(\frac{df}{dx} \right)^2 + \left(\frac{df}{dy} \right)^2 \right]^{1/2} \quad (2.2)$$

To simplify the calculation, the root operation to the Equation 2.2 is deleted until the amount of the Vector is obtained through Equation 2.3.

$$\nabla f \approx G_x^2 + G_y^2 \quad (2.3)$$

In the 2nd order series which usually uses image processing, is calculated by using *Laplacian* as illustrated in Equation 2.4.

$$\nabla^2 f(y, x) = \frac{d^2 f(y, x)}{dx^2} + \frac{d^2 f(y, x)}{dy^2} \quad (2.4)$$

2.6.2. Roberts Operation

One of the oldest and simplest operation is Roberts operation as the first order which uses partial differential operation to detect edges [53]. This operation consists of a pair of 2×2 kernel convolution [56]. To detect the edges, Roberts Operation uses an approach between two pixels which are close by diagonally from amplitude gradient. Small filter will expedite computation of the Robert operation, but it has a disadvantage that is it can be greatly affected by roaring sound. Besides that the Roberts operation gives a weak response towards edge, unless the edge is very sharp [57].

	x	x+1			
y	z ₁	z ₂			
y+1	z ₃	z ₄			

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

(a) Position of image f

(b) Matrix G_x

(c) Matrix G_y

Figure 2.18 Roberts Operation

The form of *Roberts Operation* is illustrated in Figure 2.18 with f as the image used in *Roberts Operation*. The value in *Roberts Operation* (y, x) is defined by Equation 2.5.

$$r(y, x) = \sqrt{(z_1 - z_4)^2 + (z_3 - z_2)^2} \quad (2.5)$$

Where

$$z_1 = f(y, x), z_2 = f(y, x + 1), z_3 = f(y + 1, x), \text{ dan } z_4 = f(y + 1, x + 1)$$

2.6.3. Prewitt Operation

Prewitt Operation is 3×3 matrix, which determines the pixel gradient value with 8 points around it [58]. The *Prewitt* and operator *Sobel* operations uses the same differential and filter operation. The only difference is that the template did not use the same picture. The *Prewitt* operation is almost similar to *Sobel* operation which are used to detect horizontal and vertical edge in a picture [56]. The form of *Prewitt Operation* is illustrated in Figure 2.19.

	x-1	x	x+1
y-1	z ₁	z ₂	z ₃
y	z ₄	z ₅	z ₆
y+1	z ₇	z ₈	z ₉

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

(a) Position of image f (b) Matrix G_x (c) Matrix G_y

Figure 2.19 *Prewitt Operation*

To expedite calculation, parts which has a zero value are not processed which will result calculation of *Prewitt Operation* to be as in Equation 2.6.

$$r(y, x) = \text{sqrt}((f(y-1, x-1) + f(y, x-1) + f(y + 1, x-1) - f(y-1, x + 1) - f(y, x + 1) - f(y + 1, x + 1))^2 + (f(y + 1, x-1) + f(y + 1, x) + f(y + 1, x + 1) - f(y-1, x-1) - f(y-1, x) - f(y-1, x + 1))^2)) \quad (2.6)$$

2.6.4. Sobel Operation

The *Sobel* Operation is mostly used in the image processing, especially in edge detection algorithm. Technically, *Sobel* is a discrete differential operation which calculated the approach of intensity function gradient Figure [59]. Sobel is a filtering operation form used to extract edges. The *Sobel* operation can be seen from Figure 2.20. The *Sobel* Operation is more sensitive towards diagonal edge than vertical and horizontal edges [60].

	x-1	x	x+1
y-1	z ₁	z ₂	z ₃
y	z ₄	z ₅	z ₆
y+1	z ₇	z ₈	z ₉

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

(a) Position of citra f

(b) Matrix G_x (c) Matrix G_y Figure 2.20 *Sobel* Operation

2.6.5. Canny Edge detection

Canny edge detection is a popular edge detection algorithm. It is developed by *John F. Canny* in 1986 which is a multi-step algorithm [61]. This algorithm gives a small error level, localising edge points (the distance between the edge pixels detected is in actual fact very short) and only gives one response to one edge.

In general there are 6 Canny edge detection steps which is elaborated as follows [54]:

- 1) Filter the image so as to eliminate the roaring sound. This process is made by using *Gaussian* filter with a simple *mask*. The *Mask* used is much smaller that the size of the image (Figure 2.21).

$$1/115 = \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Figure 2.21 Example of *Gaussian* mask with $\theta = 1.4$

- 2) Using *Gaussian* operation to obtain *edge strength*. The image gradient is calculated with Equation 2.7:

$$|G| = |G_x| + |G_y| \quad (2.7)$$

3) Counting direction of pixel gradient by using Equation 2.8:

$$\theta = \tan^{-1}(G_y/G_x) \quad (2.8)$$

4) Connecting the directions of the edge pixels in a direction which tracked from an image. For example from the array of 5x5 pixels (Figure 2.22), we can see that in z pixel there are 4 directions that is 0° , 45° , 90° and 135° .

```

x x x x x
x x x x x
x x z x x
x x x x x
x x x x x

```

Figure 2.22 Matrix of pixel measuring 5x5

The edge direction is placed into one of the categories from 4 directions based on the area indicated in Figure 2.23, with its conversion rule:

$$\text{Direction} = \begin{cases} 0^\circ & 0 \leq x \leq 22,5 \text{ atau } x \geq 157,5 \\ 45^\circ & 22,5 \leq x \leq 67,5 \\ 90^\circ & 67,5 \leq x \leq 112,5 \\ 135^\circ & 112,5 \leq x \leq 157,5 \end{cases} \quad (2.9)$$

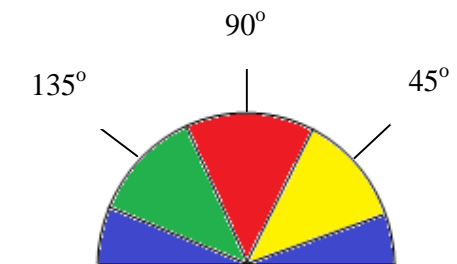


Figure 2.23 Area for edge direction conversion

With the following requirement: 1) All edge direction which are between 0° and $22,5^\circ$ as well as $157,5^\circ$ and 180° (blue) is changed to 0° ; 2) all edge direction which is approximately between $22,5^\circ$ and $67,5^\circ$ (yellow) is changed to 45° ; 3) all edge direction which is approximately between $67,5^\circ$ and $112,5^\circ$ (red) is changed to 90° ; and 4) all edge direction which is approximately between $112,5^\circ$ dan $157,5^\circ$ (green) is changed to 135° .

- 5) Using the *non maximum suppression*/NMS formula. NMS is executed along the edges on the direction edge and eliminate pixels (pixel is turned to 0) which are not considered as edge, which produce a thin edge.
- 6) The final step is *hysteresis*, a process which eliminates broken lines to edge object. The procedure is to use two thresholds T_1 (lower threshold) and T_2 (upper threshold). All image pixel which have a value of more than T_1 are considered as edge pixel all pixels connected to these pixels and have a value of more than T_2 are also considered at edge pixels.

The floating hysteresis is done by having the two thresholds T_1 and T_2 threshold. The values of $< T_1$ will be changed to become 0 or become black, and the values of $> T_2$ will change to white (value of 255). For $T_1 \leq \text{value} < T_2$, temporarily will be valued 128, or grey values or unclear, will become 0 or 255. Then, a test will be made to obtain a condition as can be seen in Figure 2.24.

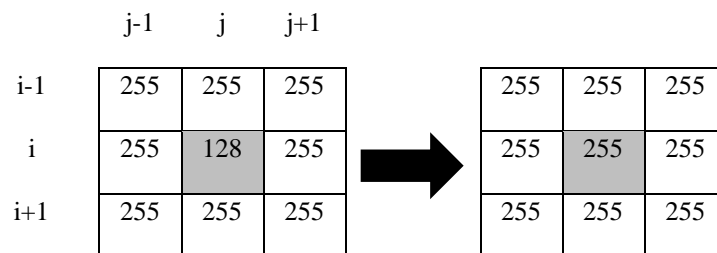


Figure 2.24 Test to change value 128 to 255

2.6.6. Laplacian Operation

The *Laplacian* operation is a 2nd series operation used to detect *zero-crossing* from image intensity and frequently producing a more accurate edge result [62]. This operation has flaws that is it is sensitive towards roaring sound, gives double thickness and is unable to detect edge direction [55]. The form of *Laplacian* operation is illustrated in Figure 2.25.

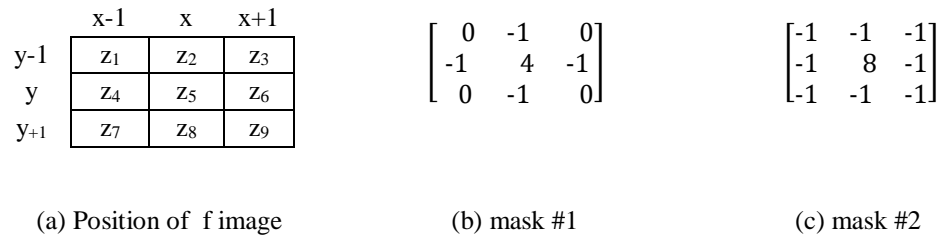


Figure 2.25 Laplacian Operation

Value Laplacian operator at (y, x) based mask #1 in Figure 2.25 (b), then the equation is defined Equation 2.10.

$$l(y, x) = 4 f(y, x) - [f(y-1, x) + f(y, x-1) + f(y, x+1) + f(y+1, x)] \quad (2.10)$$

2.6.7. Laplacian of Gaussian (LoG) Operation

The Laplacian of Gaussian (LoG) operation or Marr-Hildreth Operation was first proposed by Marr dan Hildreth in 1980 which is based on their research on human visual system [63]. This Operation is a 2nd order edge detection which is less sensitive to roaring sound. This is due to the use of Gaussian function which smoothen the image and has an effect on lessening the roaring sound to the image. The LoG operation is acquired through Convolution in Equation 2.11.

$$\text{LoG}(y, x) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (2.11)$$

The function equation 2.11 is known as Mexican/sombrero hat filter because it is shaped like a sombrero hat or Mexican hat, as in Figure 2.26 that is bigger the σ value, the bigger the mask needed.

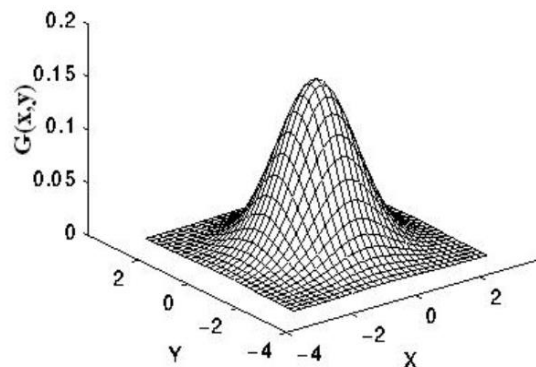


Figure 2.26 LoG(y, x) function as Mexican hat [63]

The 5x5 LoG operation is illustrated in Figure Figure 2.27.

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Figure 2.27 LoG Operation

The LoG method is combining *Gaussian* filter with *Laplacian filter* in edge detection. LoG edge detection uses 3 processing steps, that is filter, increase the picture object and detecting it [64].

2.6.8. Hough Transformation

Hough Transformation is a technique to determine the location of a shape in image [65]. Hough Transformation, which was first introduced by *P.V.C. Hough* (1962), was seen as its potential as one of the algorithm in processing image by *Rosenfeld* (1969), and then implemented to detect lines in images by *Duda* (1972), and since then has faced extensive development because of its supremacy and its big potential to further develop what is offered by this algorithm. One of the supremacies is that the Hough Transformation is able to provide just as good result with *template matching algorithm*, and much faster [66].

The classic Hough Transformation method is usually used to detect geometrical shapes which can be specified in the form of parametric such as lines, circle and ellipse. The working principle of the Hough Transformation method in detecting lines is by finding the most suitable geometric shape with a group of dots to the image. The detection method used in Hough Transformation has an advantage that is it can detect lines in the form of group of broken line edge. The Hough Transformation is also relatively unaffected by roaring sound.

There is an alternative approach in detecting lines in Hough Transformation [55]. Suppose a point (X_i, Y_i) on the xy coordinates and a straight line equation $Y_i = AX_i + b$. Unlimitedly produce many lines through (X_i, Y_i) , yet they are all fulfilled by the equation $Y_i = AX_i + b$ for various values of a and b . Then by writing the equation $b = -X_i + Y_i$ on ab coordinates (also called the parameter space) will

produce a single line to a point (X_i, Y_i) . Then, 2nd points (X_j, Y_j) also have a line at space parameter which is not parallel, but intersect at the point (a', b') , where 'a' is the slope and 'b' is the intersection of a line between the points (X_i, Y_i) and (X_j, Y_j) in the xy coordinate. So that in fact all the points on the line has a point on (a', b') as shown in Figure 2.28.

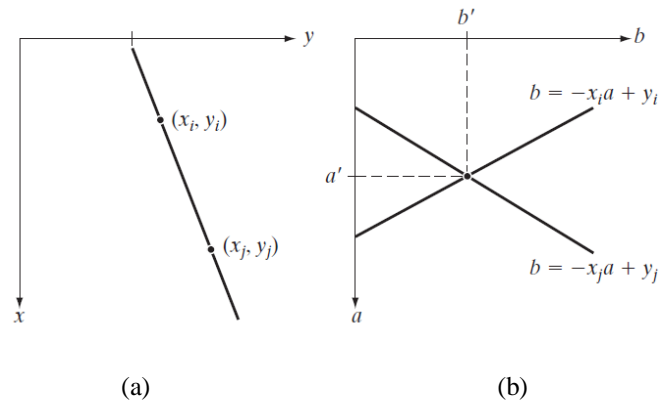


Figure 2.28 (a) xy coordinates (b) space parameter

In principle, the lines on the space parameter connecting to all points (X_k, Y_k) in the xy coordinates which can be drawn, and the lines on the xy coordinates could be connected by identifying the points in the space parameter which has a number of lines which have many intersections. In practice, this approach is difficult. And a point 'a' (slope on the line) nearing infinity on the line with vertical direction. One way to handle this difficulty is by using normal representation of line:

$$x \cos \theta + y \sin \theta \quad (2.12)$$

Figure 2.29 (a) illustrates the geometrical interpretation of parameters ρ and θ . A horizontal line has $\theta = 0^\circ$, with ρ similar to the intersection of x positive coordinate. Then the vertical line has $\theta = 90^\circ$, with ρ similar to the intersection of y positive coordinate or $\theta = -90^\circ$, with ρ the similar to the intersection of y negative coordinate. Every sinusoidal curve at Figure 2.29 (b) illustrates a family of the lines passing through certain points (X_k, Y_k) in xy coordinates. The intersection of point (ρ', θ') in Figure 2.29 (b) are connected to the lines passing between (X_i, Y_i) and (X_j, Y_j) as illustrated in Figure 2.29 (a).

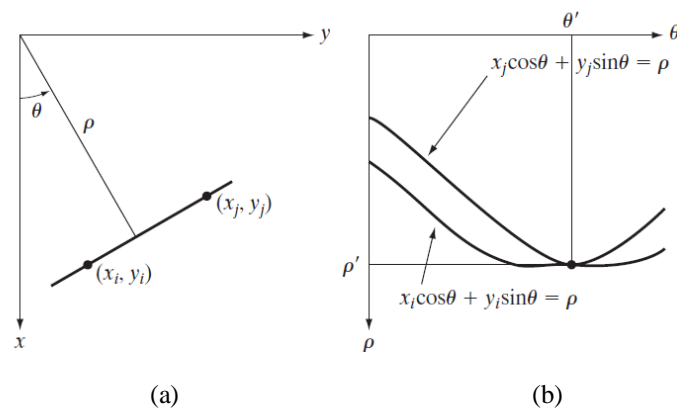


Figure 2.29 Illustration of geometrical interpretation from parameter ρ and θ

2.6.9 OpenCV for Image Processing

Specially designed software, such as Matlab and OpenCV (<http://opencv.org>), are being used for image processing. OpenCV (*Open Source Computer Vision*) [67] is an open source library which contains more than 500 algorithm optimised for pictures and video analysis. Since it was introduced 1999, most of it had been adopted as main development means for the society, researchers and developers in image processing and *computer vision*.

OpenCV was written with C and C++ languages and can be run in Linux, Windows and MacOS. OpenCV is designed for computation efficiency and with a strong focus on real-time application [68]. One of the objectives of OpenCV is to provide simple computer vision infrastructure which could be used to assist people who are in the building of highly sophisticated vision application.

The OpenCV has 3 main libraries used for image processing needs: 1) CV which are being used for image processing and vision algorithm; 2) Highgui needed for GUI, Image and Video I/O; 3) CXCORE for data structure, support XML and graphic functions. OpenCV [68] has various features which can be utilised in a study/research relating to computer vision, among others: 1) manipulate image data; 2) image and video I/O; 3) manipulate matrix, vector and linear algebra; 4) dynamic data structure; 5) processing fundamental image; 6) structure analysis; 7) for camera calibration; 8) motion analysis.

Various researches on *computer vision* and robotic using OpenCV have also continued to develop in line with the support of smartphone technology which has sophisticated processor and high camera resolution. The strongest and most prominent development of OpenCV presently is the implementation of OpenCV *library* for Android smartphone [69] which has provided a new platform for research in *computer vision* application to handphone. As an example on the usage of OpenCV is in the research on detection of traffic light for VIPs, OpenCV has been utilised as software to run its program. In this research, the status of the traffic light detected by *remote server* is communicated to the user through *text-to-speech* interface from the Android platform. The application of the traffic light detector will work on the *cloud component* using OpenCV implementation from algorithm to detect speeding object.

Moreover, OpenCV is also being used in research relating to the detection of an object and colour, and one of them by utilising OpenCV to detect edge of coloured pictures [70]. The testing method made in this research is comparing the result of the detection algorithm of edge coloured picture developed by the researcher with the result from the usage of *OpenCV Canny edge detector*.

CHAPTER 3

DEVELOPMENT AND IMPLEMENTATION OF SEES SYSTEM

3.1 Concept and Architecture on SEES

Smart Environment Explorer Stick (SEES) is designed to focus on fulfilling VIP's need for orientation and *space awareness* and provides route remote tracking assistance. One of the main innovations of SEES is internet connection, which makes it possible for VIP to get assistance and be monitored from any part of the world. Therefore, SEES is a smart cane which can be anywhere and at any time (*ubiquitous smart stick*).

The SEES system has 3 main components (Figure 3.1) namely *global remote server* (iSEE), *embedded local server* (SEE-phone) and *smart stick* (SEE-stick). iSEE is a global server which provides web services for VIP as a real-time remote guide and monitoring aid to assist the VIP.

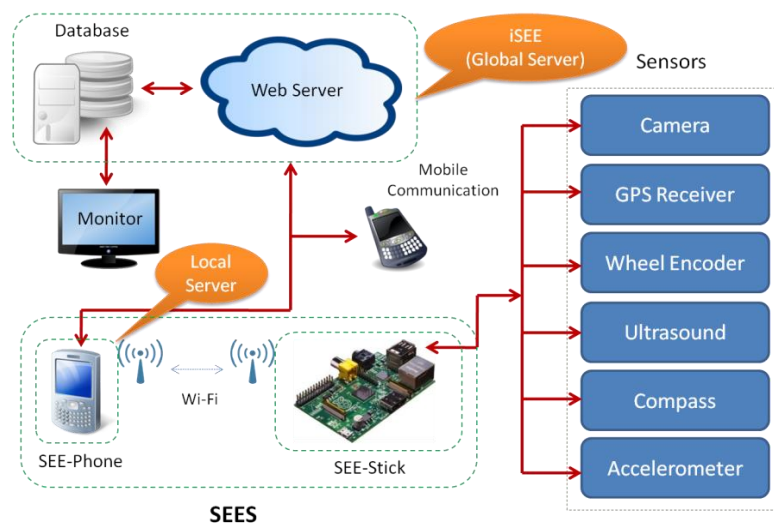


Figure 3.1 Concept of SEES system

SEE-phone is based on commercial smartphone used as local server which is implanted and provided local service with SEE-stick such as route vector and internet access to i-SEE. SEE-phone is the key tool for orientation and space awareness. SEE-phone communicates with GPS, via web server to map database with mobile equipment. SEE-phone is always connected to SEE-stick through Wi-

Fi. Special function on the SEE-phone is developed to fulfill VIP's requirement, among others to detect traffic light. Function that is provided in SEE-phone, such as introduction of voice, is used to improve practice interface technique for VIP.

The SEE-stick is a white cane with several active and passive sensors connected to the cane such as CMOS CCD camera, GPS receiver, wheel encoder, ultrasound, compass and accelerometer. These sensors are used to gather surrounding mobility signals and to implement high level mobility.

Perception and understanding on space by the last user could be improved when compared to classic tools because of point-wise input and audio feedback (from SEES system). The feedback is translated by using several sensors that are working paralelly when the data input are correctly integrated.

Using the SEES system, the VIP will be able to locate the current location. Besides, family and friends are able to monitor remotely the VIP's journey. In the near future, a new function will be developed to detect the location of traffic lights, name of roads/buildings, bus/railway station, etc.

As a note *IPv6 over Low Power Personal Area Network* (6LoWPAN) and RPL protocol routing were adopted to be applied to SEE-stick. As such, in accordance with this context, SEE-stick is able to be connected *peer to peer* to iSEE and future transportation system [71]. Therefore, the SEES can be considered as an implementation on *Intelligent Transportation System* (ITS) concept. This means that in the near future, SEES can be used by the VIP to access urban transportation system such car, bus and train.

SEES has independent mobility characteristics which are follows:

- a) VIP will always be able to know his/her location,
- b) Other people are able to remotely monitor the VIP journey and provide hints if necessary;
- c) In the future, SEES can be used by VIP to access urban transportation system (such as car, bus and train) and GIS (geographical information system and detailed and local map).

Mobility signs gathered by SEES sensors which were transformed into high level knowledge (forwarded to VIP through voice or tactile feedback) to:

- a) Acquire accurate location data (which will make possible for the user to track and check whether the VIP is on the right track);
- b) Acquire correct information on condition of the surrounding such as obstacles, the status of the traffic light and the status of the surface of the road,
- c) Quickly sent a warning message to VIP when an orientation error occurs.

All inputs are sent to VIP as a product of multi modal data fusion passed through sound or tactile stimulation channel.

3.2 SEES for VIP's Mobility Function

The objective of SEES is to assist several functions such as walking, orientation and space awareness. For the first SEES prototype, the space awareness function is directed towards detection of traffic light, the walking function is directed towards detecting obstacles, estimation of journey distance, and estimation on the roughness of the surface and the orientation function is directed towards direction to the estimation of location directed and the current estimation of location of the last user.

Its function is targeted to have a direct impact on the selected sensor. The status of the traffic light is the function of space awareness detected by using smartphone camera from SEE-phone. Figure 3.2 is the example of the status and model of traffic light on the road.

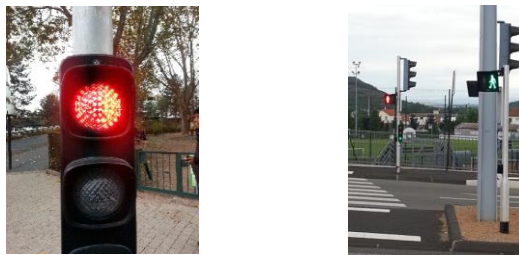


Figure 3.2 Traffic light model

The walking function is supported by a combination of ultrasonic, camera, wheel encoder, *accelerometer* and digital *compass*. The ultrasonic sensor used to detect obstacles in front of the VIP such as trees and walls up to a distance of 2m to the front. The wheel encoder is installed on the SEE-stick wheel, which is then used to estimate the distance of the VIP's journey from the starting point of the journey. The distance is estimated by combining data from wheel encoder, route vector and GPS. Distance traveled periodically is sent to the local (SEE-phone) and global (iSEE) servers for a higher level of mobility function. The compass sensor is used to detect the VIP's travel direction. The data from the compass is fused with GPS data and wheel encoder data so as to increase the accuracy of the VIP's location and the estimation of the distance.

The SEE-stick Camera is used to detect and to recognize urban geometry and organization from space (such as road intersection, sectional map and position of the traffic light.). In its relation to the camera information, the data on *accelerometer* will be used to estimate the roughness of the surface of the path so that VIP is able to adapt the texture of the path. Figure 3.3 is an example on the surface of the path especially for VIP.



Figure 3.3 Surface of the path specially made for VIP

The orientation function uses two main SEE-phone sensors: camera and GPS (including Google map). GPS is used to determine the current location of the VIP, whereas Google map will make it possible for the VIP to plan the journey to reach a special target. GPS data will be periodically sent to database server to update the VIP location for real-time remote monitoring. Table 3.1 summarises different sensors and mobility function supported by SEES system.

Table 3.1 Type dan function of sensors in SEES

Sensors	SEES System		Use of Sensor
	SEE-stick	SEE-phone	
Ultrasonic	V		Detection of obstacles
Wheel Encoder	V		Distance of Journey
Accelerometer	V		Roughness of surface
Compass	V	V	Direction
Camera	V	V	Obstacle/status of object/geometrical space
GPS	V	V	Location/position

The output from sensor is an input for SEES layer which are turned into high level knowledge useful to mobility. This knowledge will directly convey to VIP (as feedback) Figure 3.4 illustrates sensory interconnection model in the SEES system.

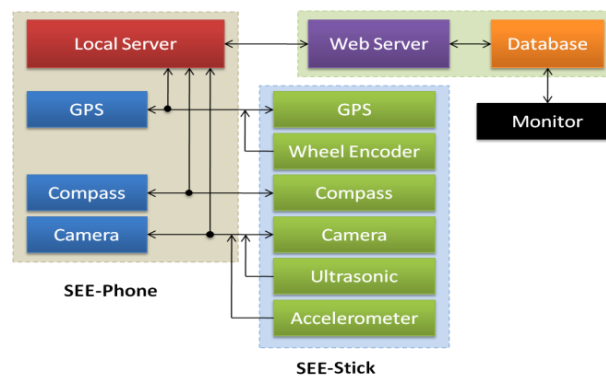


Figure 3.4 Sensory Connection Model to the SEES system

3.3 SEES Activation Model

SEES is a physical level modular system. Sending of data between SEE-phone and SEE-stick is made through *Wi-Fi*. Table 3.2 indicates all details on SEES mode. VIP can choose any mode when travelling to a destination.

Table 3.2 SEES Mode

Mode	SEES System			Description
	SEE-stick	SEE-phone	iSEE	
Mode 0	V			Basic Mode
Mode 1		V	V	Telephone Mode
Mode 2	V	V		Local Mode
Mode 3	V	V	V	Complete Mode

SEES interface technique to VIP offers 4 assistive levels:

1. Mode 0 (always active, minimum SEES configuration, only SEE-active). This mode makes it possible for VIP to walk without handphone. This can happen when VIP forgets to carry his/her handphone or the battery of the handphone runs out. Journey is assisted by the cane sensor by sending out data (GPS, ultrasonic, camera, encoder, and accelerometer and compass sensor). SEES system turns into SEE-stick. The main disadvantage of this mode is that VIP is unable to receive confirmation/information or call his/her friends/family if there is something wrong during the journey. As a note, SEE-stick tries to connect to iSEE through Wi-Fi access points if needed.
2. Mode 1 or telephone mode (SEES works by using SEE-phone and iSEE). This mode makes it possible for user to keep on moving without smart cane. This navigation is made by using handphone sensor (GPS, camera and compass) and iSEE server. The disadvantage of this mode is that VIP is unable to detect whether there is an obstacle or not or the coarseness of the surface.
3. Mode 2 or local connection mode. In this mode, SEES works well between subsections (SEE-phone and SEE-stick). SEE-stick will operate with iSEE server and SEE-phone will only be used as a router (Wi-Fi access point) to resolve mobility difficulty if it arises. VIP is able to get remote assistance from people monitoring the server. In this mode, SEES will give specific guide.
4. Mode 3 or complete/full mode. In this mode, SEES works with all its subsections (SEE-phone, SEE-stick and iSEE). SEE-stick operates with SEE-phone and iSEE to resolve mobility difficulty when it occurs. VIP is able to get remote assistance (from relatives using telephone (mobile communication) or other people monitoring the server). In this mode, SEES gives specific guiding system.

3.4 SEE-Stick Concept and Architecture

3.4.1. SEE-Stick Architecture

The architecture of the SEE-stick system consists of 3 layers as illustrated in Figure 3.5 in the form of user's interface layer, logical layer and sensory layer [72].

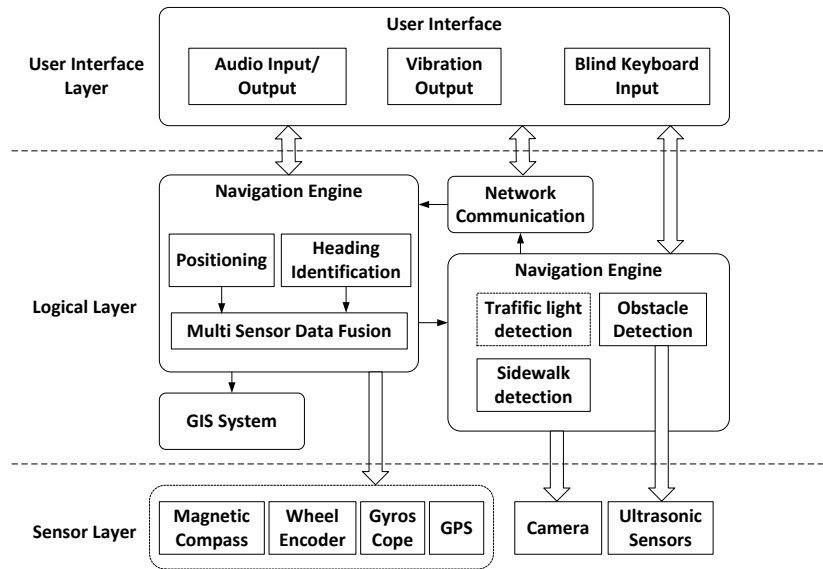


Figure 3.5 SEE-stick system Architecture

The user interface layer is responsible towards interaction with VIP. VIP will specially command the system through voice or tactile keyboard (*braille*) for example starting navigation and requesting for position information. On the other-hand VIP receives a feedback from the system through audio or vibration tool for example information on position, information on obstacle.

Logical layer is the main layer from the SEE-stick architecture which consists of 4 modules namely navigation engine, surrounding detector, communication network and *Geographic Information Systems* (GIS) modules. The navigation engine provides relatively accurate position as well directs information through combination of several data sensors. Surrounding detector module is responsible for identifying obstacle, sidewalks and traffic lights. Obstacle detector is based on several ultrasonic sensors installed at the bottom and the middle of the cane to cover the surface scope of the ground from the head height to the entire width of the front body VIP. The communication network module receives VIP position from the navigation engine and sent it to long distance server. So the VIP's track can be seen by VIP's family and friends. The surrounding detector module uses the communication network module to communicate with SEE-phone via Wi-Fi. For example SEE-stick requests the SEE-phone to identify the traffic light status. The sensory layer can be divided into 3 parts, navigation sensor, obstacle detector with ultrasonic sensor, and sensor containing surrounding or camera. Navigation sensor

includes GPS, wheel encoder, gyroscope and digital compass. Data are gathered from navigation sensor, which will be integrated with navigation engine at the logical layer.

3.4.2. Multi-sensor Data Fusion for Navigation

The latest VIP's navigation system integrates with traditional *Inertial Navigation System/Global Positioning System* (INS/GPS) so as to increase the accuracy of the estimated position of the VIP. However, research shows that *Reduced Inertial Sensor System* (RISS/GPS) integrated for inland navigation has more advantage than INS/GPS system. RISS uses wheel encoder or odometer instead of accelerometer to calculate speed. Speed calculation using accelerometer involves mathematical integration operation while the wheel encoder does not use this operation. The problem with mathematical integration operation is refractive error which can cause unlimited errors developed within time.

In view of the advantages of RISS/GPS integration, the SEES-stick uses GPS receiver, gyroscope, wheel encoder and magnetic compass to calculate 2D position, speed and attitude. For the SEE-stick prototype, the data from this sensor are integrated with *Linearized Kalman Filter/LKF* [73] as shown Figure 3.6.

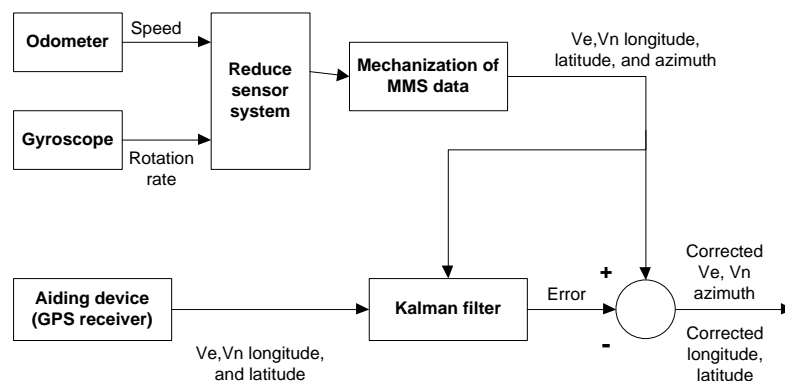


Figure 3.6 Integrasi Linearized Kalman filter berbasis RISS/GPS

Gyroscope that is parallel to the vertical axis of the VIP's body is used to acquire azimuth angle in the north-east horizontal direction. With the assumption that the VIP remains at the horizontal direction on most part of the time, the VIP's speed originates from the encoder wheel used together with the information instruction acquired from the gyroscope to determine the speed along the east and

north where longitude and latitude VIP that has been determined. The position, speed and direction error are estimated by the Kalman filter by relying on dynamic error model, GPS position and latest speed.

The accuracy short term information is provided by RISS used to resolve GPS outages and multi-path error. If the GPS went off in a long period of time, the SEES-stick uses the digital compass as a source of additional measurements. Magnetic sensor determines azimuth by sensing the earth's magnetic field component. Unfortunately, local disturbances in the field caused by the permanent magnet nearby, electric current, or a large body of iron that could significantly affect the azimuth origin.

The *Ladetto Method* [74] takes into consideration gyroscope signal as *azimuth* input and using magnetic compass when the GPS is off and the undetection of magnetic disturbances renews the gyroscope parameter. The method to detect magnetic disturbance using gyroscope is explained in it.

3.4.3 Definition of Navigation Process

During the navigation process, the SEE-stick has to track down VIP's coordinate (position), speed and azimuth (or post). Navigation process can be defined as engine condition as illustrated in Figure 3.7

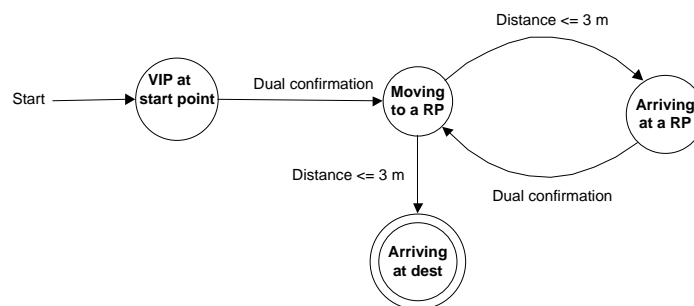


Figure 3.7 Engine condition during navigation process

There are 4 conditions: 1) VIP at the starting point; 2) VIP moving towards route check point (RPs); 3) VIP arrives at the route point; 4) VIP arrives at the destination. Re-confirmation on the engine condition means that between the VIP with his/her behavior and SEES system has confirmed the transition to the

following condition. The distance on the engine condition is the distance (for example 3 meter) between VIP and route point.

3.4.4 Definition of Navigation Route

The navigation route has to be especially adjusted for VIP. If VIP walks along the sidewalk or crosses outside his/her surrounding, his/her navigation route should include sidewalk and cross walks. In the SEE-stick system, the following definition is used: vector navigation route is a special route designed and dedicated for VIP from the starting point to the destination. This route consists of several consecutive sidewalk or crosswalk referred to in sub-route system. Each sub-route is distinguished two check-points which are transition points between 2 sub-routes. Example of vector navigation is illustrated in Figure 3.8.

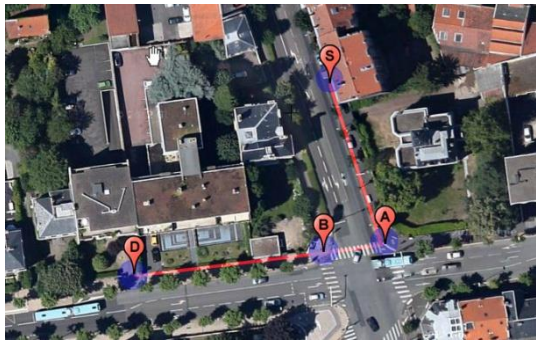


Figure 3.8 Vector Navigation Route

Vector route (S, A, B, D) in Figure 3.8 is marked with red lines. Letters 'S', 'A', 'B', 'D' represent 4 check-points, where S is the starting point and D is the destination. There are 3 sub-routes: S, A is the sidewalk, A, B is the crossing road and B, D is the sidewalk.

3.5 SEE-phone Concept and Architecture

SEE-phone is a smart phone which has two main tasks namely orientation and space awareness for the actual independent mobility. The SEE-phone is the key tool for orientation. It communicates with GPS through web server so as to access mapping database and communication with other mobile tools.

3.5.1. SEE-phone for Orientation Function

As orientation function, the SEE-phone is used to give direction of the destination location, define new route or guiding the VIP from the route which has been earlier (*route remote tracking/RRT*). To execute the navigation function, the SEE-phone uses GPS from the smart phone.

In general, the GPS in the smartphone uses *World Geodetic System 1984* (WGS84) as reference system. Presently, the WGS84 coordinate used by most GPS tools output standard [75]. WGS84 considers the earth as a spheroid object, determined based on satellite observer in the earth orbit. The main WGS84 axis is 6,378,137.0 meter and minor axis is 6,356,752.3 meters. The tracking of the VIP during his journey can be decomposed into basic operations: 1) detecting the present VIP's position; 2) tracking VIP's real-time position with GPS and Google map.

The execution on these tasks has the following objectives:

1. Developing the application to Android smart phone to detect the VIP's location.
2. Developing the application to track the VIP's real-time position with GPS.
3. Connecting the application which had been developed to the server in another place (campus).
4. Monitoring VIP's position from the view of the map to the server in the computer/notebook monitor.
5. Display VIP's position by the server.

Figure 3.9 illustrates access model from the GPS receiver (in the smart phone) to the *database server*. The data on the present VIP's location is acquired from the GPS in the smartphone and sent it to the web server which is kept in the database server. Finally, the data is displayed to the monitor in the form lines.

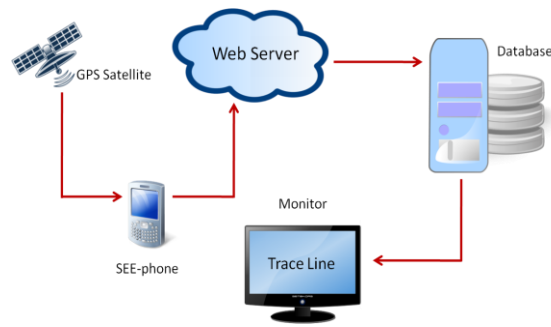


Figure 3.9 Access Model from GPS receiver

3.5.2. SEE-phone for Space awareness function

One of the functions of SEE-phone is to detect poles and the status of traffic light. There is a need for the following basic operation in detecting the status of the traffic light by using SEE-phone: 1) locate traffic light, 2) recognize the status of the traffic light (colour), and 3) gives feedback on the status of the traffic light to VIP.

Figure 3.10 illustrates the process in the implementation of SEE-phone camera on the smart phone for detecting the traffic light through 2 detection models, by the position of the traffic light detected by camera sensor at the cane (SEE-stick) and the position of of the traffic light based on GIS data in the database server, taking the photo data from the traffic light status. Then, the android smartphone processes the photo and finally produce audio output to inform the VIP.



Figure 3.10 SEE-Phone in detecting traffic light

The whole process needs 4 steps as illustrated in Figure 3.11: acquisition of image; preprocess; filtering and *thresholding*.

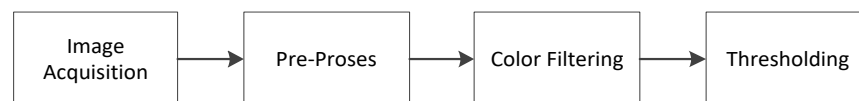


Figure 3.11 Image Processing in android

First, the photo of the traffic light is acquired by the smartphone camera. Preprocessing step is obtained certain parts from the colour of the photo by using photo editing application based on Open Source GNU *Image Manipulation Program* (GIMP). In colour filtering stage, the colour of the photo consisting of RGB (*Red, Green and Blue*) model is then changed to HSV (*Hue, Saturation and Value*) model. The final process is the *thresholding* that takes the pictures and restore the binary image (where red, yellow or green to white and the rest black). In the HSV level, each colour is given certain number (*hue*), the number of colours are given other number (*saturation*) and the brightness of the colour is also given another number (*value*).

3.5.3. User's interface on SEE-phone

Navigation guide used in verbal description and instruction is considered as an efficient way to convey data to the VIP. Two applications in the smartphone will be used, TTS dan STT [76]. *TTS mobile (Text-to-Speech)* is an application to read message on the screen of the smartphone and interpret it into the audio form for the VIP [77]. Figure 3.12 illustrates the block diagram of TTS system.

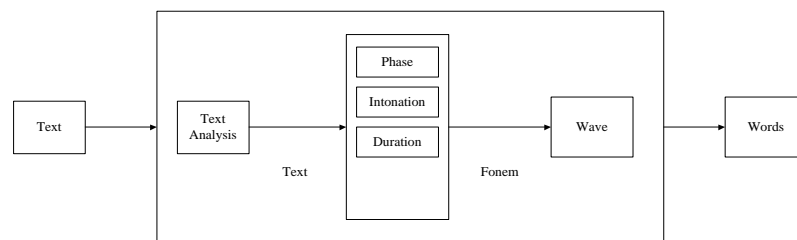


Figure 3.12 TTS System

The recognition of voice application (STT, *Speech-to-Text*) makes it possible for the engine to receive and interpret the wordings, or to understand and perform a command which are spoken [76]. In practice, the size in the recognition of voice is effective vocabulary which is directly connected to random memory access capacity of the computer where they were installed. In the SEE-phone, instruction

input or command given in voice, so as to operate certain application the VIP would talk with his/her voice (STA, *Speech-To-Action*).

3.6 Implementation and Test on RRT System

In the SEE-phone, the *Route Remote Tracking/RRT* is a sub-section of the SEES system used to guide the user (VIP) in travelling on certain route and at the same time, monitoring the position of the user (VIP). In this research, tracking the VIP's route can be explained in following basic operation that is detecting the current position of the VIP and real-time tracking VIP's position by using GPS and Google Map technology.

The objective of this experiment are 1) To build an android based application to detect the location of theVIP; 2) This application can detect and locate real-time VIP's position with GPS; 3) This application can be connected to a server in specific location, 4) anyone can monitor the position and location of the VIP from the map displayed on the monitor screen, and 5) the server is able to display data from the VIP's position.

Figure 3.13 illustrates communication model from GPS receiver (in the smartphone) to the server database. The current VIP's location is acquired from the smartphone and transmitted to webserver to be saved in the database server. Finally, the data can be displayed on the monitor as track route.

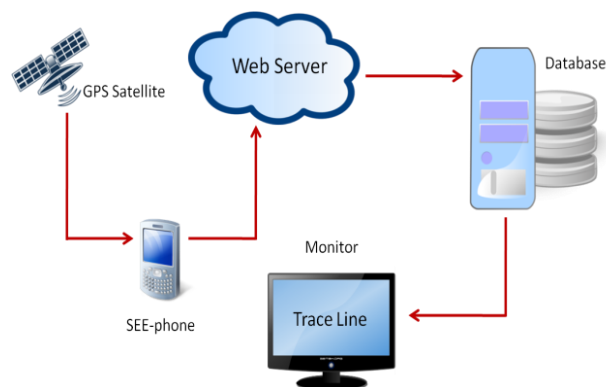


Figure 3.13 GPS Receiver Communication Model to the database server

In general, this RRT application has been built through 3 stages (Figure 3.14).

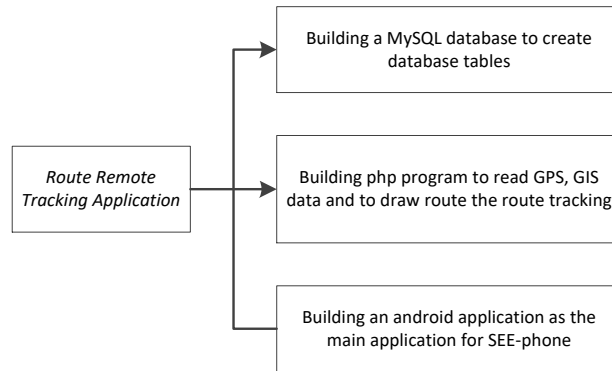


Figure 3.14 The stages in building the RRT application

The first stage in preparing the *remote route tracking* application is by building MySQL database which will be used to organize, save and extract data. MySQL [78] is a kind of database server which is very popular (very quick, reliable and easily used) as well as using SQL (*Structured Query Language*) language in accessing its database. To perform an easier administration on MySQL, phpMyAdmin can be used as *webbase* control panel for MySQL which has been installed into the computer. With phpMyAdmin, database and database table can be made, modified and deleted from MySQL. In this study, the database table is used to save data/information from the GPS receiver on the handphone (SEE-phone) and to save data/information from certain location considered as *Point of Interest* (PoI).

In the second stage, the PHP (*Hypertext Preprocessor*) application program is prepared so as to read GPS, GIS data and drawing tracking routes. PHP [79] is a popular *scripting* language because it can be used on various operation system, very simple database connection and free of charge. PHP working concept is preceded by a request of a web page (file.php) by the browser or client. Then, based on its URL address, the browser is able to acquire address from the web server, which identifies the page requested, and provides all the information required by the web server. When the web server obtain the PHP file requested, the content is transmitted to the PHP engine so as to be processed and providing the web server its result in the form of HTML code and transmitted them to the client. In this research, the PHP application is used to track and display the route passed by the VIP on the website.

Stage three in preparing RRT application is building the Android application so that it could perform its main tasks: 1) to guide the VIP on his trip to a new route (known as *new track mode*), 2) to guide the VIP to follow routes saved in the server database (known as *follow track mode*) and 3) to make an application using manual and voice activation to run RRT application.

In this research, several activities utilizing GPS receiver are also made in the Android application. As explained above, besides building Android basis program in the RRT application, PHP based program is also built and is placed in the webserver. The task of this program is to regulate communication between *Android smartphone* and *database* inside the webserver.

The test on the RRT detection system is as follows: Figure 3.15 illustrates the display of the map from the RRT Android application program where (a) Android emulator and (b) smartphone Android have been tested. In this test, we had used Android Emulator Google APIs, Platform 4.1.2, Level 16 and Smartphone Samsung SIII Model GT-I9300, Android Version 4.1.1.

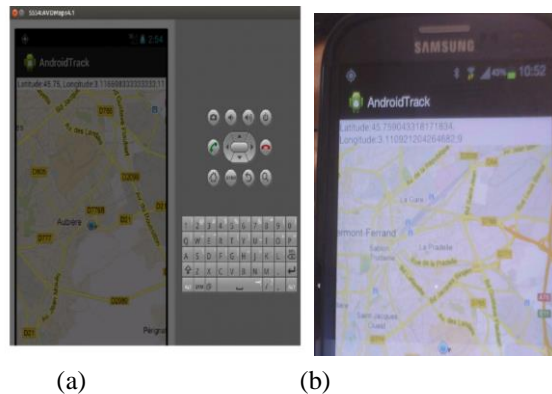


Figure 3.15 Display of the map to Android Emulator and Smartphone

Figure 3.16 illustrates map; from route example for this test. The user with his smartphone (SEE-phone) walks from point G (*start point*) to point D (*end point*). When the user starts walking the Android application will be activated and the GPS receiver will be “on” position. The GPS receiver in the *smartphone* will receive data from related GPS satellite with the actual user’s position and the

smartphone will transmit this data/information to the database server via Wi-Fi network or public network.



Figure 3.16 Example of the user's route

The location data from SEE-phone are gathered by *database server* as displayed in Table 3.3. In this test, there are 7 different points of the user's position received by the server.

Table 3.3 Data on the position of the user in the database server

Host: localhost			
Database: Android			
Generated by: phpMyAdmin 3.5.2.2/MySQL 5.5.27			
No	Latitude	Longitude	Time
1	45.75892	3.11157	17:25:16
2	45.75858	3.11185	17:25:56
3	45.75850	3.11197	17:26:10
4	45.75846	3.11212	17:26:16
5	45.75846	3.11231	17:26:26
6	45.75847	3.11248	17:26:36
7	45.75850	3.11262	17:26:49

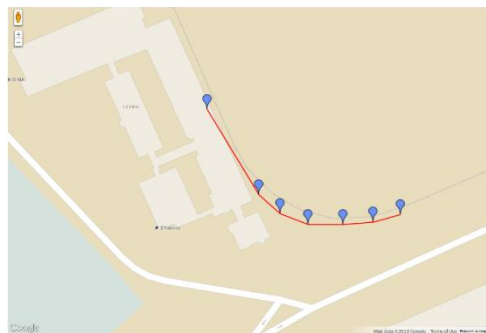


Figure 3.17 Display on the monitoring of route passed by the user

Figure 3.17 is the result of the tracking of the user's route displayed by the monitor. The display of this route can be accessed by everyone through internet network, and therefore, the latest position of the user can be seen. The red line shows the route passed by the user, and the existing dots is the representation of GPS data received by the server.

3.7 Implementation of SEES System

In this research the writer will especially discuss one part of the SEE system that is SEE-phone. In general, the SEE-phone system is built in 3 main applications: 1) pole detection application, 2) traffic light detection application 3) route remote tracking application. The architecture figure of the SEE-Phone system is illustrated in Figure 3.18:

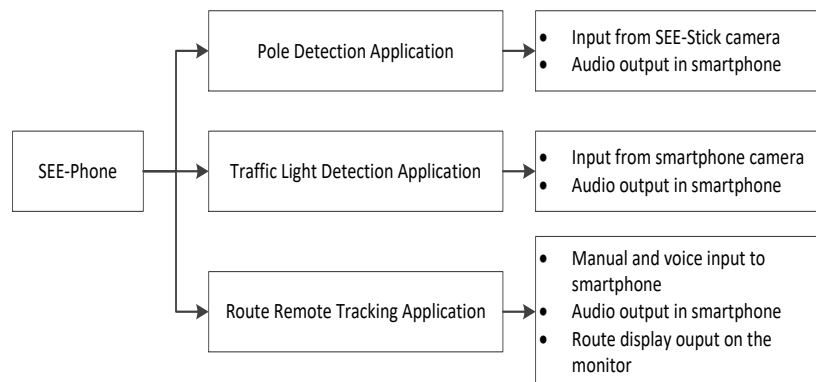


Figure 3.18 SEE-phone architecture

Each section (sub-system) has its own function but they are inter connected to one another. The pole detection application is used to detect the whereabouts of the pole. This study will focus on how to detect the position of traffic light on the road. Camera mounted on the SEE-stick is used to capture the the pole image and then sent photo data to *smartphone* so as to trigger the application on detection of traffic light. The chronology in operation principle of pole and traffic light detections is illustrated in the flowchart in the following Figure 3.19.

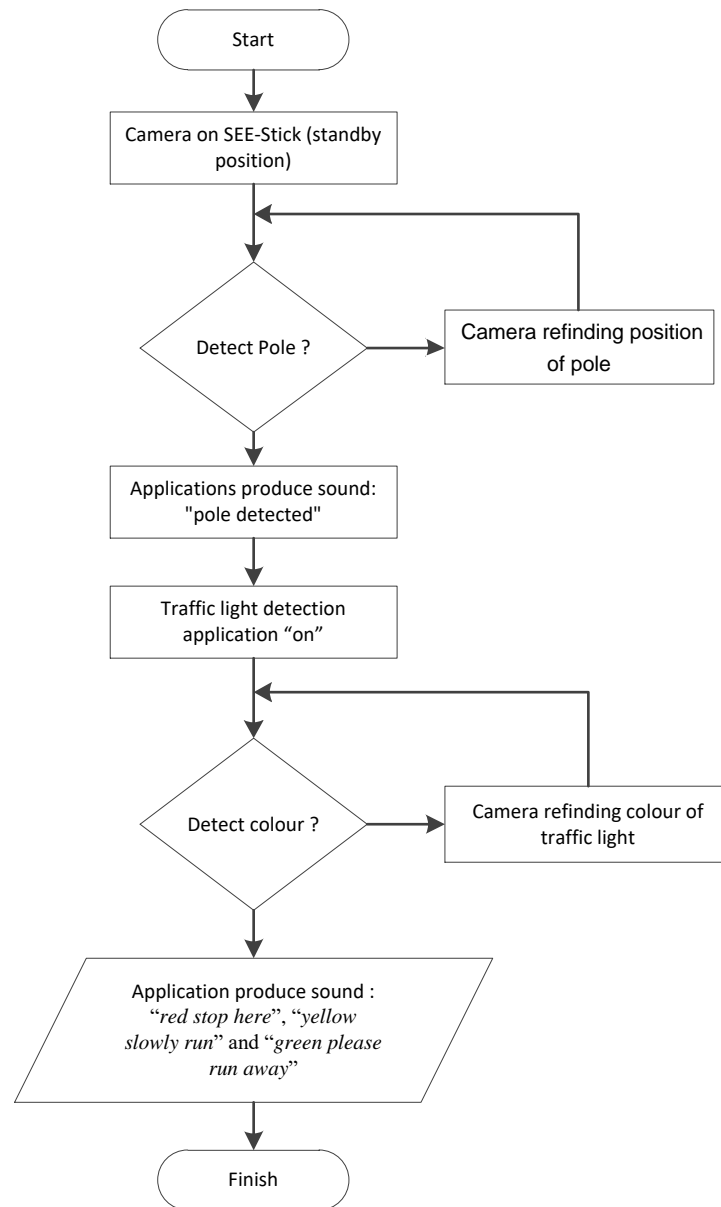


Figure 3.19 Flowchart of detection on pole and traffic light

Figure 3.19 describes that in active condition, SEE-Stick will always be actively scanning input image in detecting poles wherever the algorithm is implanted. When the pole is detected by the SEE-Stick, it will be transmitted to the smartphone, and this will produce a voice '*pole detected*' to inform the VIP. From then on, smartphone will have a dominant role where the VIP will use it to scan with the camera to detect the colour of the traffic light. The product of the colour detection of the traffic light will be in the form of voice.

3.8 Data Communication model in SEES system

As a whole, the SEES System consists of 2 main components (sub-system) that is SEE-stick and SEE-phone. In this paper, the writer will especially discuss about communication model between SEE-stick and SEE-phone.

In the pole detection process, SEE-Phone will be conducting a synergy work with SEE-stick. The camera on the SEE-stick captures photos, and then transmitted the photo data to SEE-phone (through WI-FI) to detect the position of the pole. The hardware in SEE-stick, built using Raspberry Pi board (<http://raspberrypi.org>) a single-board computer card [80]. In this research 2 USB terminals are used, one for camera (webcam) and the other wireless USB Adapter, as illustrated in Figure 3.20 below:



Figure 3.20 Hardware Raspberry Pi 2 model B

Raspberry Pi 2 Model B [81] has 1 gigabyte (GB) Random Access Memory (RAM), a 900MHz quad-core ARM processor, four universal serial bus (USB) interface, an *Ethernet port*, a mini USB for power supply and *High-Definition Multimedia Interface (HDMI)* for display. The Operation System (OS) is flashed into a *micro Secure Digital (SD)* card. This tool can be operated using a USB *keyboard, mouse and display* or through LAN port by making remote *Secure Shell (SSH)*.

The data communication model between SEE-stick and SEE-phone is built using *Wi-Fi* technology. If the pole is detected, then *Raspberry Pi* operating image processing to detect pole will transmit data using *wireless* through *data socket transmission* to *Android smartphone* which will activate the traffic light detection program. The operation process of data communication model between SEE-stick and SEE-phone is illustrated on the pseudo code of Figure 3.21.

```

Algorithm listenServer
  Input: portno, port number server
  Initialization socket address for server and client
  struct sockaddr_in serv_addr, cli_addr;
  Initialization new socket standart ipv4 with mode stream
  sockfd = socket(AF_INET, SOCK_STREAM, 0);
  if sockfd < 0
    error("ERROR opening socket");
  bind socket to server address
  if bind(sockfd, serv_addr)
    error("ERROR on binding");
  looping as long as there is a connection with SEE-stick
  while listen(sockfd,5)
    clilen = sizeof(cli_addr);
    read input string from SEE-stick
    newsockfd = accept(sockfd, cli_addr, clilen);
    text = newsockfd.read();
    if strcmp(text,"pole")
      voice("pole is detected");
      startActivity("trafficLightActivity");

```

Figure 3.21 Pseudo code of communication model between SEE-stick and SEE-phone

On Figure 3.21, acting as server is SEE-phone which will receive transmission from SEE-stick (client). Communication made will be using TCP/IP protocol so that so it has to configure port server number. The data in the main structure used in the communication is socket, which is an object that gathers data that have transmitted. The communication between SEE-stick and SEE-phone will always be connected since both tools are switched on.

The *Socket* is a means to transmit data from one application to another in one computer or different computer which have one network. In this case, there is one network between *Raspberry Pi* and *Android Smartphone* which is facilitated by a wireless router. In programming the socket, there is a port number. In this experiment we use port 6000. After initializing the port, we also need an *IP Address* from the destination application so that the data could be transmitted.

After the socket has been made, the next step is to build communication data, where on the Android part the socket is programmed in listening condition. When the communication is not built, then a message “*ERROR connecting*” will appear. After the communication between *Raspberry Pi* and *Android* is built, the data can then be transmitted. In this case the data sent is the form of string “*pole*”.

3.9 Implementation and Testing of Traffic Light Detection System

In this research, the traffic light detection system is designed to work in Android operation system environment. At the beginning this program operates with a trigger from SEE-stick which transmits text that the pole is detected through a network with *TCP/IP* protocol to *Android (smartphone)* device. Next there is another application in the smartphone which has a task of recognizing traffic light colours. Pseudo code principle on the operation of traffic light detection is illustrated in the following Figure 3.22:

```

Algorithm initProgram
    const int SERVERPORT=6000;
    create UI
    createUI ();
    int status = createTextToSpeech ();
    if status < 0
        printf("Text to Speech Initialization Failed");
    produce voice
    voice("Welcome to traffic light detection");
    setupTextView ();
    Handler handle = new Handler(Context);
    create the thread listenServer to receive communication
    Thread t=new Thread(listenServer,SERVERPORT);
    t.start ();

```

Figure 3.22 Pseudo code on the traffic light detection

The algorithm to the traffic light detection is implanted in SEE-phone. Therefore, the *Text to Speech* library translates the text into voice. Other than that, the initial display is initialized on this algorithm. This Android application program uses a class that is derived from the activity and implement an interface *OnInitListener*. Because the *activity* interact with the user, then the activity is usually almost always open the window so that the user can relate. Then the *OnInitListener* is an *interface* for *TextToSpeech (TTS)* class. By using the *onInit method* we are able to see whether we succeed or not in producing the *TTS* object.

In this research, we use the C++ language, which utilize OpenCV library, to detect the status of the traffic light (whether it is on or off). The program made using C++ language is then included in the Android program

with *Java Native Interface* (JNI) technique [82]. JNI is a framework programming which makes it possible for the java code in *Java Virtual Machine* (JVM) to be summoned by native application (special programs to the hardware and operating system) and library written in languages other than C, C++ and assembly.

To detect the colour of the traffic light a function in the OpenCV *CvInranges* is used, that is, whether each pixel in the photo enters into the given range in the parameter. If the pixel in the photo enters into the range, then the pixel to the coordinate of the photol will then be set at *0xff*, on the other hand, if it did not enter the range, it will be set at *0x00*. For example:

Red

Bottom limit in BGR (176, 219, 175)

Top limit in BGR (179, 237, 215)

The use of OpenCV function, the result is (Figure 3.23):

```
cvInRangeS(hsv_frame, cvScalar(176, 219, 175, 0), cvScalar(179, 237, 215, 0), thresholded);
```

Figure 3.23 Function of cvInRangeS

The *hsv_frame* is the original image and threshold is the destination image color detection results. Table 3.4 is the criteria for each colour:

Table 3.4 Criteria in RGB value

No	Colour	Bottom limit of RGB	Top limit of RGB
1	Red	175, 219, 176	215, 237, 179
2	Yellow	171, 154, 20	193, 173, 24
3	Green	79, 218, 74	142, 255, 80

The testing on this part is made through several steps:

- 1) Takes several samples of colour photos/pictures from the traffic light status.
- 2) Convert photos/pictures into colour data by using image processing.
- 3) Make colour data as reference value.

- 4) Compares colour reference with colour of the photo/picture from the traffic light status detected by SEE phone.

In the initial process, the colour of the photo/picture from the traffic light is taken from the smartphone. In this research, we use a camera from Smartphone Samsung SIII Model GT-I9300 to take colour image (traffic light status). This process starts with conversion of RGB colour into HSV (*Hue, Saturation and Value*) image. The final process is *thresholding function*. This function will take photos/pictures and return the binary image (were red, yellow and green will turn into white and the rest into black). The result of the *thresholding process* is illustrated in the following Figure 3.24:



Figure 3.24 Thresholding Process

A check will be made on the h, s, and v values to find out whether they are bigger than the previous values. If they are bigger, they will be placed in the *maxh*, *maxs*, and *maxv* variables. The values are also checked whether they are smaller than the previous values. If they are then they will be placed in the *minh*, *mins*, and *minv* values. Table 3.5 shows the data from the HSV level tested for the 3 colours (red, yellow and green) in this experiment.

Table 3.5 Example of data from the HSV level

Colour of Picture	HSV min	HSV max
Red1	175, 236,169	179, 255, 255
Red2	175, 234,201	179, 255, 255
Red3	175, 242,242	177, 255, 255
Red4	175, 240,246	177, 255, 255

Mostly Red	175, 238, 215	178, 255, 255
Yellow1	29, 240, 242	32, 255, 255
Yellow2	26, 245, 243	32, 255, 255
Yellow3	27, 242, 244	32, 255, 255
Yellow4	30, 247, 252	32, 255, 255
Mostly Yellow	28, 244, 245	32, 255, 255
Green1	89, 244, 238	91, 255, 255
Green2	89, 228, 239	91, 255, 255
Green3	88, 233, 229	90, 255, 255
Green4	90, 218, 249	91, 255, 255
Mostly Green	89, 231, 239	91, 255, 255

Average value of HSV is a reference data for the photos taken by SEE-phone camera. When the value of the colour (traffic light) caught by SEE-phone camera that are within reference data range, the programme will summon the voice application. In this study, the voice function uses TTS concept that is converting colour detection into voice, that is “*red stop here*”, “*yellow slowly run*” and “*green please run away*”. The result of the simulation test in the traffic light detection is illustrated in Figure 3.25 where the result of the colours is displayed.

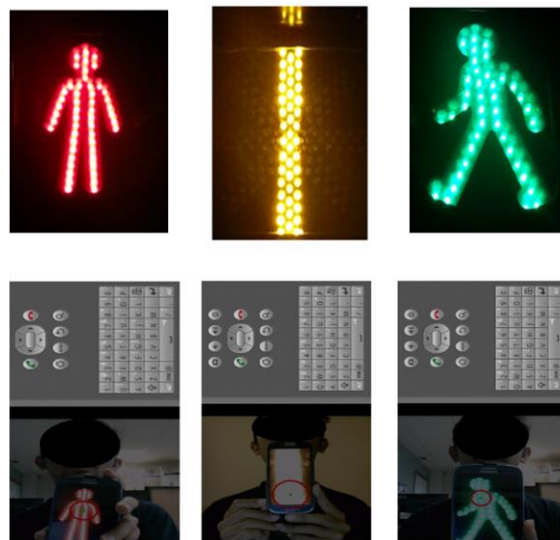


Figure 3.25 Simulation Test on Traffic Light Detection

CHAPTER 4

DEVELOPMENT OF THE NEW POLE DETECTION ALGORITHM

4.1 Research Related to Object Detection

Related studies on detection of object had been made by several researchers, but none of the researches which especially discussed on pole detection. In 2006, there was a research to detect straight lines with various orientation using *Principal Component Analysis* (PCA) [83]. PCA is a calculation used to analyse data on multi-dimension dataset, especially graphic. The PCA Method is used to reduce information without to having to eliminate its main information. The reduced information is deleted to expedite the performance of the system and to acquire accurate information. To implement this algorithm, it only needs input in the form of images. By processing the images using PCA *eigen value* straight lines will be found in certain *threshold*. To-date this algorithm has never been tested to implant solution system. It is declared that this algorithm will produce the same performa as Hough Transformation but with a simpler algorithm complexity.

Other researches detected traffic light by processing image input [14]. The basis of this detection process involves thresholding colour model, lessening of *noise* using morphology operation and resemblance to the shape ratio information. The shape/framework that have a ratio suitable to the traffic light object are taken and calculate the resemblance by using *template matching*. The introduction of the traffic light is made to compare the image taken by the mobile camera to the pictures that are in the database. This system can only be used by smart vehicle in complex urban surrounding.

In 2014 a research to detect pole objects such as traffic light, traffic sign posts and street lamp post based on *Mobile Laser Scanning Cloud* had been made. The *Mobile Laser Scanning Cloud Mobile Laser Scanning Cloud* is derived from *Light Detection and Ranging* (LiDAR) [13]. LiDAR is a long distance/remote sensor technology using the properties of scattered light to find range and get an object from the intended target. The detection method used depended on scanning point data *scanning point* from a sensor for further voxelization, *selection of candidates*,

and 3D reconstruction. In general, the working principle of LiDAR is a sensor that emits a laser beam on the target and then the light is reflected back to the sensor. The captured beam is then analysed by the detector equipment. The change in the light composition received from a target is designated as an object character. The journey time of the beam when emitted and receiving it again are required as determining variables in the calculation of distance of the object to the sensor. This special method developed to detect *pole-like street furniture objects* cannot be used if the sensors used are different, besides the sensoric equipment used cannot be acquired easily and is expensive. Other than the objective in the use of this system is for mapping and structuring urban facilities.

Study to detect pedestrian facilities has also been made by detecting the existence of steps and pedestrian crossing [84]. The advantage of this method over the other methods which are based on image is that this method processes and takes into account 3 image attributes: *Red, Green and Blue* to calculate the detection and the depth/distance to the picture is also one of the calculation factors. Hough transform is used in the detection process, but Support Vector Machine (SVM) is used for extra processing. SVM is a learning machine method which works on the *Structural Risk Minimization* (SRM) principle with the objective of finding the best *hyperplane* which separates 2 classes of space input. The patterns are members of 2 classes, +1 and -1, and divides alternative separation lines. Margin is the distance between these hyperplanes with the closest pattern of each class. The closest pattern is known as *support vector*. The attempt to find *hyperplane* locations is the main learning process to the SVM.

This research tries to improvise detecting pole object by using image input, compared to [13] which specialized on laser scanning cloud input. This research implements Hough transform with addition of vertical line connection process, so it can run with time efficiency complexity and not having to use all the steps in Hough transform. The research [83] claimed that the Hough transform has an expensive computation. However, when it was compared with [84] which also uses Hough transform, the algorithm does not need complex processing such as SVM and has been especially *tuned-in* for pole detection.

Method proposed in this research had been tested in the *embedded system* at a relatively low cost when compared to similar systems which have different method. This research does not use colour information model as performed by [14] because the colouring in the pole object does not follow certain colouring pattern. Compared to this research, the proposed method is relative faster because it does not use *template matching*. The speed and accuracy in the pole detection becomes a featured part of the algorithm developed in this study.

From the various related research, we can see that the development of the object detection is not to assist with people with special needs. The existing researches are directed to the usage of urban facilities [13], smart vehicle in the urban surrounding [14] or for pedestrians in the cities [84]. However, this research could be utilized by VIP as an assistive mobility tool. Briefly, related studies with this research and latest innovation in this research is illustrated in Table 4.1.

Table 4.1 Summary of related studies and latest innovation in this research

Specification	Y.S. Lee, 2006	C. Wang, 2011	C. Cabo, 2014	S. Wang, 2014	Yusro, 2016
Input System	Sensor based on image	Sensor based on image	Sensor based on LiDAR	Sensor based on image	Sensor based on image
Object Detection	Straight Lines	Traffic Light	Post such road furniture	Steps, pedestrian crossing and traffic signs	Poles
Implementation of hardware	None	Camera mounted on a vehicle	Implanted system	Implanted system	Implanted to the cane system
Method/algorithm	Eigen value Threshold	Recognition of pattern and verification of pattern/framework	<i>Laser scanning clouds</i>	Hough Transformation with	Calculation of distance and searching of vertical line pairing as a result of the optimization of Canny Edge detection and Hough Transformation Hough
Objectives	Not specific	Smart Vehicle in the cities	Management and mapping of Urban facilities	Assistive tool for pedestrian	Assistive tool for VIP

4.2 The Basis on the Development of YuRHos Pole Detection Algorithm

The Algorithm developed in this research is called **YuRHoS** pole detection algorithm, which is taken from the names the researchers (M. Yusro, K. Ramli, K.M. Hou and D. Sudiana). The development of the YuRHos pole detection algorithm uses the optimization of vertical Canny edge line detection and Hough

Transform. The Canny method is used to detect edge from a picture input [61][85]. In this method there is a process called *Non-Maximum Suppression* (NMS) which compares the pixel with the neighbouring pixel based on pixel gradient (θ/θ).

Optimising the vertical line edge detection was made in the NMS stage, by having a compared pixel is the pixel which is in the $\theta = 0^\circ$ group, but the pixel which has a group θ other than 0° will be ignored. The product of this process is vertical edge line to an image. Figure 4.1 (a) illustrates the process in the pole detection and Figure 4.1 (b) is the procedure in the Canny edge detection procedure.

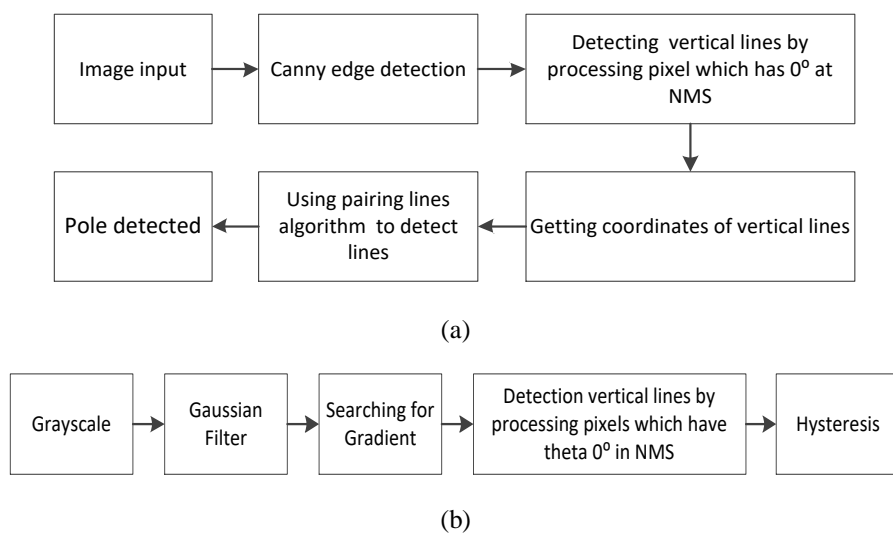


Figure 4.1 (a) Process in detecting pole (b) Canny Edge detection Procedure

The *Houghlines* function is producing group of vertical lines data which becomes the input basis for the algorithm in pole detection. These vertical lines are rough contour of a line. The YuRHoS algorithm will then process these lines through a series of operation sequence.

First step is to ensure vertical lines adjacent in the x-coordinate are connected to one line and edge set as the strongest edge of a pole. **Secondly**, all the advacent lines are paired off with one another due to the consideration that a pole is formed by two lines with a certain distance. This will result into a condtion where several pairs of lines may come from the same x coordinates. **The last step** from this algorithm is to choose the widest pairing lines as detected poles. In general, the pairing lines algorithm to detect poles can be seen from Figure 4.2.

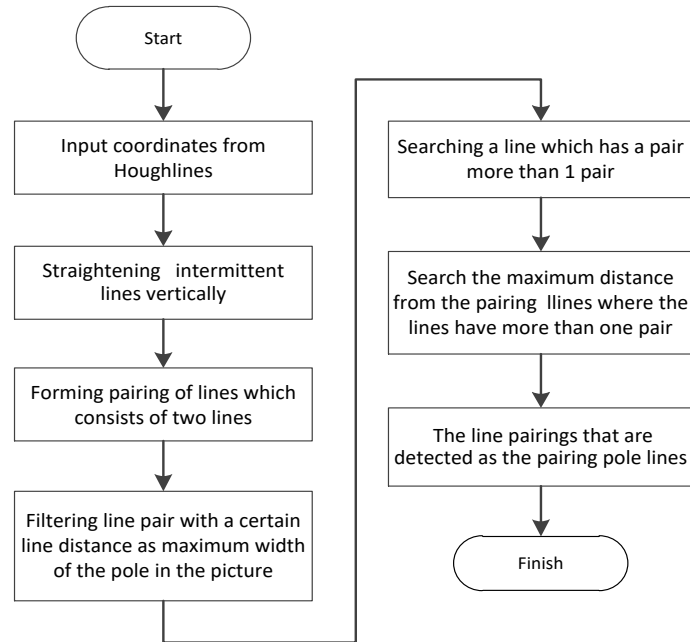


Figure 4.2 Algorithm of lines pairing for pole detection

4.3 Optimising Canny and use of Houghlines function

The Canny Edge detection algorithm begins with using Gaussian filter to eliminate noise from the picture input which will produce minimum noise to the final result to the edge detection. *Gaussian* function functions as refining the picture using *Gaussian function* equation. The following is ID Gaussian equation:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (4.1)$$

And the next is 2D *Gaussian* equation:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.2)$$

Where sigma (σ) is a standard deviation and pixel in the centre (x, y) gets the largest value of 1. Because the data in the picture are discrete, there is a need to make *Gaussian* distribution so as to apply *Gaussian* function to a picture. Gaussian

1D or 2D equations can be used to turn the *Gaussian* distribution into mask kernel which will convolute into the picture. However, if we use the *Gaussian* 1D equation, we need to the next derivative, because *Gaussian* 1D equation only has 1st coordinate, x , as a distance from its distribution *means*, while the picture data forms a matrix which has 2 coordinates (x dan y). Therefore, the *Gaussian* 1D equation needs 1 more derivative for the y coordinate and for its own mask kernel which is multiplication between the x and y values.

The σ notation states that deviation standard to the *Gaussian* function which refers to the refinement level in the *Gaussian* filter. The bigger the σ value is the more refined/blurry will the picture turn out which is produced by the filter. The bigger the σ value is we will need a big kernel measurement so that we can get an accurate result. In the *Gaussian* theory, there should not be any zero to *Gaussian* distribution at all places and therefore, there is a need to take into account that all pixels in the picture will cause a big kernel size. However, in practice, the distribution value of *Gaussian* whose distance is more than 3σ from *mean relative* approaching zero, and this will cause the size of the kernel could be cut and ignoring the *Gaussian* distribution value which relative approaching zero.

In this research, deviation standard $\sigma = 1$ is used, and therefore, there is a need to determine the size of the kernel from its *mask*. The following Figure 4.3 is *Gaussian* distribution for $\sigma = 1$ with the kernel size of 7×7 .

0.000036	0.000363	0.001446	0.002291	0.001446	0.000363	0.000036
0.000363	0.003676	0.014662	0.023226	0.014662	0.003676	0.000363
0.001446	0.014662	0.058488	0.092651	0.058488	0.014662	0.001446
0.002291	0.023226	0.092651	0.146768	0.092651	0.023226	0.002291
0.001446	0.014662	0.058488	0.092651	0.058488	0.014662	0.001446
0.000363	0.003676	0.014662	0.023226	0.014662	0.003676	0.000363
0.000036	0.000363	0.001446	0.002291	0.001446	0.000363	0.000036

Figure 4.3 Gaussian distribution with deviation standard of $\sigma=1$

We can see that from middle point the matrix which has the value of 0.146768 and a distance more than 3σ has the value of 0.001446 which is relatively close to zero and 101x more than 0.146768. Therefore, the cut on the kernel can be made so as to ignore the values that are almost zero. Figure 4.4 is a matrix which has been cut to 5×5 :

0.003765	0.015019	0.023792	0.015019	0.003765
0.015019	0.059912	0.094907	0.059912	0.015019
0.023792	0.094907	0.150342	0.094907	0.023792
0.015019	0.059912	0.094907	0.059912	0.015019
0.003765	0.015019	0.023792	0.015019	0.003765

Figure 4.4 Matrix 5x5

So as the matrix figures could be rounded off, the ends of the matrix figure could be changed to 1, so, all the figures in the matrix needs to be multiplied by $1/0.003765 = 265.6$ which is rounded off as 266 (Figure 4.5).

1	3.98911	6.319256	3.98911	1
3.98911	15.91288	25.2077	15.91288	3.98911
6.319256	25.2077	39.93147	25.2077	6.319256
3.98911	15.91288	25.2077	15.91288	3.98911
1	3.98911	6.319256	3.98911	1

Figure 4.5 Matrix multiplication of 265,6

Figure 4.6 is the result of rounding off:

1	4	6	4	1
4	16	25	16	4
6	25	40	25	6
4	16	25	16	4
1	4	6	4	1

Figure 4.6 Result of matrix rounding off

The matrix kernel form will look like Figure 4.7

$$\frac{1}{266} = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 25 & 16 & 4 \\ 6 & 25 & 40 & 25 & 6 \\ 4 & 16 & 25 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

Figure 4.7 Form of kernel matrix

Next Figure 4.8 (b) illustrates the result of the matrix filter on the sample of grayscale picture input to Figure 4.8 (a).

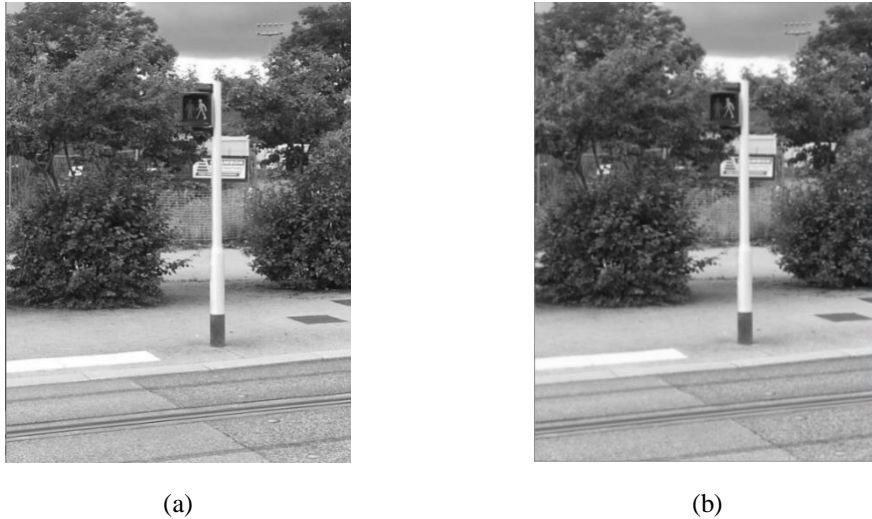


Figure 4.8 (a) Input of grayscale photo (b) Outcome from *Gaussian filter*

In the Canny algorithm, after filtering with *Gaussian filter*, the next step is finding the gradient value from the picture. The gradient value can be acquired by using *Sobel operation*. The equation for *Sobel operation* is [86], where dx and dy the distance along x any direction.

$$\frac{\partial f(x,y)}{\partial x} = \Delta x = \frac{f(x+dx,y) - f(x,y)}{dx} \quad (4.3)$$

$$\frac{\partial f(x,y)}{\partial y} = \Delta y = \frac{f(x,y+dy) - f(x,y)}{dy} \quad (4.4)$$

To do this, *Sobel operation* is being used. Every pixel in the picture is coorelated with G_x and G_y by using the matrix in Figure 4.9.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & 1 \end{bmatrix}$$

Figure 4.9 *Sobel operation Matrix*

With the *Pythagore's* equation, the *G* value is acquired from $\sqrt{Gx^2 + Gy^2}$. Figure 4.10 is the product of the picture determined by its gradient value from the picture that had been filtered by *Gaussian*.

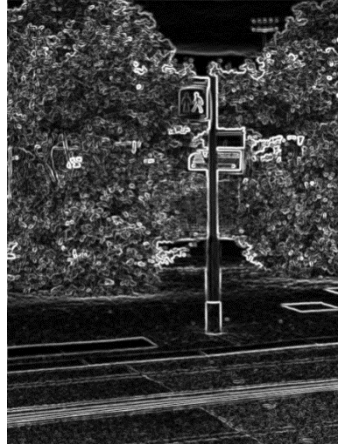


Figure 4.10 Gradient product from *Sobel* operation

G_y is attained by co-relating the picture pixel to *Sobel* mask kernel.

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (4.5)$$

Example of the co-relation:

$$\begin{aligned} \text{Picture Pixel} &= \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 1x1 & 2x2 & 3x1 \\ 4x0 & 5x0 & 6x0 \\ 7x-1 & 8x-2 & 9x-1 \end{bmatrix} \\ &= 1x1 + 2x2 + 3x1 + 4x0 + 5x0 + 6x0 + 7x-1 + 8x-2 + 9x-1 \\ &= 1 + 4 + 3 + 0 + 0 + 0 + (-7) + (-16) + (-9) \\ &= -24 \end{aligned}$$

Value -24 is placed into pixel 5

G_x is attained byu co-relating the picture pixel to the *Sobel* mask.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (4.6)$$

Then, the value of the gradient direction could be calculated from every pixel of the picture by using the following equation:

$$\theta = \tan^{-1} \frac{Gy}{Gx} \quad (4.7)$$

In the Canny algorithm the θ value acquired will be divided into 4 groups that approximately 0° , 45° , 90° dan 135° with 0° to 22.5° and 157.5° to 180° to be grouped as 0° . 22.5° to 67.5° to be grouped as 45° . 67.5° to 112.5° to be grouped as 90° and 112.5° to 157.5° to be grouped as 135° . Table 4.2 illustrates the grouping of the gradient direction.

Table 4.2 Grouping of the gradient direction

Group 0° and 180°	Group 45°	Group 90°	Group 135°
0 – 22.5 and 157.5 - 180	22.5 – 67.5	67.5 – 112.5	112.5 – 157.5

In the Canny edge detection, the value of the gradient direction (θ) determines the comparative direction of the gradient value (G) of a picture pixel towards its neighbouring pixel. If the pixel value is bigger than its neighbouring pixel, then this value will be maintained, if not, it will be deleted. The comparison direction of the pixel will follow the θ value. If it is 0° then it will be compared to its right a left neighbor, if it is 45° then it will be compared to diagonally left and right neighbor, if it is 90° then it will be compared and finally, if it is 135° it will be compared to the value on top left diagonal to bottom right diagonal. In the Canny edge detection algorithm, this stage is called *Non – Maximum Suppresion* (NMS) method.

NMS method is a thinning edge mechanism by implementing comparative method of G (gradient) value from each pixel with its neighbours Figure 4.11 is the product of the picture using NMS method.

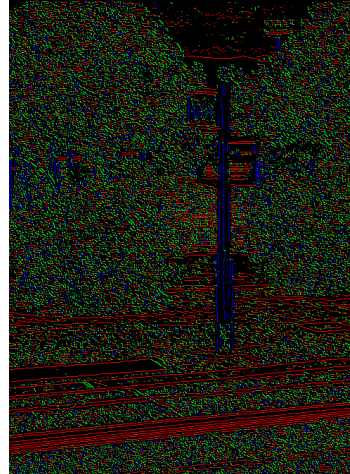


Figure 4.11 Product of NMS method

The comparison between the result of the *Sobel* operation and NMS method can be seen where the result from the NMS method is thinner from the previous input. The result of the application in the calculation of gradient to every colour in Figure 4.11 represents the group of gradient direction, such as blue represents 0° atau 180° angles, red represents 90° , green represents 135° and yellow represents 45° .

In the NMS stage, modification is made so that only perpendicular lines are displayed, with the ruling that only those grouped in $\theta=0^\circ$ are maintain, while the rest are eliminated. To get vertical lines, the direction of the edge is in the 0° group. The illustration in Figure 4.12 only shows that the result of the picture in 0° group. So that whe writing the pixel in 0° group only, we will get information on vertical edge line.

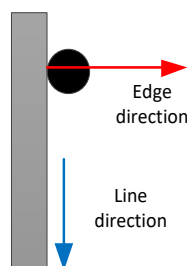


Figure 4.12 Value of θ 0° group

The result of the elimination of pixels which are not 0° is illustrated in Figure 4.13.



Figure 4.13 Result of the group of $\theta=0^\circ$ value

The following are the criteria on NMS which have been modified to maintain the pixel with 0° angle:

1. For the 0° position, if the picture pixel is bigger than the neighbouring pixel, then these pixels will be maintained.
2. For 45° position, if the picture pixel is the largest when compared to the neighbouring pixel, then these pixels will be suppressed or set at 0.
3. For 90° position, if the picture pixel is the largest when compared to the neighbouring pixel, then these pixels will be suppressed or set at 0.
4. For 135° position, if the picture pixel is the largest when compared to the neighbouring pixel then these pixels will be suppressed or set at 0.

The next step in following the Canny algorithm is Hysteresis that is eliminating undesired noise by applying two threshold parameters. The two parameters are divided into 3 parts, noise, strong and weak pixels. If the pixel value is below the minimum threshold parameter, it is called noise. If the value is bigger than the maximum threshold parameter, then it is called strong pixel, and finally, if the value of the pixel is between the two parameters then it is called weak pixel.

Then the noise will be eliminated by giving 0 value to this pixel. The value to the strong pixel will be increased to a maximum value of 8-bit 1 *channel* to picture to 255. Then the weak pixel will be taken into further account by using 3x3 kernel.

The 3x3 kernel is 3x3 matrix used as a mask or area to calculate the weak pixel. If a pixel enters into the weak pixels group, the pixel will be set at 255 or eliminated. This is determined by looking at its neighbours with a scope space of 3x3 with a weak pixel to be middle point. Figure 4.14 illustrates 3x3 matrix as a mask for weak pixel.

$$\begin{array}{ccc} P_0 & P_1 & P_2 \\ P_3 & P_{weak} & P_5 \\ P_6 & P_7 & P_8 \end{array}$$

Figure 4.14 3x3 matrix as mask

If there is a strong pixel to $P_0 - P_8$ then P_{weak} is placed in the strong pixel and is given a pixel value of 255. On the other hand, if there is no strong pixel, the P_{weak} will be eliminated. Figure 4.15 illustrates graph in the illustration of hysteresis process.

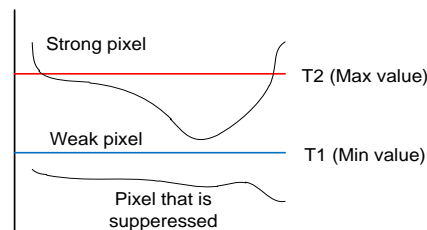


Figure 4.15 Graph of the hysteresis process

The emphasis stage is also mentioned in this hysteresis stage that is there is only 2-pixel values, 255 and 0. In this stage, the edge lines can be clearly seen and there is no noise edge pixel. Figure 4.16 illustrates the sample of the result of hysteresis.

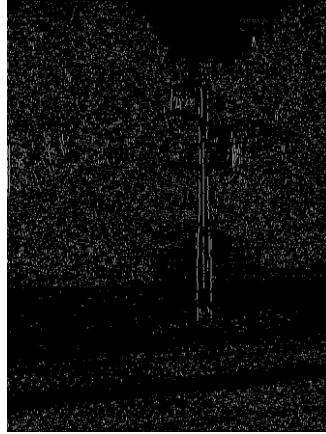


Figure 4.16 Result of the hysteresis process

At the end of the Canny detection, of which the NMS section had been modified to optimize the perpendicular line displayed, is continued with the Houghlines function which is used to detect shaped straight lines with filter in the form of certain long lines. The lines are acquired from an info in the form of coordinates to the picture. Figure 4.17 display the result of Houghlines displayed in the original picture.



Figure 4.17 Result of Houghlines process

In Figure 4.17 we can see that there are detected lines on the post which is indicated by a red line. This line is the product of Houghlines from the picture detected by Canny edge and optimized to detect vertical lines only.

Because displayed in the Figure 4.18 are broken lines due to the result of edge detection, we need to have a function to straighten the lines. In the algorithm, if there is a difference between 2 pixels, then the line will be considered a solid line. Figure 4.18 display the result of optimizing in straightening the lines.



Figure 4.18 Optimising Result in straightening the lines

CHAPTER 5
ANALYSIS ON THE PERFORMANCE OF YuRHoS POLE DETECTION
ALGORITHM

5.1 YuRHoS Pole Detection Algorithm

This algorithm needs Canny operation to detect edge and Houghlines function to obtain line coordinates. The line coordinates are then saved in point pairing data structure, where a line is drawn from two points. One point is saved in the (x, y) array which represents these points in the Cartesian space.

The structure in the point pairing data is saved as L array, where $L = [X_1, X_2, Y_1, Y_2]$.

X_1 : Array from X coordinate for the 1st point

X_2 : Array from X coordinate for the 2nd point

Y_1 : Array from Y coordinate for the 1st point

Y_2 : Array from Y coordinate for 2nd point

In the alternative form, L can be illustrated as a matrix to equation 5.1.

$$L = \begin{bmatrix} x_{11} & x_{12} & y_{11} & y_{12} \\ x_{21} & x_{22} & y_{21} & y_{22} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & y_{n1} & y_{n2} \end{bmatrix} \quad (5.1)$$

Where n exhibits the number of lines detected. In this case, the straight vertical lines will have the same x_1 and x_2 values so it can also be written as $L = [X, Y_1, Y_2]$.

In the case of pole detection, the pair points will have the same x coordinates, because the pole object is considered to be straight vertical line. This algorithm will process L input and produce G which is the pairing of the vertical lines detected as pole. G is measured as $m \times 3$, where m is the number of poles detected. $G = [P_1, P_2, D]$, and $P = [p_1, p_2, \dots, p_m]^T$

P_1 : a set of 1st line, P_2 : a set of 2nd line, and D : a set of distance from p_1 to p_2 .

The YuRHoS algorithm operates as illustrated in Figure 5.1.

1. **Algorithm** pole_detection
2. **Input:** L matrix, saved in point pairing array
3. **Output:** G matrix, which represents detected pole
4. $L_2 \leftarrow \text{connect_line_segment}(L)$, connecting vertical broken lines
5. $G \leftarrow \text{detect_pole}(L_2)$
6. **Return** G

Figure 5.1 YuRHoS pole detection Algorithm

This algorithm summons 2 other sub-functions, *connect_line_segment* is used to connect straight vertical broken lines to turn into a single vertical line for nearby x coordinates. Next *detect_pole* is used to look for pairs of close by vertical lines as characteristic of pole.

5.1.1. Connect_line_segment function

The input parameter from this procedure is L matrix and produce L_2 matrix output, which is L that have been connected. The objective of this algorithm is to connect vertical lines that are within the distance of the nearest vertical lines or x_1 : x position at the 1st point, x_2 : x point in the 2nd point, and d : distance between x_1 and x_2 , thus $d := |x_1 - x_2|$, $d \leq \text{threshold}$.

Then, for the lines that fulfil the criteria or considered to be within the nearest x coordinate, the two lines will be connected to form a line, or y_{min} is the lowest coordinate from the two differentiated lines, y_{max} is the highest coordinate from two differentiated lines and $y_{min} = \min(y_1, y_2, y_3, y_4)$, $y_{max} = \max(y_1, y_2, y_3, y_4)$. One line saves 2 y coordinates so that the two lines are compared it will have y coordinates.

Every point is compared with all points including the point itself. This process is called a *sweep*. Consider $n :=$ a number of lines detected, then 1 sweep will run in n number of time. Thus will run n^2 sweep so as to compare all lines. This process can be streamlined by sorting prior to the coordinates x , which will ensure the most

$$L_2^T =$$

x	3
y_1	274
y_2	295
	k

The initial value of k is 0 index and has a function as the index of L_2 matrix. The initial of i is index 1, and k is index 0. $x(i)$ and x_{pivot} will be compared if the distance value of $|x_{pivot} - L_x(i)| < thres$, on this iteration, $|3-6| \geq thres$, till the lines are disconnected and *updated* L_2 is operated by making an entry of the new lines taken from L and increase the value of k . The pivot value is always updated to the line on L_2 matrix to k index.

$$L_2^T =$$

x	3	6
y_1	274	226
y_2	295	258
		k (<i>pivot</i>)

On the 2nd **iteration**, i is in the 2nd index.

$$L^T =$$

x	3	6	6	258	299	300	302	302	302	306
y_1	274	226	276	153	481	386	204	285	318	153
y_2	295	258	300	171	500	401	220	303	335	181
			i							

The $|x_{pivot} - L_x(i)|$ value will be re-calculated, $|6-6|$ is included in the update criteria so that L_2 value will be *updated* by looking for minimum and maximum between the 4 points : 2 points on y_{pivot} and 2 points on $L_y(i)$.

$$L_2^T =$$

x	3	6
y_1	274	226
y_2	295	300
		k (<i>pivot</i>)

On the 3rd iteration, i is in the 3rd index.

$L^T =$

x	3	6	6	258	299	300	302	302	302	306		
y_1	274	226	276	153	481	386	204	285	318	153		
y_2	295	258	300	171	500	401	220	303	335	181		
				i								

The distance between x_{pivot} and $L_x(i)$ is too far, therefore a new entry on L_2 will be made, The same thing happens to 4th iteration.

$L_2^T =$

x	3	6	258	299
y_1	274	226	153	481
y_2	295	300	171	500
			$k (pivot)$	

On the 5th iteration **ke-5**, i is in the 5th index

x	3	6	6	258	299	300	302	302	302	306	
y_1	274	226	276	153	481	386	204	285	318	153	
y_2	295	258	300	171	500	401	220	303	335	181	
				i							

The difference distance between x_{pivot} and $L_x(i)$ is only 1, till the entry of y to L_2 is updated by taking the minimum and maximum from the four points: $\{481,500,386,401\}$, and entry x to L_2 is updated by filling it from $L_x(i)$

x	3	6	258	300
y_1	274	226	153	386
y_2	295	300	171	500
			$k (pivot)$	

The updated condition towards L_2 will keep on occurring until the 8th iteration On the 9th iteration

$L^T =$

x	3	6	6	258	299	300	302	302	302	306	
y_1	274	226	276	153	481	386	204	285	318	153	
y_2	295	258	300	171	500	401	220	303	335	181	
				i							

$$L_2^T =$$

x	3	6	258	302
y_1	274	226	153	204
y_2	295	300	171	500
				k (pivot)

The distance between x_{pivot} and $L_x(i)$ will be out of the *thres* limit and the new value to L_2 will be inserted.

x	3	6	258	302	306
y_1	274	226	153	204	153
y_2	295	300	171	500	181
					k (pivot)

The next iteration will follow the same procedure until all inputs have been process.

5.1.2. Detect_pole Function

The input parameter to this procedure is matrices L resulted from the `connect_line_segment` function and producing G . This algorithm has the task of pairing the nearest paired lines or the distance of coordinate x between 2 lines are less than a certain threshold where the maximum distance will later be taken. This is done with the consideration that the assumption is that the detected pole is a pair of adjacent horizontal lines. Only the paired lines fulfilling the criteria will be saved as G .

```

1 Algorithm detect_pole
2   Input:  $L$ , which saves the line data that have been connected
3            $thres$ , as the horizontal distance tolerance threshold
4   Output:  $G$ , matrix which save the the pairs with maximum distance
5           Let  $n$  as length from  $L$ 
6   Initialize  $G$  a empty matrix with 3 columns
7   Initialize  $D$  as matrix distance measuring  $n \times n$ 
8           Let  $k$  as index tracker from  $G$ 
9            $k := 1$ 
10  For  $i := 0$  to  $n-1$ 
11    For  $j := i+1$  to  $n-1$ 
12      Let  $d$  as  $x$  distance between 2 lines
13      Set  $d := |x_i - x_j|$ 
14      If  $d \leq thres$  AND  $i$  not equal  $j$  then
15         $D(i,j) = d$ 
16      Initialize  $max := 0$ 
17      Initialize  $imax := 0$ 
18      Initialize  $ctr := 0$ 
19      For  $j := i+1$  to  $n-1$ 

```

```

20         If  $D(i, j) > max$ 
21             Update  $max := D(i, j)$ 
22                 Update  $imax := j$ 
23                 Update  $ctr := ctr + 1$ 
24         If  $ctr > 1$ 
25             Set  $G(k, 1) := i$ 
26             Set  $G(k, 2) := j$ 
27             Set  $G(k, 3) := max$ 
28             Update  $k := k + 1$ 

```

Figure 5.3 Detect_pole Algorithm

This algorithm has two tasks, finding paired lines which has a distance less than *threshold*, and then from the pairs which have been filtered determine a line a strong candidate.

From the paired line which has the same origin of the left vertical line, look for a paired line with longest horizontal distance. For the first goal, each pair of lines that meet the criteria are stored in a distance matrix D. To find this paired lines, each line will be paired with other lines and the distance noted. For example, the output of the connect_line segment algorithm is the line with next x coordinate.

$$X = \{3, 6, 258, 302, 308, 317, 320, 354\}$$

The number of X member, $|X| = 8$, then 6-15 lines will result in the next *distance matrix* if the value of the *threshold* is 20.

$D =$

0	3	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	6	15	18	0
0	0	0	0	0	9	12	0
0	0	0	0	0	0	3	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

The paired lines that are formed are as follows: (1, 2), (4, 5), (4, 6), (4, 7), (5, 6), (5, 7), (6, 7). The matrix is filled only on the side of the triangle, the algorithm is designed to be redundant line pair which does not include, for example (1, 2) redundant with (2, 1).

Line 16 – 23, has the task of finding the farthest paired lines which comes from the same starting line. The farthest distance will be saved in *max*, and the index of the farthest line will be saved in *imax*, and each pairs detected will be calculated via *ctr*. After this line is operated, it will produce the pairs of these farthest lines (1, 2), (4, 7), (5, 7), (6, 7), with a counter values of each lines: *ctr* = {1, 0, 0, 3, 2, 1, 0, 0}

Line 24 – 28, has the task of filtering the pairs which has a counter value of *ctr* > 1. So it will only generated (4, 7) & (5, 7) pairs. In the last iteration, it will only produce the following pairs,

$$G =$$

```

4  7  18
5  7  12

```

These two lines displays the two poles that are detected, where the poles are wedged in by 2 vertical lines. The first pole is wedged by a 4 and 7 lines with a width of 18, and the second pole is being wedged in with 5 and 7 line with a width of 12.

5.2 Evaluation of YuRHoS algorithm

Below is the product of *log running* algorithm and the test on pole detection program *Matlab* is used to operate the algorithm by entering the Houghlines data to each pictures and *Eclipse* is used to operate detection program by testing several pictures.

From both results, between log operating the algorithm and pole detection program, we are able to see the consistency in the detection of poles. The green rectangular box on the figure shows the poles detected in the log running the algorithm and the blue rectangular box in the picture shows that the detected poles in the application program. On case 1 (Figure 5.4), there are 2-3 poles that are detected however all of them detected the same pole. Therefore, we are able to conclude that the result of the detection is correct,

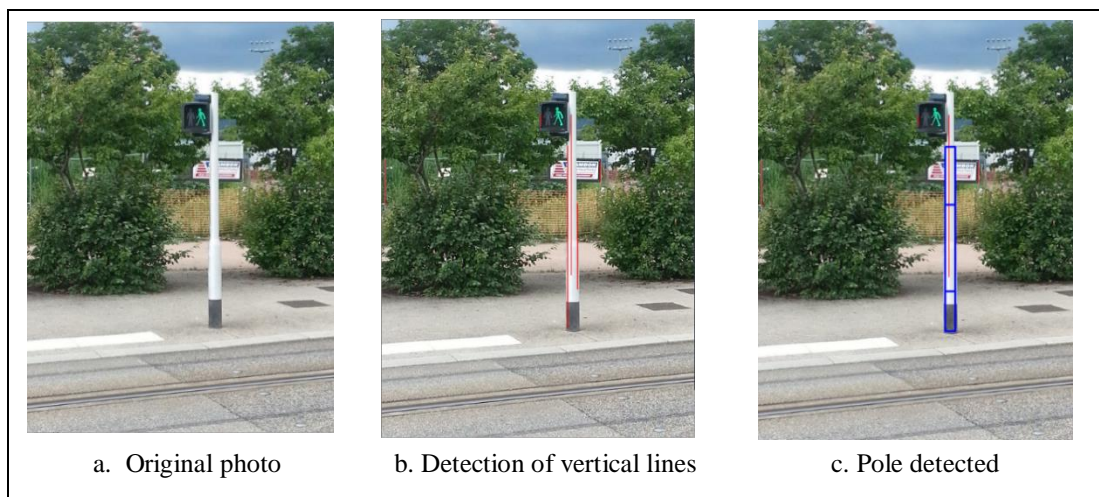
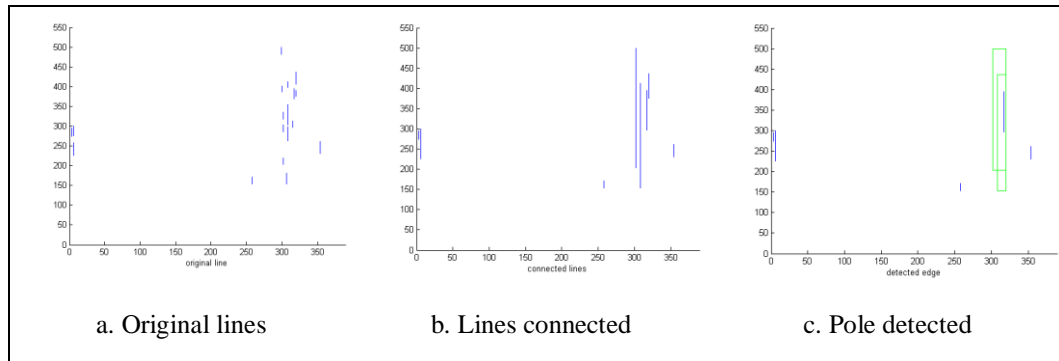
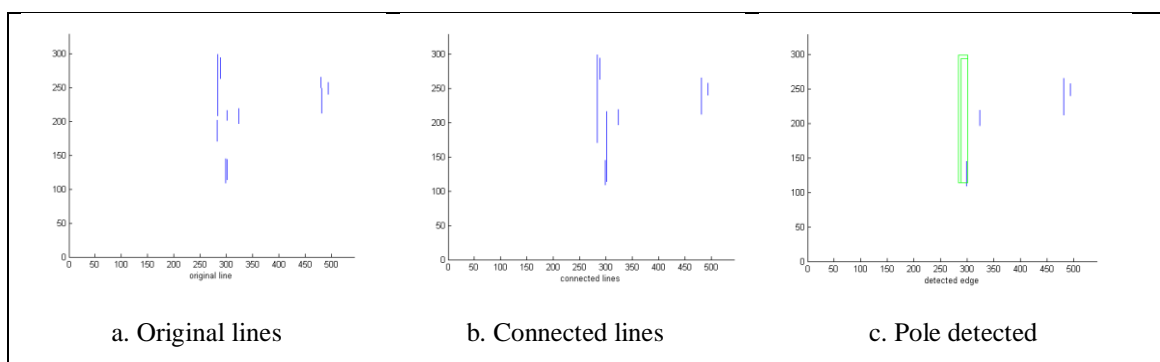


Figure 5.4 Algorithm Test and pole detection program of picture 1

In case 2 (Figure 5.5), the input picture consists of short and long distance poles. As a note, all poles are indicated with vertical Houghlines, however, the result only detects one pole. This is because, although both of the produces vertical Houghlines but the poles located the farthest will be eliminated because the width of these poles are small.



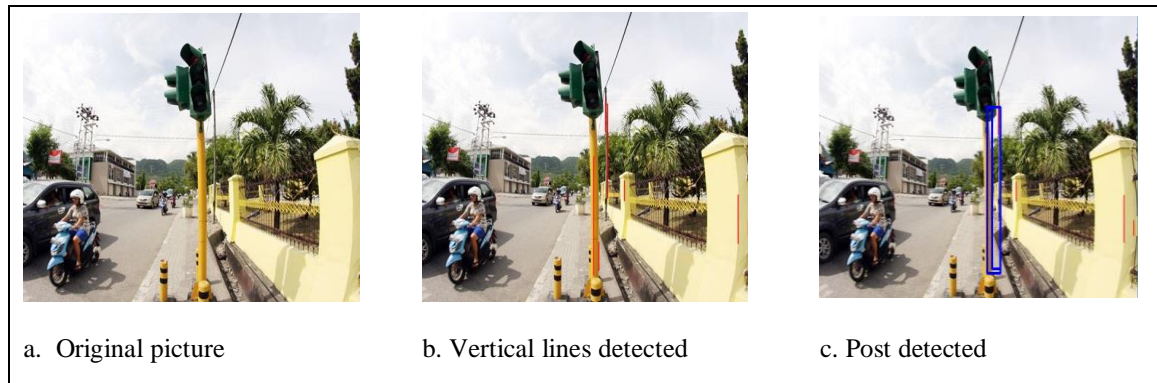


Figure 5.5 Algorithm Test and Pole detection program to Picture 2

In Case 3 (Figure 5.6), the picture input consists of a series of poles on one road. Notice that all poles have certain vertical Houghlines value, for the nearest pole, both sides of the poles are correctly specified. However, for the poles that are far away, the distance between the two sides of the poles are too small and therefore, cannot be detected. The important thing here is that even though only nearby pole is detected, but as the user walks towards the pole, the pole will become closer. The algorithm will operate again until the closest poles are detected again.

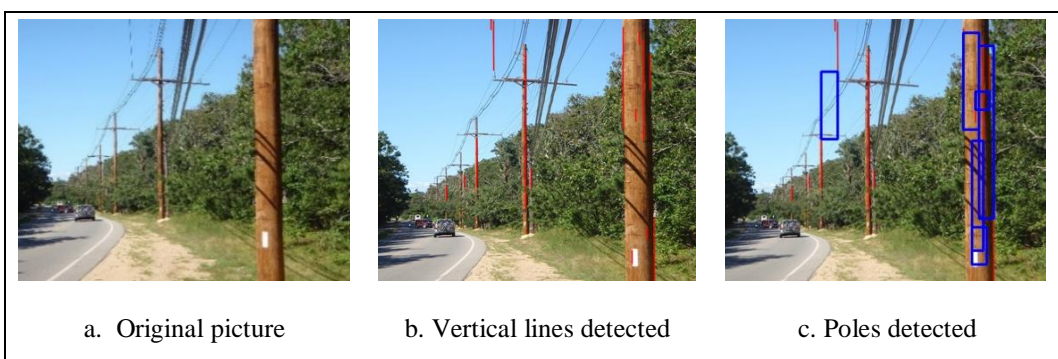
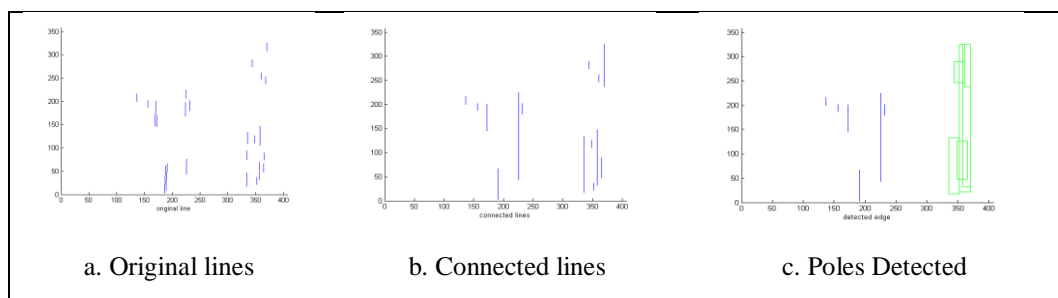


Figure 5.6 Algorithm Test and pole detection program to photo 3

5.3 Analysis Comparison on Pole Detection Algorithm

To test the accuracy on the pole detection from the YuRHoS algorithm, the YuRHoS will be compared to similar pole detection algorithm known as Non-YuRHoS algorithm. This comparative algorithm has a working procedure as illustrated in Figure 5.7

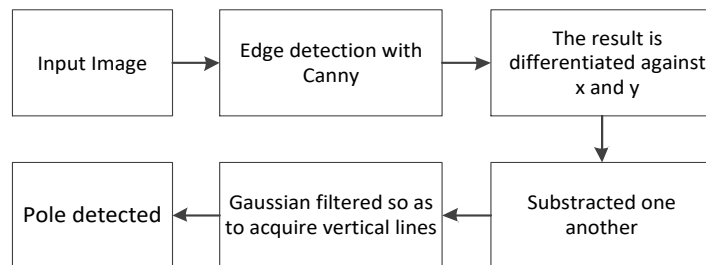


Figure 5.7 Working procedure of non-YuRHoS algorithm

To achieve picture 5.7, we need to build a program using C++ language which uses *library image processing* belonging to OpenCV [68]. The edge detection process using Canny method is operated by writing the following program lines.

```
cvCanny( dst, dst, 96, 96*3, 3 );
```

The *Canny* method uses two *thresholds*, which are useful in detecting edges that can be seen clearly, and the edges that cannot be seen clearly or weak

The above *cvCanny* line program (*dst, dst, 96, 96*3, 3*) illustrates value as follows: 96 is the 1st *threshold* value and 96*3 is the 2nd *threshold* value. Next the product of the Canny edge detection are differentiated against *x* and also against *y*, the result of this differentiation is subtracted against one another. The objective of this process is to get edge lines which are straight vertical lines. These vertical lines have big gradients horizontally and a small gradient vertically so the difference between them is very big.

The vertical lines obtained from the subtraction process is continued with convolution process by using *Gaussian filter*. The objective of this process is to upgrade the quality of the image (*image enhancement*). Then, the final process in

the line detection is identifying picture information caught by the camera is a pole. In this experiment, the writer uses basic assumption that the definition of pole is two vertical parallel lines with almost the same length. For identification process on vertical lines, the writer uses Hough transformation or Houghlines function found in Library OpenCV. To establish this, *cvHoughLines2* [67] image processing function in the OpenCV is used. In this research, the comparison in the detection result of the two pole detection algorithms are tested via a simulation using Eclipse software [87].

5.3.1. Testing for single pole and multi poles

a. The Procedure of Algorithm Simulation in the Detection of Poles

The procedure of algorithm simulation in the detection of poles of Non-YuRHoS and YuRHoS is explained in Table 5.1 below.

Table 5.1 The simulation procedure on poles

YuRHoS and Non-YuRHoS
1. Expert judgment is used to classify the poles and non-poles in the image examined.
2. The use of image template for TP score (<i>True Positive</i>), TN score (<i>True Negative</i>), FP score (<i>False Positive</i>) and FN score (<i>False Negative</i>) from the result in the detection of poles.
3. The use of confusion matrix for scores of sensitivity, specificity and accuracy from the result in the detection of poles.
4. To compare the result of simulation in the detection of YuRHoS and Non-YuRHoS.

The simulation of the algorithm result of the detection of YuRHoS and Non YuRHoS starts with classifying objects in the image as pole object (positive object) and non-pole object (negative object). The determination of positive or negative objects (pole or non-pole) is based on expert judgment via image processing to classify objects from the image examined [88]. In theory, expert judgment can be used in four functions [89], namely: a) to complete, validate, interpret and integrate existing data, to assess the impact of changes; b) to predict future incidents and consequences from decisions made; c) to determine the condition of knowledge in

a field; d) to provide elements required for decision making between several acceptable alternatives.

In the field of machine learning, particularly in statistical classification, confusion matrix/error matrix of specific tables, which facilitate the visualization of the performance of an algorithm, is used [90]. In the confusion matrix (Figure 5.8), each matrix column represents an example of predicted condition/result, meanwhile, each matrix row represents an example of true conditions.

		Predicted condition	
		Predicted Condition positive	Predicted Condition negative
True condition	condition positive	True positive	False Negative (Type II error)
	condition negative	False Positive (Type I error)	True negative

Figure 5.8 Confusion Matrix

The explanations of the confusion matrix from Figure 5.8 are as follow:

- True Positive (TP)** is a condition where something is identified as correctly identified. In this research, true positive is obtained when the marker of pole-detection is in the template of the image of pole.
- False Positive (FP)** is a condition where something is incorrectly identified. In this research, false positive is obtained when the marker of pole-detection is outside the template of the image of pole.
- True Negative (TN)** is a condition of correctly rejected. In this research, true negative is obtained when no marker of pole-detection is in objects other than poles.
- False Negative (FN)** is a condition of incorrectly rejected. In this research, false negative is obtained when there is no marker of pole-detection in pole objects.

To calculate the algorithm performance in machine learning, there are three aspects of measurement that can be used [91] [92]; sensitivity, specificity and accuracy. **Sensitivity** (tru positive level) which measures the positive proportion correctly identified. **Specificity** (true negative level) which measures the negative proportion correctly identified. **Accuracy** refers to the closeness of value measured with standard values or known values. Whereas, the formulation of the three aspects is [93] :

$$\text{Sensitivity (sen)} = \frac{TP}{TP+FN}$$

$$\text{Specificity (spc)} = \frac{TN}{TN+FP}$$

$$\text{Accuracy (acc)} = \frac{TP+TN}{TP+FP+TN+FN}$$

The evaluation of the simulation of both algorithms in pole-detection is obtained by using sensitivity, specificity, and accuracy score [90]. The scores depend on four variables; **true positive**, **true negative**, **false positive** and **false negative**. The calculation process is obtained by matching pixel detection area with ground-truth area [94]. In example, in a pole area, four points are detected, therefore, is is considered as 1 detection. Therefore, a group of poles is clustered as 1 pole per matched ground-truth area.

In this research, sensitivity is the ability of algorithm in predicting the correct result; specificity is the ability of algorithm in detecting the correct result by diverting false result; and accuracy is the amount of correct results compared to existing population.

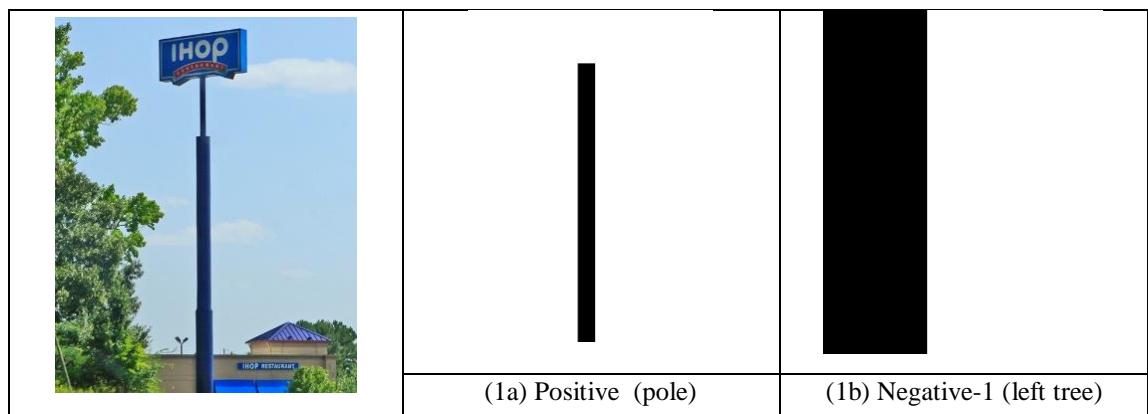
b. Labeling Procedure by Experts

In this simulation, the images examined are a one pole image and the image of poles. The algorithm execution resulted in markers for every pole detected/found, whereas, markers are able to identify poles correctly or incorrectly. To evaluate the result of the simulation automatically (using a program), therefore, each image is labeled (ground-truth) between pole objects (positive object) and non-pole objects

(negative object). The labeling mechanism is obtained by using procedures as follow:

1. For every image, the expert will determine which objects are positive and which are negative.
2. Label every positive object (pole images). The positive objects are read one by one and then labeled with black color, meanwhile, other images are in white color. Every positive label is saved in a file named $\langle name_file \rangle_t\langle i \rangle.png$ where i stands for *truth* (index starts from 1), in example, $g1_t1.png$ (meaning $g1$ =image examined number 1, $t1$ =positive object number 1 or, *true-1*). After one positive object is labeled into a new image, other positive object is labeled and so forth until every positive object is read.
3. Label every negative object (non-pole images). The characteristic of negative objects is having vertical edges. Negative objects are then read one by one, whereas, every negative object is labeled in black color and other objects are in white color. Every negative label is saved in a file named $\langle name_file \rangle_f\langle i \rangle.png$, in example, $g1_f1.png$ (meaning, $g1$ =image examined number 1, $f1$ =negative object number 1 or, *false-1*). After one negative object is labeled into a new image, other negative objects are labeled and so forth until every negative object is read.

Below are examples of images examined and labeled (ground-truth) with positive and negative object (Figure 5.9). Labeling is performed for every image, based on expert judgment.











1. Single Pole Image		
	(1c) Negative-2 (billboard)	(1d) Negative-3 (building)
 2. Multi-Poles Image		
	(2a) Positive-1 (pole 1)	(2b) Positive-2 (pole 2)
		
	(2c) Positive-3 (pole 3)	(2d) Negative-1 (cars+motorcycles)
		
	(2e) Negative-2 (lamp pole)	(2f) Negative-3 (wall)

Figure 5.9 Labeling Process (Ground-Truth)

c. The Result of Algorithm Simulation on Non-YuRHoS and YuRHoS

Figure 5.10 is showing ten one-pole images which were tested using two different algorithms (Non-YuRHoS and YuRHoS).



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)






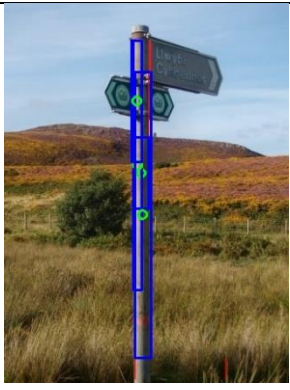





(j)


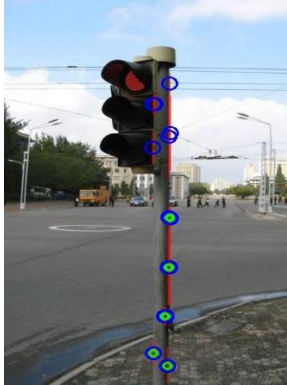
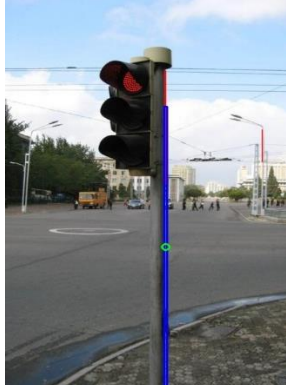



Figure 5.10 Images for One-Pole Simulation




The result on the simulation of one-pole images using algorithm in the detection of Non-YuRHoS and YuRHoS poles is presented in Figure 5.11 below:


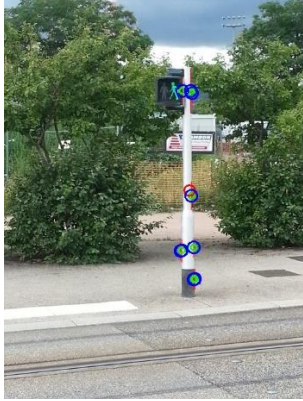




Image Examined	Result for Non YuRHoS	Result for YuRHoS
----------------	-----------------------	-------------------




		
	<p>TP = 1, FN = 0, TN = 0, FP = 1</p>	<p>TP = 1, FN = 0, TN = 1, FP = 0</p>
<p>Expert judgment: one pole object and one non-pole object (street sign)</p>	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole detected as pole. - TN = 0; no non-pole object detected as non-pole. - FP = 1; one non-pole object detected as pole (street sign). 	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 1; one non-pole object not detected as non-pole. - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,5$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Conclusions derived from the Simulation :</p>		
<ul style="list-style-type: none"> - YuRHoS Algorithm has a better specificity (score 1) compared to Non-YuRHoS Algorithm (score 0). YuRHoS Algorithm is capable of detecting the correct result (pole) and able to detect the incorrect result (non-pole). - YuRHoS Algorithm is better in accuracy (score 1) in comparison to Non-YuRHoS Algorithm (score 0,5). YuRHoS Algorithm is a capable of detecting the correct result (pole) more out than of every object detected. 		
		
	<p>TP = 1, FN = 0, TN = 3, FP = 1</p>	<p>TP = 1, FN = 0, TN = 4, FP = 0</p>
<p>Expert judgment: one pole object and four non-pole</p>	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. 	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole.

objects (three street signs and one tree)	<ul style="list-style-type: none"> - TN = 3; three non-pole objects detected as non-pole. - FP = 1; one non-pole object detected as pole (top street sign). 	<ul style="list-style-type: none"> - TN = 4; four non-pole objects detected as pole. - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0,75$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,8$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Conclusions derived from the simulation :</p> <ul style="list-style-type: none"> - YuRHoS Algorithm has a better specificity (score 1) compared to Non-YuRHoS Algorithm (score 0,75). YuRHoS Algorithm is capable of detecting the correct result (pole) and the incorrect ones (non pole). - YuRHoS Algorithm has a better accuracy (score 1) in comparison to Non-YuRHoS Algorithm (score 0,8). YuRHoS Algorithm is capable of detecting the correct result (pole) more out of all the objects detected. 		
		
	TP = 1, FN = 0, TN = 2, FP = 0	TP = 1, FN = 0, TN = 2, FP = 0
Expert judgment: one pole object and two non-pole objects (two street signs)	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 2; two non-pole objects detected as non-pole. - FP = 0; no non-pole object detected as pole. 	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 2; two non-pole objects detected as non-pole. - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Conclusion derived from the Simulation :</p> <ul style="list-style-type: none"> - Both Algorithms (YurHoS and Non-YuRHoS) have the same capabilities in detecting poles. 		

		
	TP = 1, FN = 0, TN = 0, FP = 1	TP = 1, FN = 0, TN = 1, FP = 0
Expert Judgment: one pole object and one non-pole object (traffic light)	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 0; no non-pole object detected as non pole. - FP = 1; one non-pole object detected as pole (traffic light). 	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 1; one non-pole object detected as non-pole. - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,5$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
Conclusions derived from the Simulation :		
<ul style="list-style-type: none"> - YuRHoS Algorithm has a better specificity (score 1) compared to Non-YuRHoS Algorithm (score 0). YuRHoS Algorithm is capable of detecting the correct result (pole) and incorrect ones (non-pole). - YuRHoS Algorithm is better in accuracy (score 1) in comparison to Non-YuRHoS Algorithm (score 0,5). YuRHoS Algorithm is more capable of detecting the correct result (pole) out of all the objects detected. 		
		
	TP = 1, FN = 0, TN = 1, FP = 1	TP = 1, FN = 0, TN = 1, FP = 1
Expert Judgment: one pole object and two non-pole	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. 	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole.

objects (one STOP sign and one car).	- TN = 1; one non-pole object detected as non-pole (a car). - FP = 1; one non-pole object detected as pole (STOP sign).	- TN = 1; one non-pole object detected as non-pole (a car). - FP = 1; one non-pole object detected as pole (STOP sign).
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0,5$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,67$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0,5$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,67$
Conclusion derived from the Simulation :		
- Both Algorithms (YurHoS and Non-YuRHoS) have the same capabilities in detecting poles.		
		
	TP = 1, FN = 0, TN = 1, FP = 1	TP = 1, FN = 0, TN = 2, FP = 0
Expert Judgment: one pole object and two non-pole objects (STOP sign and a house)	- TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 1; one non-pole object detected as non-pole (STOP sign) - FP = 1; one non-pole object detected as pole (a house).	- TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 2; two non-pole objects detected as non-pole (STOP sign and a house) - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0,5$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,67$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
Conclusions derived from the Simulation :		
<ul style="list-style-type: none"> - YuRHoS Algorithm has a better specificity (score 1) compared to the Non-YuRHoS Algorithm (score 0,5). YuRHoS Algorithm is capable of detecting the correct result (pole) and incorrect ones (non-pole). - YuRHoS Algorithm is better in accuracy (score 1) in comparison to Non-YuRHoS Algorithm (score 0,67). YuRHoS Algorithm is more capable of detecting the correct result (pole) out of all the objects detected. 		

		
	<p>TP = 1, FN = 0, TN = 2, FP = 0</p>	<p>TP = 1, FN = 0, TN = 2, FP = 0</p>
<p>Expert Judgment: one pole object and two non-pole objects (left and right tree)</p>	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 2; two non-pole objects detected as non-pole (left and right tree). - FP = 0; no non-pole object detected as pole. 	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 2; two non-pole objects detected as non-pole (left and right tree). - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Conclusion derived from the Simulation :</p>		
<p>- Both Algorithms (YurHoS and Non-YuRHoS) have the same capabilities in detecting poles.</p>		
		
	<p>TP = 1, FN = 0, TN = 2, FP = 3</p>	<p>TP = 1, FN = 0, TN = 5, FP = 0</p>
<p>Expert Judgment: one pole object and five non-pole objects (three street signs and two trees)</p>	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 2; two non-pole objects detected as non-pole (two trees) 	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 5; five non-pole objects detected as non-pole (three street names and two trees)

	- FP = 3; three non-pole objects detected as pole (three street signs).	- FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0,4$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,5$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Cocnclusion from the Simulation :</p> <ul style="list-style-type: none"> - YuRHoS Algorithm has a better specificity (score 1) compared to Non-YuRHoS Algorithm (score 0,4). YuRHoS Algorithm is capable of detecting the correct result (pole) and incorrect ones (non-pole). - YuRHoS Algorithm is better in accuracy (score 1) in comparison to Non-YuRHoS Algorithm (score 0,5). The YuRHoS Algorithm is more capable of detecting the correct result (pole) out of all the objects detected. 		
		
	TP = 1, FN = 0, TN = 0, FP = 3	TP = 1, FN = 0, TN = 3, FP = 0
Expert Judgment: one pole object and three non-pole objects (IHOP pole, tree and a house)	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 0; no non-pole object detected as on-pole. - FP = 3; three non-pole objects detected as pole (IHOP pole, tree and a house). 	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 3; three non-pole objects detected as non-pole (IHOP pole, tree and a house) - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,33$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Conclusions derived from the Simulation :</p> <ul style="list-style-type: none"> - YuRHoS Algorithm has a better specificity (score 1) compared to Non-YuRHoS Algorithm (score 0). YuRHoS Algorithm is capable of detecting the correct result (pole) and incorrect ones (non-pole). - YuRHoS Algorithm is better in accuracy (score 1) in comparison to Non-YuRHoS Algorithm (score 0,33). YuRHoS Algorithm is more capable of detecting the correct result (pole) out of all the objects detected. 		




		
	TP = 1, FN = 0, TN = 2, FP = 1	TP = 1, FN = 0, TN = 3, FP = 0
Expert Judgment: one pole object and three non-pole objects (one round lamp, grass and window)	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 2; two non-pole objects detected as non-pole (round lamp and grass) - FP = 1; one non-pole object detected as pole (window). 	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 0; no pole not detected as pole. - TN = 3; three non-pole objects detected as non-pole (one round lamp, grass and window) - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{SpC} = \frac{TN}{TN+FP} = 0,67$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,75$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{SpC} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Conclusions derived from the Simulation :</p> <ul style="list-style-type: none"> - YuRHoS Algorithm has a better specificity (score 1) compared to Non-YuRHoS Algorithm (score 0,67). YuRHoS Algorithm is capable of detecting the correct result (pole) and the incorrect ones (non-pole). - YuRHoS Algorithm is better in accuracy (score 1) in comparison to Non-YuRHoS Algorithm (score 0,75). YuRHoS Algorithm is more capable of detecting the correct result (pole) out of all the objects detected. 		

Figure 5.11 The Result of Image Simulation of Single Poles

The conclusions derived from the simulation of single pole detection, using Non-YuRHoS and YuRHoS Algorithms, are presented in Table 5.2 as follow.

Table 5.2 Conclusions derived from the Simulation of Single Poles

Image Examined	Parameter	Non-YuRHoS	YuRHoS
1	<i>sensitivity</i>	1	1
	<i>specificity</i>	0	1
	<i>accuracy</i>	0,5	1

2	<i>sensitivity</i>	1	1
	<i>specificity</i>	0,75	1
	<i>accuracy</i>	0,8	1
3	<i>sensitivity</i>	1	1
	<i>specificity</i>	1	1
	<i>accuracy</i>	1	1
4	<i>sensitivity</i>	1	1
	<i>specificity</i>	0	1
	<i>accuracy</i>	0,5	1
5	<i>sensitivity</i>	1	1
	<i>specificity</i>	0,5	0,5
	<i>accuracy</i>	0,67	0,67
6	<i>sensitivity</i>	1	1
	<i>specificity</i>	0,5	1
	<i>accuracy</i>	0,67	1
7	<i>sensitivity</i>	1	1
	<i>specificity</i>	1	1
	<i>accuracy</i>	1	1
8	<i>sensitivity</i>	1	1
	<i>specificity</i>	0,4	1
	<i>accuracy</i>	0,5	1
9	<i>sensitivity</i>	1	1
	<i>specificity</i>	0	1
	<i>accuracy</i>	0,33	1
10	<i>sensitivity</i>	1	1
	<i>specificity</i>	0,67	1
	<i>accuracy</i>	0,75	1

From the general conclusions obtained from ten pole images simulated in Table 5.2, it is concluded that pole-detection using YuRHoS Algorithm is better compared to Non YuRHoS Algorithm in two aspects of measurement, specificity and accuracy. The superiority in specificity indicates that YuRHoS Algorithm is better in detecting pole objects and is capable of differentiating non-pole objects (non-poles). The superiority of accuracy indicates that YuRHoS Algorithm is more accurate in detecting pole objects compared to Non YuRHoS Algorithm. Whereas, for sensitivity aspect, both algorithms have equal superiorities in predicting pole objects correctly.

Next simulations were executed for multi poles. Figure 5.12 showed ten multi poles which will be simulated using two different algorithms (Non-YuRHoS and YuRHoS).



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)


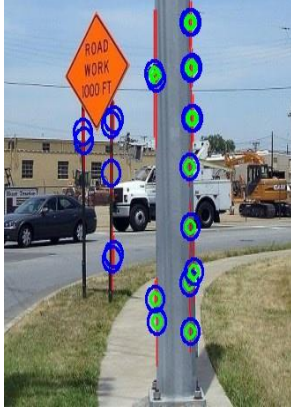




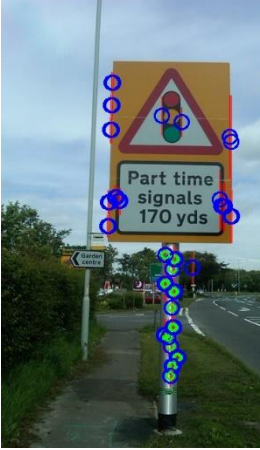




(j)




Figure 5.12 Image of Multi-Pole Simulation







The simulation results of multi-poles using pole-detection algorithm of Non-YuRHoS and YuRHoS is presented in Figure 5.13 as follow:




Image Examined	Result for Non YuRHoS	Result for YuRHoS
	TP = 2, FN = 0, TN = 3, FP = 1	TP = 2, FN = 0, TN = 4, FP = 0
Expert Judgment: two pole objects and four non-pole objects (one car, one street sign, one person and a tree)	<ul style="list-style-type: none"> - TP = 2; two poles detected as poles. - FN = 0; no pole not detected as pole. - TN = 3; three non-pole objects detected as non-pole. - FP = 1; one non-pole object detected as pole (street sign). 	<ul style="list-style-type: none"> - TP = 2; two poles detected as poles. - FN = 0; no pole not detected as pole. - TN = 4; four non-pole objects detected as non-pole. - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{SpC} = \frac{TN}{TN+FP} = 0,75$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,83$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{SpC} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$


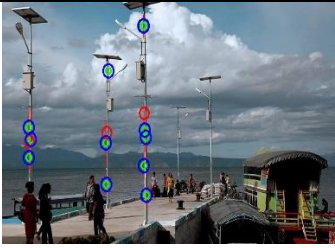
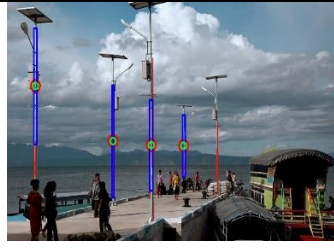

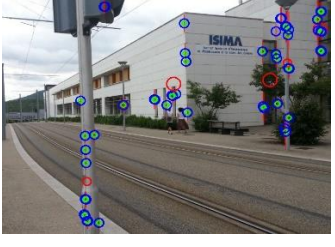

<p>Conclusions derived from the Simulation :</p> <ul style="list-style-type: none"> - YuRHoS Algorithm has a better specificity (score 1) compared to Non-YuRHoS Algorithm (score 0,75). YuRHoS Algorithm is capable of detecting the correct result (pole) and incorrect ones (non-pole). - YuRHoS Algorithm is better in accuracy (score 1) in comparison to Non-YuRHoS Algorithm (score 0,83). YuRHoS Algorithm is more capable of detecting the correct result (pole) out of all the objects detected. 		
		
	<p>TP = 3, FN = 0, TN = 4, FP = 0</p>	<p>TP = 3, FN = 0, TN = 4, FP = 0</p>
<p>Expert Judgment: three pole objects and four non-pole objects (one street sign and three cars)</p>	<ul style="list-style-type: none"> - TP = 3; three poles detected as pole. - FN = 0; no pole not detected as pole. - TN = 4; four non-pole objects detected as non-pole. - FP = 0; no non-pole object detected as pole. 	<ul style="list-style-type: none"> - TP = 3; three poles detected as poles. - FN = 0; no pole not detected as pole. - TN = 4; four non-pole objects detected as non-pole. - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Conclusion derived from the Simulation :</p> <ul style="list-style-type: none"> - Both Algorithms (YurHoS and Non-YuRHoS) have the same capabilities in detecting pole. 		

		
<p>Expert Judgment: two pole objects and three non-pole objects (two street signs and a tree)</p>	<p>TP = 1, FN = 1, TN = 2, FP = 1</p> <ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 1; one pole not detected as pole. - TN = 2; two non-pole objects detected as pole (one street sign and a tree). - FP = 1; one non-pole object detected as pole (front street sign). 	<p>TP = 2, FN = 0, TN = 2, FP = 1</p> <ul style="list-style-type: none"> - TP = 2; two poles detected as pole. - FN = 0; no pole not detected as pole. - TN = 2; two non-pole objects detected as non-pole (one street sign and a tree). - FP = 1; one non-pole object detected as pole (front street sign).
	$\text{Sen} = \frac{TP}{TP+FN} = 0,5$ $\text{Spc} = \frac{TN}{TN+FP} = 0,67$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,6$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0,67$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,8$
<p>Conclusions derived from the Simulation :</p> <ul style="list-style-type: none"> - YuRHoS Algorithm has a better sensitivity (score 1) compared to Non-YuRHoS Algorithm (score 0,5). YuRHoS Algorithm is more capable of detecting the correct result (pole). - YuRHoS Algorithm is better in accuracy (score 0,8) in comparison to Non-YuRHoS Algorithm (score 0,6). YuRHoS Algorithm is more capable in detecting the correct result (pole) out of all the objects detected. 		
		
<p>Expert Judgment: three pole objects and five non-pole</p>	<p>TP = 3, FN = 0, TN = 3, FP = 2</p> <ul style="list-style-type: none"> - TP = 3; three poles detected as poles. 	<p>TP = 3, FN = 0, TN = 4, FP = 1</p> <ul style="list-style-type: none"> - TP = 3; three poles detected as poles.

objects (three trees, one street signs, one building)	<ul style="list-style-type: none"> - FN = 0; no pole not detected as pole. - TN = 3; three non-pole objects detected as non-poles (three trees). - FP = 2; two non-pole objects detected as poles (street sign and a building). 	<ul style="list-style-type: none"> - FN = 0; no pole not detected as pole. - TN = 4; four non-pole objects detected as non-poles (three trees and one building). - FP = 1; one non-pole object detected as pole (street sign).
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0,6$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,75$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0,8$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,88$
<p>Conclusions derived from the Simulation :</p> <ul style="list-style-type: none"> - YuRHoS Algorithm ghas a better specificity (score 0,8) compared to Non-YuRHoS Algorithm (score 0,6). YuRHoS Algorithm is capable of detecting the correct result (pole) and incorrect ones (non-pole). - YuRHoS Algorithm is better in accuracy (score 0,88) in comparison to Non-YuRHoS Algorithm (score 0,75). YuRHoS is more capable of detecting the correct result (pole) out of all the objects detected. 		
		
	TP = 1, FN = 2, TN = 2, FP = 1	TP = 2, FN = 1, TN = 3, FP = 0
Expert Judgment: three pole objects and three non-pole objects (one car and motorcycle, one traffic light, a wall)	<ul style="list-style-type: none"> - TP = 1; one pole detected as pole. - FN = 2; two poles not detected as poles. - TN = 2; two non-pole objects detected as non-pole (one car and motorcycle, one traffic light). - FP = 1; one non-pole object detected as pole (wall). 	<ul style="list-style-type: none"> - TP = 2; two poles detected as poles. - FN = 1; one pole not detected as pole. - TN = 3; three non-pole objects detected as non-pole (one car and motorcycle, one traffic light, wall). - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 0,33$ $\text{Spc} = \frac{TN}{TN+FP} = 0,67$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,5$	$\text{Sen} = \frac{TP}{TP+FN} = 0,67$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,83$
<p>Conclusion derived from the Simulation :</p> <ul style="list-style-type: none"> - YuRHoS Algorithm has better sensitivity (score 0,67) compared to Non-YuRHoS Algorithm (score 0,33). YuRHoS Algorithm is more capable of detecting the correct result (pole). - YuRHoS Algorithm has a better specificity (score 1) compared to Non-YuRHoS Algorithm (score 0,67). YuRHoS Algorithm is capable of detecting the correct result (pole) and incorrect ones (non-pole). 		

<p>- YuRHoS Algorithm is better in accuracy (score 0,83) in comparison to Non-YuRHoS Algorithm (score 0,5). YuRHoS Algorithm is more capable of detecting the correct result (pole) out of all the objects detected.</p>		
		
	<p>TP = 3, FN = 0, TN = 4, FP = 1</p>	<p>TP = 3, FN = 0, TN = 5, FP = 0</p>
<p>Expert Judgment: three pole objects and five non-pole objects (one left tree, one car and three right trees)</p>	<ul style="list-style-type: none"> - TP = 3; three poles detected as poles. - FN = 0; no pole not detected as pole. - TN = 4; four non-pole objects detected as non-pole (left tree, one car and two tres on the right). - FP = 1; one non-pole object detected as pole (right tree). 	<ul style="list-style-type: none"> - TP = 3; three poles detected as poles. - FN = 0; no pole not detected as pole. - TN = 5; five non-pole objects detected as non-pole (one left tree, one car and three trees on the right). - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0,8$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,9$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Conclusions derived from the Simulation:</p> <ul style="list-style-type: none"> - YuRHoS Algorithm has a better specificity (score 1) compared to Non-YuRHoS Algorithm (score 0,8). YuRHoS Algorithm is capable of detecting the correct result (pole) and incorrect ones (non-pole). - YuRHoS Algorithm is better in accuracy (score 1) in comparison to Non-YuRHoS Algorithm (score 0,9). YuRHoS Algorithm I smore capable of detecting the correct result (pole) out of all the objects detected. 		
		
	<p>TP = 2, FN = 1, TN = 0, FP = 1</p>	<p>TP = 3, FN = 0, TN = 1, FP = 0</p>
<p>Expert Judgment: three pole objects and one non-pole object (road)</p>	<ul style="list-style-type: none"> - TP = 2; two poles detected as poles. - FN = 1; one pole not detected as pole. - TN = 0; no non-pole object detected as non-pole. 	<ul style="list-style-type: none"> - TP = 3; three poles detected as poles. - FN = 0; no pole not detected as pole. - TN = 1; one non-pole object detected as pole (road).

	- FP = 1; one non-pole object detected as pole (road)	- FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 0,67$ $\text{Spc} = \frac{TN}{TN+FP} = 0$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,5$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Conclusions derived from the Simulation :</p> <ul style="list-style-type: none"> - YuRHoS Algorithm has a better sensitivity (score 1) compared to Non-YuRHoS Algorithm (score 0,67). YuRHoS Algorithm is more capable of detecting the correct result (pole). - YuRHoS Algorithm has a better specificity (score 1) compared to Non-YuRHoS Algorithm (score 0). YuRHoS Algorithm is capable in detecting the correct result (pole) and incorrect ones (non-pole). - YuRHoS Algorithm is better in accuracy (score 1) in comparison to Non-YuRHoS Algorithm (score 0,5). YuRHoS Algorithm is more capable of detecting the correct result (pole) out of all the objects detected. 		
		
	TP = 3, FN = 0, TN = 2, FP = 0	TP = 3, FN = 0, TN = 2, FP = 0
Expert Judgment: three pole objects and two non-pole objects (tree and a tent)	<ul style="list-style-type: none"> - TP = 3; three poles detected as poles. - FN = 0; no pole not detected as pole. - TN = 2; two non-pole objects detected as non-pole (tree and a tent). - FP = 0; no non-pole object detected as pole. 	<ul style="list-style-type: none"> - TP = 3; three poles detected as poles. - FN = 0; no pole not detected as pole. - TN = 2; two non-pole objects detected as non-pole (tree and a tent). - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$	$\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Conclusion derived from the Simulation :</p> <ul style="list-style-type: none"> - Both Algorithms (YurHoS and Non-YuRHoS) have the same capabilities in detecting poles. 		

		
	TP = 3, FN = 2, TN = 6, FP = 0	TP = 4, FN = 1, TN = 6, FP = 0
Expert Judgment: five pole objects and six non-pole objects (four panel boxes, a boat and harbour)	<ul style="list-style-type: none"> - TP = 3; three poles detected as poles. - FN = 2; two poles not detected as poles. - TN = 6; six non-pole objects detected as non-poles (four panel boxes, a boat and harbour). - FP = 0; no non-pole object is detected as pole. 	<ul style="list-style-type: none"> - TP = 4; four poles detected as poles. - FN = 1; one pole not detected as pole. - TN = 6; six non-pole objects detected as non-pole (four panel boxes, a boat and harbour). - FP = 0; no non-pole object detected as pole.
	$\text{Sen} = \frac{TP}{TP+FN} = 0,6$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,8$	$\text{Sen} = \frac{TP}{TP+FN} = 0,8$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,9$
Conclusions derived from the Simulation :		
<ul style="list-style-type: none"> - YuRHoS Algorithm has a better sensitivity (score 0,8) compared to Non-YuRHoS Algorithm (score 0,6). YuRHoS Algorithm is more capable of detecting the correct result (pole). - YuRHoS Algorithm is better in accuracy (score 0,9) in comparison to Non-YuRHoS Algorithm (score 0,8). YuRHoS Algorithm is more capable of detecting the correct result (pole) out of all the objects detected. 		
		
	TP = 3, FN = 2, TN = 1, FP = 3	TP = 3, FN = 2, TN = 2, FP = 2
Expert Judgment: five pole objects and four non-pole objects (bench, window 1, window 2, a building)	<ul style="list-style-type: none"> - TP = 3; three poles detected as poles. - FN = 2; two poles not detected as poles. - TN = 1; one non-pole object detected as non-pole (bench). - FP = 3; three non-pole objects detected as poles (window 1, window 2 and a building) 	<ul style="list-style-type: none"> - TP = 3; three poles detected as poles. - FN = 2; two poles not detected as poles. - TN = 2; two non-pole objects detected as non-pole (a bench and window 2). - FP = 2; two non-pole objects detected as poles (a building and window 1)

	$\text{Sen} = \frac{TP}{TP+FN} = 0,6$ $\text{Spc} = \frac{TN}{TN+FP} = 0,25$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,44$	$\text{Sen} = \frac{TP}{TP+FN} = 0,6$ $\text{Spc} = \frac{TN}{TN+FP} = 0,5$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,56$
<p>Conclusions derived from the Simulation :</p> <ul style="list-style-type: none"> - YuRHoS Algorithm has a better specificity (score 0,5) compared to Non-YuRHoS algorithm (score 0,25). YuRHoS Algorithm is capable of detecting the correct result (pole) and incorrect ones (non-pole). - YuRHoS Algorithm is better in accuracy (score 0,56) in comparison to Non-YuRHoS Algorithm (score 0,44). YuRHoS Algorithm is more capable of detecting the correct result (pole) out of all the objects detected. 		

Figure 5.13 The Result of Image Simulation of Multi-Poles

The conclusions derived from the simulation of multi-poles detection, using Non-YuRHoS and YuRHoS Algorithms, are presented in Table 5.3 as follow.

Table 5.3 Conclusions derived from the Simulation of Multi- Poles

Image Examined	Parameter	Non-YuRHoS	YuRHoS
1	<i>sensitivity</i>	1	1
	<i>specificity</i>	0,75	1
	<i>accuracy</i>	0,83	1
2	<i>sensitivity</i>	1	1
	<i>specificity</i>	1	1
	<i>accuracy</i>	1	1
3	<i>sensitivity</i>	0,5	1
	<i>specificity</i>	0,5	0,5
	<i>accuracy</i>	0,5	0,67
4	<i>sensitivity</i>	1	1
	<i>specificity</i>	0,6	0,8
	<i>accuracy</i>	0,75	0,88
5	<i>sensitivity</i>	0,33	0,67
	<i>specificity</i>	0,67	1
	<i>accuracy</i>	0,5	0,83
6	<i>sensitivity</i>	1	1
	<i>specificity</i>	0,8	1
	<i>accuracy</i>	0,9	1
7	<i>sensitivity</i>	0,67	1
	<i>specificity</i>	0	1
	<i>accuracy</i>	0,5	1

8	<i>sensitivity</i>	1	1
	<i>specificity</i>	1	1
	<i>accuracy</i>	1	1
9	<i>sensitivity</i>	0,6	0,8
	<i>specificity</i>	1	1
	<i>accuracy</i>	0,8	0,9
10	<i>sensitivity</i>	0,6	0,6
	<i>specificity</i>	0,25	0,5
	<i>accuracy</i>	0,44	0,56


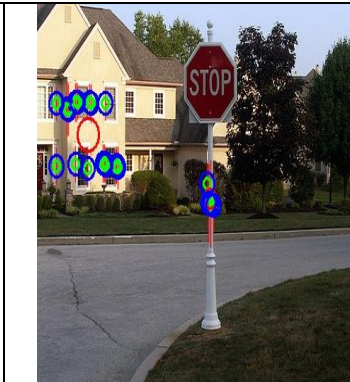


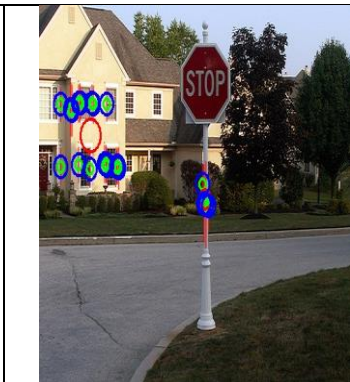

From the general conclusions obtained from ten pole images simulated in Table 5.3, it is concluded that pole-detection using YuRHoS Algorithm is better compared to Non YuRHoS Algorithm in three aspects of measurement, sensitivity, specificity and accuracy. The superiority in specificity indicates that YuRHoS Algorithm is better in detecting pole objects and is capable of differentiating non-pole objects (non-poles). The superiority of accuracy indicates that YuRHoS Algorithm is more accurate in detecting pole objects compared to Non YuRHoS Algorithm. Whereas, for sensitivity aspect, in several simulations, YuRHoS Algorithm is better in predicting pole objects correctly.

d. Simulations with different Expert Judgment

In the process of classifying/labeling images examined, different expert judgments are often encountered. Whenever differences in classifications among experts are found when deciding compositions of pole objects and non-pole objects, a consensus must be reached for a single decision [95].

In this research, differences among expert judgments do not implicate significant differences in the process of algorithm simulation in pole-detection (the result is relatively the same). Below are examples of differences in expert judgments in labeling images of single pole examined, as well as the results (Figure 5.14).

Expert Judgment (1)

		
<p>Expert Judgment-1: one pole object and two non-pole objects (STOP sign and a house)</p>	<p>TP = 1, FN = 0, TN = 1, FP = 1</p> $\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0,5$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,67$	<p>TP = 1, FN = 0, TN = 2, FP = 0</p> $\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Expert Judgment (2)</p>		
		
<p>Expert Judgment-2: one pole object and three non-pole objects (STOP sign and house to left and right)</p>	<p>TP = 1, FN = 0, TN = 2, FP = 1</p> $\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 0,67$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 0,75$	<p>TP = 1, FN = 0, TN = 3, FP = 0</p> $\text{Sen} = \frac{TP}{TP+FN} = 1$ $\text{Spc} = \frac{TN}{TN+FP} = 1$ $\text{Acc} = \frac{TP+TN}{TP+FP+TN+FN} = 1$
<p>Expert Judgment (3)</p>		


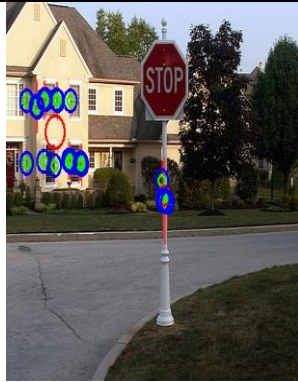

		
Expert Judgment-3: one pole object and four non-pole objects (STOP sign, house to the left, house to the right and front yard)	$TP = 1, FN = 0, TN = 2, FP = 2$ $Sen = \frac{TP}{TP+FN} = 1$ $Spc = \frac{TN}{TN+FP} = 0,5$ $Acc = \frac{TP+TN}{TP+FP+TN+FN} = 0,6$	$TP = 1, FN = 0, TN = 4, FP = 0$ $Sen = \frac{TP}{TP+FN} = 1$ $Spc = \frac{TN}{TN+FP} = 1$ $Acc = \frac{TP+TN}{TP+FP+TN+FN} = 1$

Figure 5.14 The Result of Single-Pole Simulation from a different Expert Judgment

From the general conclusions obtained from the simulation in Figure 5.14, among the three expert judgments, it can be concluded that the result from the pole detection using YuRHoS Algorithm is better compared to Non YuRHoS Algorithm. YuRHoS Algorithm in pole-detection is better in specificity and accuracy compared to Non-YuRHoS. The conclusions from the result of a single-pole detection taken from several expert judgments are shown in Table 5.4 as follow.

Table 5.4 Conclusions from The Result of a Single-Pole Detection of Several Expert Judgments

Expert Judgment	Parameter	Non-YuRHoS	YuRHoS
1	<i>sensitivity</i>	1	1
	<i>specificity</i>	0,5	1
	<i>accuracy</i>	0,67	1
2	<i>sensitivity</i>	1	1
	<i>specificity</i>	0,67	1
	<i>accuracy</i>	0,75	1
3	<i>sensitivity</i>	1	1
	<i>specificity</i>	0,5	1
	<i>accuracy</i>	0,6	1

Next are examples of the differences of expert judgments in labeling multi-pole-detection images and the results obtained (Figure 5.15).










Expert Judgment (1)		
		
Expert Judgment: three pole objects and one non-pole object (wall)	$TP = 1, FN = 2, TN = 0, FP = 1$ $Sen = \frac{TP}{TP+FN} = 0,33$ $Spc = \frac{TN}{TN+FP} = 0$ $Acc = \frac{TP+TN}{TP+FP+TN+FN} = 0,25$	$TP = 2, FN = 1, TN = 1, FP = 0$ $Sen = \frac{TP}{TP+FN} = 0,67$ $Spc = \frac{TN}{TN+FP} = 1$ $Acc = \frac{TP+TN}{TP+FP+TN+FN} = 0,75$
Expert Judgment (2)		
		
Expert Judgment: three pole objects and two non-pole objects (traffic light, a car and motorcycle)	$TP = 1, FN = 2, TN = 2, FP = 0$ $Sen = \frac{TP}{TP+FN} = 0,33$ $Spc = \frac{TN}{TN+FP} = 1$ $Acc = \frac{TP+TN}{TP+FP+TN+FN} = 0,6$	$TP = 2, FN = 1, TN = 2, FP = 0$ $Sen = \frac{TP}{TP+FN} = 0,67$ $Spc = \frac{TN}{TN+FP} = 1$ $Acc = \frac{TP+TN}{TP+FP+TN+FN} = 0,8$
Expert Judgment (3)		
		
Expert Judgment: three pole objects and three non-pole objects (a car and motorcycle, traffic light, a wall)	$TP = 1, FN = 2, TN = 2, FP = 1$ $Sen = \frac{TP}{TP+FN} = 0,33$ $Spc = \frac{TN}{TN+FP} = 0,67$ $Acc = \frac{TP+TN}{TP+FP+TN+FN} = 0,5$	$TP = 2, FN = 1, TN = 3, FP = 0$ $Sen = \frac{TP}{TP+FN} = 0,67$ $Spc = \frac{TN}{TN+FP} = 1$ $Acc = \frac{TP+TN}{TP+FP+TN+FN} = 0,83$

Figure 5.15 The Result of Multi-Pole Simulation from Several Different Expert Judgments

From the general conclusions obtained from the simulation in Figure 5.15, among the three expert judgments, it can be concluded that the detection result of YuRHoS Algorithm is better than Non_YuRHoS. The pole-detection using YuRHoS Algorithm has a better sensitivity, specificity and accuracy compared to Non_YuRHoS. The conclusions from the result of a multi-pole detection taken from several expert judgments are shown in Table 5.5 as follow.

Table 5.5 Conclusions from The Result of a Multi-Pole Detection of Several Expert Judgments

Expert Judgment	Parameter	Non-YuRHoS	YuRHoS
1	<i>sensitivity</i>	0,33	0,67
	<i>specificity</i>	0	1
	<i>accuracy</i>	0,25	0,75
2	<i>sensitivity</i>	0,33	0,67
	<i>specificity</i>	1	1
	<i>accuracy</i>	0,6	0,8
3	<i>sensitivity</i>	0,33	0,67
	<i>specificity</i>	0,67	1
	<i>accuracy</i>	0,5	0,83

5.4 SEE-Stick Real-Time Test

5.4.1. Specification on Hardware and Criteria of the Test

The object (pole) detection function is a sub-system of the SEES system. The block diagram of the object (pole) detection system to the SEE-stick used in this research is illustrated in the following Figure 5.16.

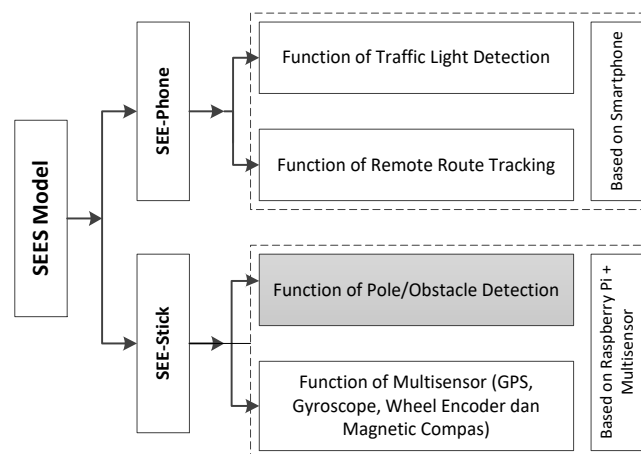


Figure 5.16 The function of the SEE-stick Object (Pole) Detection in the SEE System

In the object pole detection test, the SEE-stick used consisted of the following hardware: 1) Raspberry Pi Model B mini-computer, 2) USB Wi-Fi, 3) *webcam* camera, 4) DC power Regulator: 5 Volt 2,1A. The SEE-stick is designed in the form of 1.5 m long cane which can be adjusted, 45 cm wide and has 3 wheels (2 in front and 1 behind). The SEE-stick hardware is mounted on the SEE-stick cane. Figure 5.17 (a) and (b) are the physical form of the SEE-stick used in the object detection test.



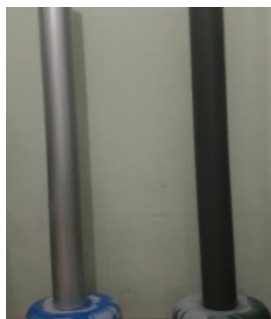
a. SEE-Stick (side view)



b. SEE-Stick (front view)

Figure 5.17 Physical form of the SEE-stick

The pole tested is a 2.5 m long cylindrical pole with a diameter of 3 inches, as can be seen in Figure 5.18:



(a) Two coloured pole



(b) Silver coloured Pole



(c) Black coloured pole

Figure 5.18 Coloured Testing Poles

The program has to process the information of the picture acquired through a webcam camera. In this research, real-time test on the detection of pole uses several models of cameras with different resolutions, the tests are made during day and night time, outdoor and indoor. The criteria of the object (pole) and camera used are as follows (Tables 5.6 dan 5.7):

Table 5.6 Criteria on the shape of the pole

Name	Height	Diameter	Colour	Material
Pole 1	2.5 meter	3 inch	Silver	Plastic (PVC)
Pole 2	2.5 meter	3 inch	Black	Plastic (PVC)

Tabel 5.7 Criteria of the tested Camera

Name	Type	Image Capture	Video Capture	Lens
Webcam 1	Type-1	Up to 1.3 megapixels	640 x 480 pixels	Manual
Webcam 2	Type-2	Up to 3.0 megapixels	1280 x 720 pixels	Manual
Webcam 3	Type-3	Up to 5.0 megapixels	1280 x 720 pixels	Manual
Webcam 4	Type-4	Up to 8.0 megapixels	1280 x 720 pixels	Autofocus

5.5 Result of *Real-Time* Test on Pole Detection

The 1st test is executed outdoor, in the morning and night time using silver and black poles. The 2nd test is executed indoor, morning and night time using silver and black poles. The result of the object (pole) detection is written with fig. 1 if the object is detected and 0 if the object is not detected. The test result is illustrated in Table 5.8.

Table 5.8 Result of the test based on the criteria on pole testing

Camera	Time of Test	Location of Test	Colour of Pole	Distance of Camera to the Post (cm)						Detection Ability \leq 300cm
				50	100	150	200	250	300	
1.3 MP	Daytime	Outdoor	Silver	1	1	1	1	1	0	0
1.3 MP	Daytime	Outdoor	Black	1	1	1	0	0	0	0
1.3 MP	Nighttime	Outdoor	Silver	1	1	1	1	1	0	0
1.3 MP	Nighttime	Outdoor	Black	1	1	1	0	0	0	0
1.3 MP	Daytime	Indoor	Silver	1	1	1	1	0	0	0
1.3 MP	Daytime	Indoor	Black	1	1	1	0	0	0	0

1.3 MP	Nighttime	Indoor	Silver	1	1	1	1	0	0	0
1.3 MP	Nighttime	Indoor	Black	1	1	0	0	0	0	0
3.0 MP	Daytime	Outdoor	Silver	1	1	1	1	1	1	1
3.0 MP	Nighttime	Outdoor	Silver	1	1	1	1	1	1	1
3.0 MP	Nighttime	Outdoor	Black	1	1	1	1	0	0	0
3.0 MP	Daytime	Outdoor	Black	1	1	1	1	0	0	0
3.0 MP	Daytime	Indoor	Silver	1	1	1	1	1	0	0
3.0 MP	Daytime	Indoor	Black	1	1	1	1	0	0	0
3.0 MP	Nighttime	Indoor	Silver	1	1	1	1	0	0	0
3.0 MP	Nighttime	Indoor	Black	1	1	1	0	0	0	0
5.0 MP	Daytime	Outdoor	Silver	1	1	1	1	1	1	1
5.0 MP	Daytime	Outdoor	Black	1	1	1	1	0	0	0
5.0 MP	Nighttime	Outdoor	Silver	1	1	1	1	1	1	1
5.0 MP	Nighttime	Outdoor	Black	1	1	1	1	0	0	0
5.0 MP	Daytime	Indoor	Silver	1	1	1	1	1	1	1
5.0 MP	Daytime	Indoor	Black	1	1	1	1	0	0	0
5.0 MP	Nighttime	Indoor	Silver	1	1	1	1	0	0	0
5.0 MP	Nighttime	Indoor	Black	1	1	1	0	0	0	0
8.0 MP	Daytime	Outdoor	Silver	1	1	1	1	1	1	1
8.0 MP	Daytime	Outdoor	Black	1	1	1	1	1	1	1
8.0 MP	Nighttime	Outdoor	Silver	1	1	1	1	1	1	1
8.0 MP	Nighttime	Outdoor	Black	1	1	1	1	1	1	1
8.0 MP	Daytime	Indoor	Silver	1	1	1	1	1	1	1
8.0 MP	Daytime	Indoor	Black	1	1	1	1	0	0	0
8.0 MP	Nighttime	Indoor	Silver	1	1	1	1	1	0	0
8.0 MP	Nighttime	Indoor	Black	1	1	1	0	0	0	0

The following is the result of the field test on the SEE-stick capability in detecting poles. Figure 5.19 illustrates the result of the test on pole detection on (a) silver coloured pole and (b) black coloured pole. Figure 5.12 illustrates test on undetected pole (a) undetected Silver coloured pole and (b) undetected black coloured pole.

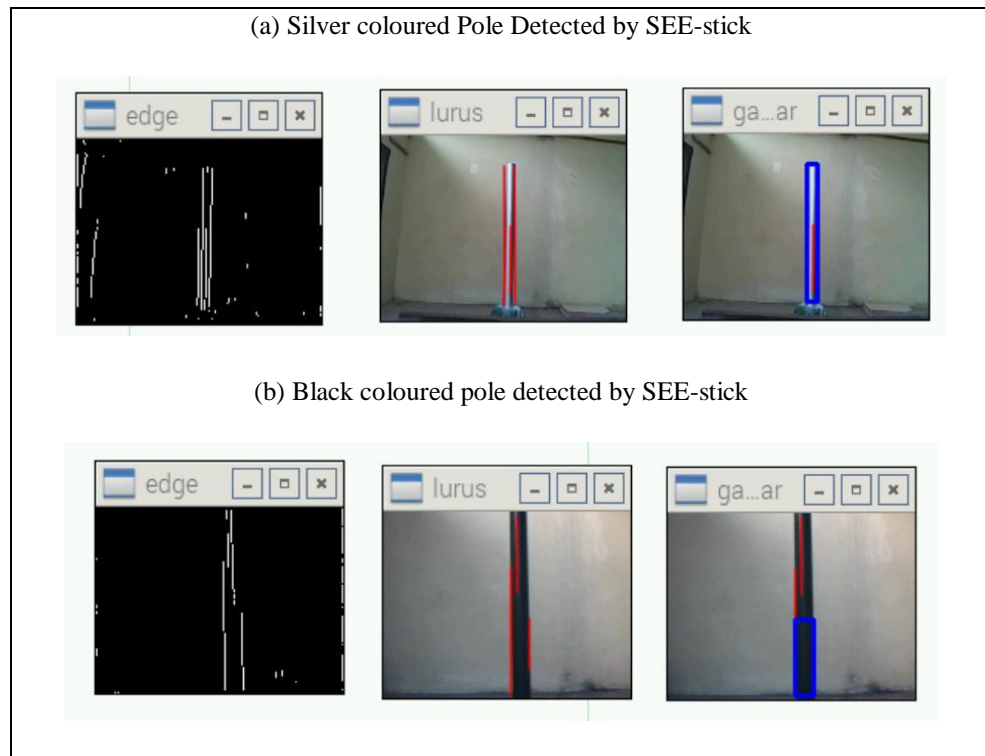


Figure 5.19 Example of poles detected by SEE-stick

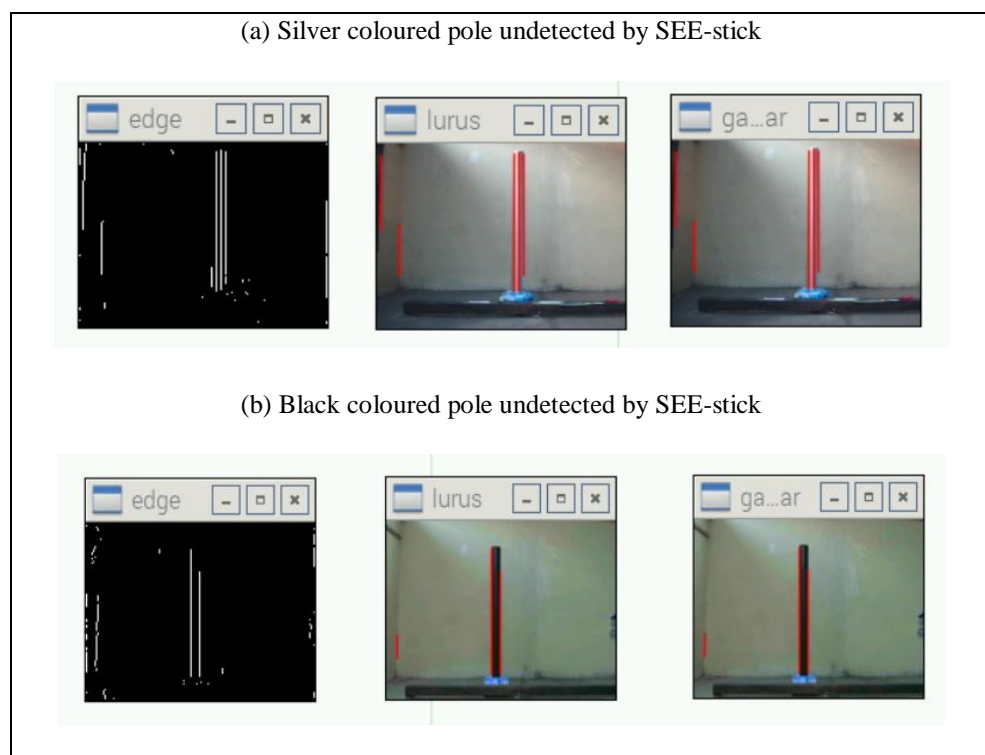


Figure 5.20 Example of poles undetected by SEE-stick

The images in the far left of Figures 5.19 dan 5.20 are the product of edge detection using Canny operation which are optimized until only the vertical lines are detected. This is done by modifying the *Non-Maximum Suppression* (NMS) section, that only 0° group are detected. These vertical lines are detected as the coordinates of vertical lines by using Hough Transformation.

The images in the middle are the result of the process in straightening broken lines vertically. In this process, the broken lines are integrated, marked with red line. The photo in the far right shaped like blue box, is the product of pole detection which indicates that the pole is clearly detected.

5.6 Analysis on Accuracy and Statistical Tests

5.6.1. Analysis on Accuracy Test

The degree of accuracy in the Pole detection test using SEE-stick can be assessed as in the following graphs

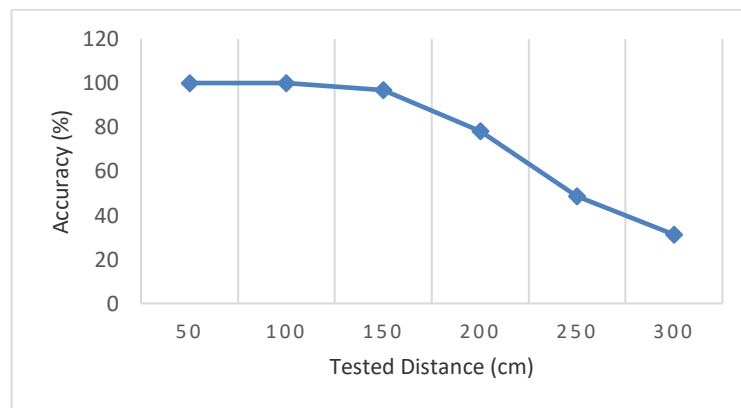


Figure 5.21 Test of Accuracy based on distance

Figure 5.21 shows the graphic on the level of accuracy based on distance between the pole and camera. From the whole trial, it is found that the best accuracy (100%) is when the distance between the object and the camera is between 50 cm – 100 cm. The level of accuracy will then keep decreasing with increase in the distance measurement that 31.25% at a distance of 300 cm.

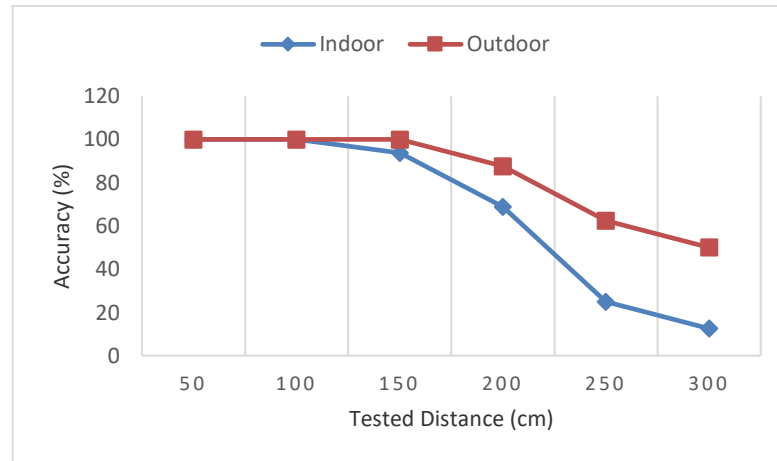


Figure 5.22 Accuracy based on distance in the indoor and outdoor test

Figure 5.22 illustrates the graphic of level of system accuracy based on distance in indoor and outdoor tests. In the whole trial, there is no difference in the level of accuracy at the 50 cm distance in the indoor and outdoor test. With the increase of distance between the camera and pole, the accuracy level is sharply decreased in the indoor test when compared to the outdoor test.

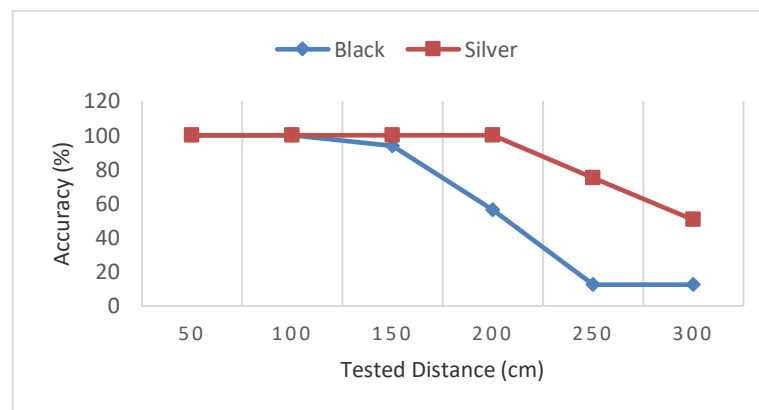


Figure 5.23 Accuracy based on distance in the pole coloured test - Black and Silver

The Figure 5.23 illustrates the graphic of system accuracy level based on the distance to the silver and black coloured poles test. In the whole trial, it was found that there is no difference in accuracy at the distance of 50cm-100cm in the silver and black colour test. However, with the increase in distance between the camera

and poles, there is a sharp decrease in the level of accuracy on the test of black poles when compared to the Silver coloured poles.

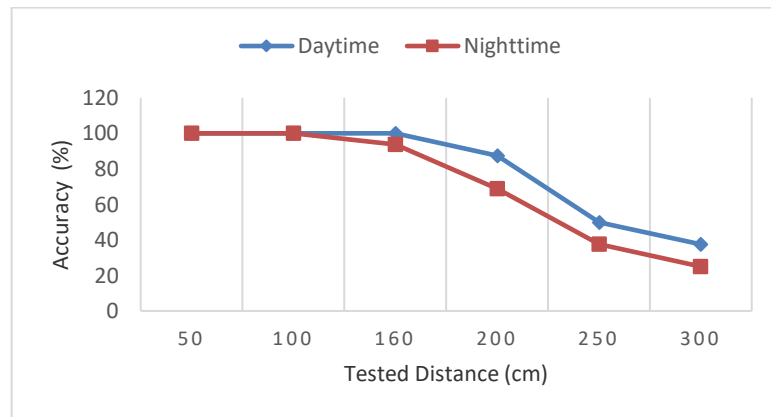


Figure 5.24 Accuracy based on distance to the day and night time test

The Figure 5.24 illustrates the graphic of system accuracy level based on distance in the daytime and nighttime. In the whole trial, there is a decrease of accuracy with every addition of distance. However, the decrease in accuracy in the night time is sharper when compared to the day time.

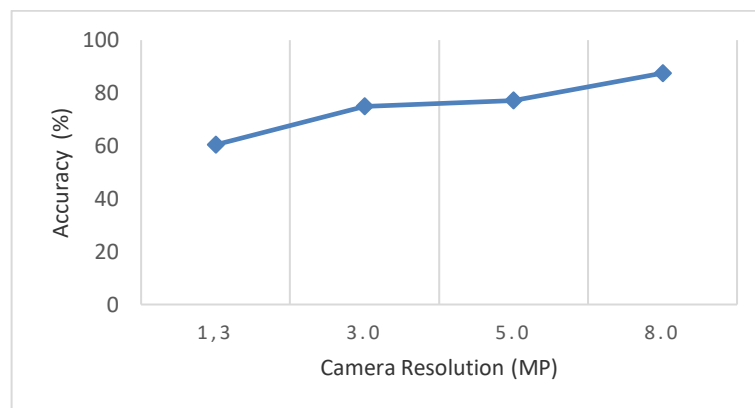


Figure 5.25 Accuracy based on camera resolution

Figure 5.25 illustrates the level of system accuracy based on the camera resolution for all tests. From the graphic, we can see that camera with 8.0 MP resolution has the highest accuracy level in pole detection. The whole test shows that there is an increase in accuracy with the increase in the camera resolution.

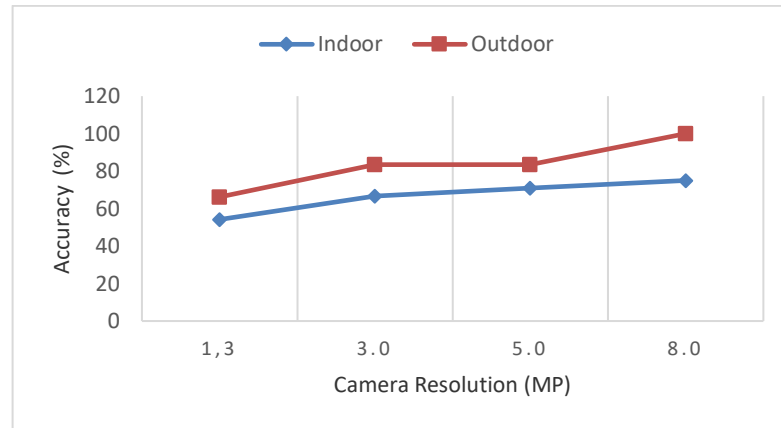


Figure 5.26 Accuracy based on camera resolution in the indoor and outdoor test

Figure 5.26 illustrates the graphic on the system level of accuracy based on camera resolution in the indoor and outdoor test. From the whole trial, we found that the level of accuracy outdoor is better than indoor with the increase in camera resolution.

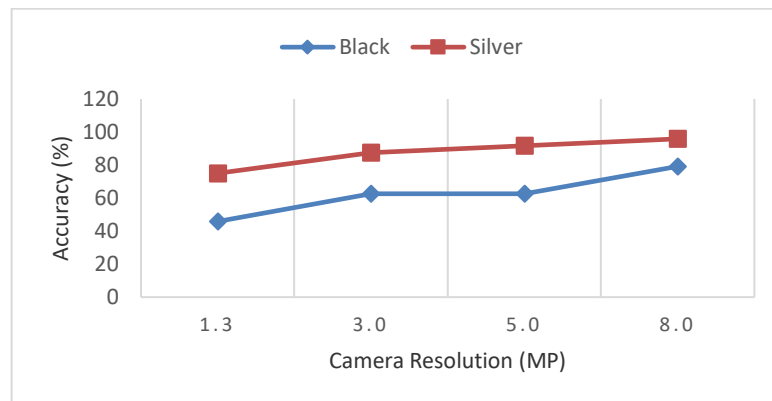


Figure 5.27 Accuracy based on type of camera on the colour of pole (black and silver)

Figure 5.27 illustrates the level of accuracy based on camera resolution to the test on colour of pole (silver and black). From the whole trial, we found that with the increase of resolution, the level of accuracy on the test on silver coloured poles is better than the black coloured poles.

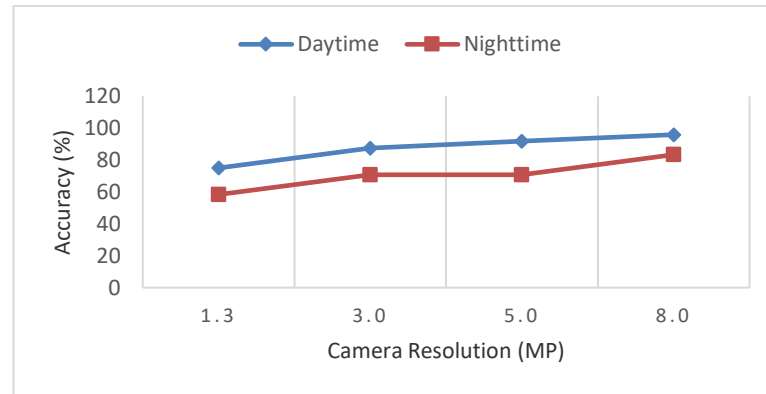


Figure 5.28 Accuracy based on type of camera to the day and night time test

Figure 5.28 illustrates level of accuracy based on camera resolution to the day and night time test. From the graphic, it can be seen that the camera with the resolution at 8.0 MP has the highest level in detecting poles in the day time. The whole test shows that there will be an increase in accuracy with the increase of resolution both in day and night time.

In general, seen from these graphics, the level of accuracy will decrease if the distance of the poles increases and the accuracy will increase if the camera used has a high resolution. In the test comparing indoor and outdoor, it can be seen that the level of accuracy is better in the outdoor test, while in the test in comparing silver and black poles, it can be seen that the silver pole has a better accuracy percentage than the black pole. To understand the variables effect examined, the accuracy rate can be explained by statistical evidence ties.

5.6.2. The Analysis on Time Response on Pole Detection

Researchers tested the ability of the algorithm YuRHoS program in response to the object pole by different resolution levels. In this test, the algorithm is tested by running a program to detect pole based on image pole resolution (black and silver) are taken by the camera. Algorithm programs are executed using *Eclipse* software and measurement of response time by using the system time on the program. Calculation of the response time is done by using the difference in time between when an image is entered as the input program (start time) and then the

image is processed by an algorithm program, up to the time the program finishes detecting the pole on the image (when completed).

The response time = completion time - the start time

By calculating the difference between the end time and the start time, then obtained the response time of the algorithm program. The test results included in Tables 5.9 and 5.10 as well as illustrated by the graph Figure 5.29.

Table 5.9 Response time program algorithms for black pole image

No	Image size (pixels)	Response time detection pole (sec)
1	1600 x 896	1,282
2	2048 x 1536	3,221
3	2560 x 1920	4,844
4	3264 x 2448	6,391
5	4096 x 2304	7,131

Table 5.10 Response time program algorithms for silver pole image

No	Image size (pixels)	Response time detection pole (sec)
1	1600 x 896	1,253
2	2048 x 1536	2,666
3	2560 x 1920	3,750
4	3264 x 2448	5,920
5	4096 x 2304	6,926

Table 5.9 shows the program's algorithm response time in detecting black pole images. It appears that the lower the image resolution of the pole, the faster the response time of the program, vice versa. Table 5.10 shows the program's algorithm response time in detecting silver pole images. Similarly to Table 5.9, the higher the image resolution of the pole, the algorithm response time of the program will be slower in detecting the pole. Figure 5.29 below shows a comparison chart program algorithm response time detecting images of black and silver pole.

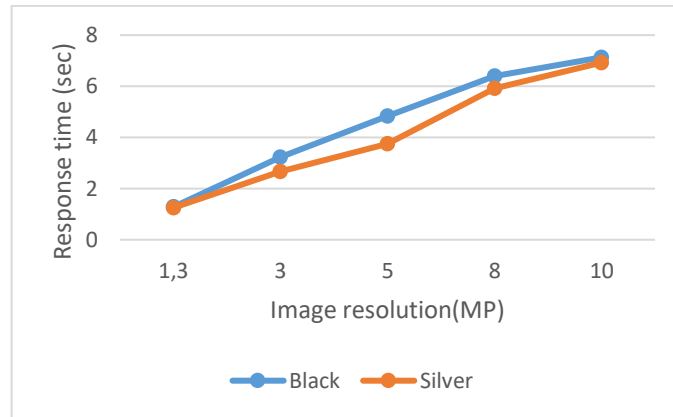


Figure 5.29 The difference in the response time by resolution of the image black and silver poles

In the Figure 5.29 shows that response time of the detection algorithm will require shorter time if using low-resolution images than high-resolution images. The response time for low-resolution images (around 1.3 MP) was 1.282 seconds for black pole and 1,253 seconds for silver pole. As for the high-resolution images (approximately 10.0 MP), the response time is 7.131 seconds for black pole and 6.926 seconds for silver pole. In this test, both the response time difference is not very significant.

From these tests we concluded that the response time of the pole detection algorithm is faster when it detects the low-resolution pole images and vice versa. In addition, the response time of the pole detection algorithm is faster to detect a silver pole compared with the black pole to varying levels of image resolution.

The difference in the response time in the program algorithm detects with different levels of resolution, showing its relationship with complexity. The size of the image resolution relates to the size of the image pixel space, 1 MP (1000 x 1000, for example), 8 MP (4000 x 2000, for example). The algorithm is the same for all the images that were tested ($Y_uRHos, n^2 \log n^2$). So that automatic, higher resolution image will be longer process.

These test results show both silver and black pole has a linear trend increases, increasing image resolution algorithms result in increased processing time consuming. The image resolution is denoted as *Mega Pixel* (MP) represents

the number of pixels contained in an image equivalent to the amount of input data to be processed. So the algorithm will process with varying amounts, for data that the number of large inputs will also be correlated with large amount of time.

5.6.3. Analysis on Statistical Test

Researchers tested the ability of pole detection system (SEE-stick) and perform a statistical test to ensure the variables that affect the ability of the system. The processing of the trial data on the pole detection uses the statistic application software. The data Analysis uses *chi square* and *logistic regression*. In this trial, variables assumed to have an effect on the pole detection ability are: 1) camera resolution (1.3 MP, 3.0 MP, 5.0 MP dan 8.0 MP); 2) testing time (day and night time); 3) location (indoor and outdoor); 4) colour of the pole (Black and Silver).

The criteria of the trial are as follows:

1. The Resolution of the camera is differentiated based on the type of camera used.
2. Day time testing time is between 9.00 am – 11.00 am when the sky is bright and the sun is not covered with cloud, while the night testing time is between pukul 7.00 pm – 9.00 pm with the aid of 25 watts TL lighting.
3. The outdoor location is an open air surrounding while the indoor location is in a room. The location test indoor between daytime and nighttime is assumed the same with the aid of 25 watt TL lighting.
4. Silver coloured pole is a pole which had been painted doff silver colour while the black pole is a pole which has been painted with doff black colour. The shape of the object (pole) tested is a cylindrical pole with a height of 2.5 meter and a diameter of 3 inch.

When the test is being made on the performance of every criteria of the pole test, the research gives a code 1 if the SEE-stick is able to detect the pole and code 0 if the SEE-stick is unable to detect the pole. After 32 testing criterias the researcher categorizes the detection ability that is SEE-stick ability in detecting the location of the post at a distance up to 300 cm. If the system is able to detect the pole at the distance of 50 cm, 100 cm, 150 cm, 200 cm, 250 cm dan 300 cm it will

be categorized as good but if the system is unable to categorize the location of the pole on one of the distance, that it will be categorized as bad. After operating the trial, the research acquired the result as shown in Table 5.11.

Table 5.11 Relation between camera resolution, time, location, colour of object detected up to a distance of ≤ 300 cm

Variable	Object detected up to the distance of ≤ 300 cm				OR (95%CI)	P value
	Bad		Good			
	N	%	N	%		
Camera Resolution						
1.3 MP	8	100	0	0		
3.0 MP	6	75.0	2	25.0	1 (0.104 – 9.614)	1
5.0 MP	5	62.5	3	37.5	1 (1.104 – 9.614)	1
8.0 MP	3	37.5	5	62.5	5 (0.584 – 42.797)	0.142
Time						
Daytime	10	62.5	6	37.5	0.556 (0.122 – 2.536)	0.703
Nighttime	12	75.0	4	25.0		
Location						
Outdoor	8	50.0	8	50.0	0.143 (0.024 – 0.844)	0.057
Indoor	14	87.5	2	12.5		
Colour of Pole						
Black	14	87.5	2	12.5	7 (1.185 – 41.359)	0.057
Silver	8	50.0	8	50.0		

Table 5.11 illustrates that the use camera resolution at 8.0 MP is better in detecting pole when compared with other camera resolution (OR= 5, p value= 0.142). Daytime is better in detecting pole compared to night time (OR = 0.556, p value – 0.703). Indoor test is not as good as outdoor one (OR = 0.143, p value = 0.057). While silver coloured pole is better detected than black pole one (OR= 7, p value= 0.057).

The result of *bivariate* test can be upgraded to *multivariate* test so to find out a more dominant variable. *Cut off point* used is p value < 0.25 . If the variable has p value < 0.25 then the variable is eligible for multivariate test.

The above table shows that there are 3 variables that have p value < 0.25 that is: 1) camera resolution; 2) location; and 3) colour of the pole, and multivariate test are conducted simultaneously. The initial modeling is by removing the highest p value that camera resolution (p value 0.703) and will obtain the same initial modeling result with p value between the location and colour of poles and it is proven that there is a significant effect on the SEE-stick capability variable in detecting the pole. Therefore the initial modelling at the same time becomes the final modelling with the result illustrated in Table 5.12.

Table 5.12 Variable of the multivariate modelling which are connected to the detected object
With a distance of ≤ 300 cm

Variable	OR (95%CI)	P value
Location		
Outdoor	0.085 (0.010-0.699)	0.022
Indoor		
Colour of the Pole		
Black	11.757 (1.430-96.634)	0.022
Silver		

Table 5.12 shows that at a distance of ≤ 300 cm, there are 2 (dua) dominant variables which have an effect on the SEE-stick in detecting the existence of the pole object that is location and pole colour (p value < 0.05). The SEE-Stick is less able to detect pole when test is made indoor at 0.085 times when compared to outdoor test (OR =0.085, p value 0.022), while the SEE-stick has a better detection ability when using of Silver pole by 11.757 times when compared to detecting Black pole.

Poor indoor lighting is the cause for poor detection on pole object by SEE-stick while the result is good when testing outdoor, due to good lighting, especially when the test is conducted day time. The SEE-stick trial in detecting pole is better when using silver pole rather than black pole.

CHAPTER 6

CONCLUSION AND FUTURE RESEARCH

6.1 Conclusion of the Research

The conclusion of the research are as follows:

1. The development of the SEES (*Smart Environment Explorer Stick*) system which integrates the *global remote server* (iSEE), *embedded local server* (SEE-phone) and *smart stick* (SEE-stick) components can be used as assistive aid for VIP's mobility.
2. The development of *YuRHoS* algorithm is able to improve the SEE-stick accuracy in detecting object by measuring the distance and finding vertical line pairs based on optimizing Canny edge detection and Hough Transform.
3. During the test on the *YuRHoS* pole detection algorithm and other pole detection algorithms through simulation test, it is concluded that the *YuRHoS* algorithm is more accurate in detecting pole.
4. The *YuRHoS* algorithm is able to improve the ability of SEE-stick in detecting pole where the level of accuracy of SEE-stick is optimal when operated outdoor and when using a silver coloured pole.
5. The response time of pole detection algorithm will be faster when processing pole images with low resolution compared to high resolution images.
6. The performance of the algorithm indoor test is not as good as outdoor test (OR=0.085, p value = 0.022). The use of silver coloured pole increases further the performance of the algorithm by 11 times when compared to the black pole one (OR = 11.757, p value = 0.022).

6.2 Future Research

To complete this research as well as to focus the image processing image from the SEES system, the researcher must execute the following:

1. Integrating SEES system which is still separate parts, into a complete and integrated system.
2. Validate the operation of the SEES system by testing it directly to visual impaired people (VIP)

3. Develops the detection algorithm by applying *superpixel* and *blobs* technique for image segmentation.
4. Improve the accuracy and speed of the algorithm in detecting various obstacle objects in the field.

LIST OF REFERENCE

- [1] G. A. Stevens, R. A. White, S. R. Flaxman, H. Price, J. B. Jonas, J. Keeffe, J. Leasher, K. Naidoo, K. Pesudovs, S. Resnikoff, and H. Taylor, “Global Prevalence of Vision Impairment and Blindness Magnitude and Temporal Trends , 1990 - 2010,” *Am. Acad. Ophthalmol.*, vol. 120, pp. 2377–2384, 2013.
- [2] World Health Organization, “Media centre Visual impairment and blindness,” *Http://Www.Who.Int/Mediacentre/Factsheets/Fs282/En/#*, 2014. [Online]. Available: <http://www.who.int/mediacentre/factsheets/fs282/en/#>. [Accessed: 01-Jan-2015].
- [3] European Blind Union, “Facts , figures and definitions concerning blindness and sight loss,” 2012. [Online]. Available: <http://www.euroblind.org/resources/information/#details>. [Accessed: 01-Jan-2015].
- [4] A. J. Lafuma, A. P. Brézin, F. L. Fagnani, M. Mesbah, and G. H. Berdeaux, “Prevalence of visual impairment in relation to the number of ophthalmologists in a given area: a nationwide approach.,” *Health Qual. Life Outcomes*, vol. 4, p. 34, 2006.
- [5] Kemenkes RI, “Pusat Data dan Informai KEMENKES RI - Penyandang Disabilitas pada Anak.” Jakarta, 2014.
- [6] R. Velázquez, *Wearable Assistive Devices for the Blind*. Springer International Publishing, 2010.
- [7] G. Gayathri, M. Vishnupriya, R. Nandhini, and M. Banupriya, “Smart Walking Stick for Visually Impaired,” *Int. J. Eng. Comput. Sci.*, vol. 3, no. 3, pp. 4057–4061, 2014.
- [8] M. A.Hersh and M. A. Johnson, “Mobility: An Overview,” in *Assistive Technology for Visually Impaired and Blind People*, Springer London, 2008, pp. 167–208.
- [9] B. Tversky, “Functional Significance of Visuospatial Representations Barbara,” in *The Cambridge Handbook of Visuospatial Thinking*, Cambridge University Press, 2012, pp. 1–34.
- [10] M. Lehtomaki, A. Jaakkola, J. Hyyppa, A. Kukko, and H. Kaartinen, “Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data,” *Remote Sens.*, vol. 2, no. 3, pp. 641–664, 2010.
- [11] H. Yokoyama, H. Date, S. Kanai, and H. Takeda, “Detection and Classification of Pole-like Objects from Mobile Laser Scanning Data of Urban Environments,” *Int. J. CAD/CAM*, vol. 13, no. 2, pp. 31–40, 2013.

- [12] A. Serna and B. Marcotegui, "Detection, segmentation and classification of 3D urban objects using mathematical morphology and supervised learning," *ISPRS J. Photogramm. Remote Sens.*, vol. 93, pp. 243–255, 2014.
- [13] C. Cabo, C. Ordoñez, S. García-Cortés, and J. Martínez, "An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 47–56, 2014.
- [14] C. Wang, T. Jin, M. Yang, and B. Wang, "Robust and Real-Time Traffic Lights Recognition in Complex Urban Environments," *Int. J. Comput. Intell. Syst.*, vol. 4, no. 6, p. 1383, 2011.
- [15] A. K. Khurana, *Comprehensive Ophthalmology*, Fourth. Rohtak, India: New Age International, 2007.
- [16] L. Dandona and R. Dandona, "Revision of visual impairment definitions in the International Statistical Classification of Diseases.," *BMC Med.*, vol. 4, no. 1, p. 7, 2006.
- [17] R. Sihota and R. Tandon, "The Causes and Prevention of Blindness," in *Parson's Diseases of the Eye*, 22nd ed., Elsevier India, 2015, pp. 523–536.
- [18] O. O. Oladayo, "A Multidimensional Walking Aid for Visually Impaired Using Ultrasonic Sensors Network with Voice Guidance," *Int. J. Intell. Syst. Appl.*, vol. 6, no. 8, pp. 53–59, 2014.
- [19] V. Pradeep, G. Medioni, and J. Weiland, "A Wearable System for the Visually Impaired," in *32nd Annual International Conference of the IEEE EMBS*, 2010, pp. 6233–6236.
- [20] X. Wang and L. Wang, "Research on Local Image navigation Method for Visual-impaired person," in *2010 Fifth International Conference on Frontier of Computer Science and Technology*, 2010, pp. 613–618.
- [21] P. Strumillo, "Electronic interfaces aiding the visually impaired in environmental access, mobility and navigation," in *3rd International Conference on Human System Interaction, HSI'2010 - Conference Proceedings*, 2010, pp. 17–24.
- [22] P. Angin, B. Bhargava, and S. Helal, "A mobile-cloud collaborative traffic lights detector for blind navigation," *Proc. - IEEE Int. Conf. Mob. Data Manag.*, pp. 396–401, 2010.
- [23] P. Parette and M. Scherer, "Assistive Technology Use and Stigma," *Educ. Train.*, vol. 39, no. 3, pp. 217–226, 2004.
- [24] M. J. Scherer, "What is Assistive Technology?," *SCI Psychosocial Process*, 2006. [Online]. Available: <http://www.washington.edu/accessit/print.html?ID=1109&FontSize=12&Submit=Submit>.
- [25] WHO, "A Glossary of Terms for Community Health Care and Services for

- Older Persons,” 2004.
- [26] National Center on Accessible Information Technology in Education (AccessIT), “Accessibility of Electronic Tools & Features Used in Distance Learning,” 2004.
- [27] B. R. Beech and D. Roberts, “Assistive technology and older people,” *SCIE Research briefing*, vol. 28, no. August. pp. 1–12, 2008.
- [28] D. Dakopoulos and N. G. Bourbakis, “Wearable obstacle avoidance electronic travel aids for blind: A survey,” in *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 2010, vol. 40, no. 1, pp. 25–35.
- [29] B. . Blasch, W. . Wiener, and R. . Welsh, *Foundations of Orientation and Mobility*, Second. American Foundation for the Blind (AFB) Press, 1997.
- [30] M. Mazo, F. J. Rodriguez, I. Urei, J. C. Garcia, E. Santiso, P. Revenga, J. Jesi, and I. Garcia, “Wheelchair for Physically Disabled People with Voice, Ultrasonic and Infrared Sensor Control,” *Auton. Robots*, vol. 2, pp. 203–224, 1995.
- [31] P. Rohan, G. Ankush, S. Vaibhav, M. Dheeraj, B. M., P. Kolin, and M. Dipendra, “‘Smart’ Cane for The Visually Impaired: Technological Solutions for Detecting Knee-Above Obstacles and Accessing Public Buses,” in *11th International Conference on Mobility and Transport for Elderly and Disabled Persons*, 2007, no. 91.
- [32] R. G. Golledge, J. R. Marston, J. M. Loomis, and R. L. Klatzky, “Stated Preferences for Components of a Personal Guidance System for Nonvisual Navigation,” *J. Vis. Impair. Blind.*, vol. 98, no. 3, pp. 135–147, 2004.
- [33] T. Pallejà, M. Tresanchez, M. Teixidó, and J. Palacin, “Bioinspired electronic white cane implementation based on a LIDAR, a tri-axial accelerometer and a tactile belt,” *Sensors*, vol. 10, no. 12, pp. 11322–11339, 2010.
- [34] D. M. Willough and S. L. Monthei, *Modular Instruction for Independent Travel for Preschool Through High School*. NFB Publisher, 1998.
- [35] R. Farcy, R. Leroux, A. Jucha, R. Damaschini, C. Grégoire, and A. Zogaghi, “Electronic Travel Aids and Electronic Orientation Aids for Blind People: Technical, Rehabilitation and Everyday Life Points of View,” in *Conference & Workshop on Assistive Technologies for People with Vision & Hearing Impairments Technology for Inclusion-CVHI*, 2006.
- [36] D. Mândru, I. Lungu, A. Mociran, and O. Tatar, “Development of a Mechatronic Blind Stick,” *Ann. Oradea Univ. , Fascicle Manag. Technol. Eng.*, vol. VI, no. XVI, pp. 797–802, 2007.
- [37] T. Terlau and W. M. Penrod, “*K” Sonar Curriculum Handbook*. Louisville, Kentucky: American Printing House for the Blind, Inc., 2008.
- [38] I. Ulrich and J. Borenstein, “The GuideCane-applying mobile robot

- technologies to assist the visually impaired,” in *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans.*, 2001, vol. 31, no. 2, pp. 131–136.
- [39] H. Jeremy and B. John, “The Miniguide: a new electronic travel device,” *J. Vis. Impair. Blind.*, vol. 97, no. 10, pp. 655–656, 2003.
- [40] D. Haas, S. Mothersell, D. Nielsen, and K. Yelamarthi, “A Semi-Autonomous Navigational System for the Visually Impaired,” in *ASEE North Central Sectional Conference 2C-1*, 2010, pp. 1–5.
- [41] T.-H. Chang, C.-J. Ho, D. C. Hsu, Y.-H. Lee, M.-S. Tsai, M.-C. Wang, and J. Hsu, “iCane--A partner for the visually impaired,” in *Embedded and Ubiquitous Computing – EUC 2005 Workshops*, Springer Berlin Heidelberg, 2005, pp. 393–402.
- [42] G. Balakrishnan, G. Sainarayanan, R. Nagarajan, and S. Yaacob, “A stereo image processing system for visually impaired,” *Int. J. Comput. Electr. Autom. Control Inf. Eng.*, vol. 2, no. 8, pp. 2794–2803, 2008.
- [43] P. B. Meijer, “An experimental system for auditory image representations,” in *IEEE transactions on bio-medical engineering*, 1992, vol. 39, no. 2, pp. 112–21.
- [44] S. Kammoun, G. Parseihian, O. Gutierrez, A. Brilhault, A. Serpa, M. Raynal, B. Oriola, M. J. M. MacÉ, M. Auvray, M. Denis, S. J. Thorpe, P. Truillet, B. F. G. Katz, and C. Jouffrais, “Navigation and space perception assistance for the visually impaired: The NAVIG project,” *Irbm*, vol. 33, no. 2, pp. 182–189, 2012.
- [45] B. F. G. Katz, S. Kammoun, G. Parseihian, O. Gutierrez, A. Brilhault, M. Auvray, P. Truillet, M. Denis, S. Thorpe, and C. Jouffrais, “NAVIG: Augmented reality guidance system for the visually impaired: Combining object localization, GNSS, and spatial audio,” *Virtual Real.*, vol. 16, no. 4, pp. 253–269, 2012.
- [46] J. Xu, Z. G. Fang, D. H. Dong, and F. Zhou, “An outdoor navigation aid system for the visually impaired,” *Int. Conf. Ind. Eng. Eng. Manag.*, pp. 2435–2439, 2010.
- [47] D.-H. Kim and H.-G. Ryu, “Obstacle recognition system using ultrasonic sensor and Duplex Radio-Frequency Camera for the visually impaired person,” in *13th International Conference on Advanced Communication Technology (ICACT)*, 2011, pp. 326–329.
- [48] J. Llinas and D. L. Hall, “An introduction to multi-sensor data fusion,” in *International Symposium on Circuits and Systems (Cat. No.98CH36187)*, 1998, vol. 6, pp. 537–540.
- [49] A. V Sutagundar and S. S. Manvi, “Context Aware Multisensor Image Fusion for Military Sensor Networks using Multi Agent System,” *Int. J. Ad hoc, Sens. Ubiquitous Comput.*, vol. 2, no. 1, pp. 147–167, 2011.

- [50] J. L. Crowley, "Integration and control of reactive visual processes," *Rob. Auton. Syst.*, vol. 16, no. 1, pp. 17–27, 1995.
- [51] R. C. Gonzalez and R. E. Woods, *Digital Image Processing, 3rd Edition*. New Jersey: Pearson Education, Inc, 2008.
- [52] W. Kong, P. Chang, and Z. Bi, "Real-Time Sobel Edge Detector," in *The 6th PSU-UNS International Conference on Engineering and Technology (ICET-2013)*, 2013, pp. 1–4.
- [53] W. Burger and M. J. Burge, *Digital Image Processing - An Algorithmic Introduction using Java*, First. New York, NY 10013, USA: Springer Science+Business Media, LLC, 2008.
- [54] K. Abdul and S. Adhi, *Image Processing: Theory and Application*, 1st ed. Yogyakarta: Andi Offset, 2013.
- [55] R. C. Gonzalez and R. E. Woods, *Digital image processing, third edition*. Pearson Education, Inc, 2008.
- [56] H. S. Bhadauria, A. Singh, and A. Kumar, "Comparison between Various Edge Detection Methods on Satellite Image," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 6, pp. 2–6, 2013.
- [57] R. Fisher, S. Perkins, a Walker, and E. Wolfart, "Hypermedia image processing reference," pp. 65–66, 1997.
- [58] W. D. W. Dong and Z. S. Z. Shisheng, "Color Image Recognition Method Based on the Prewitt Operator," *2008 Int. Conf. Comput. Sci. Softw. Eng.*, vol. 6, pp. 170–173, 2008.
- [59] J. Y. Zhang, C. Yan, and X. X. Huang, "Edge detection of images based on improved sobel operator and genetic algorithms," *Proc. 2009 Int. Conf. Image Anal. Signal Process. IASP 2009*, no. 3, pp. 32–35, 2009.
- [60] R. Crane, *A Simplified Approach to Image Processing: Classical and Modern Techniques in C*. Prentice Hall PTR Upper Saddle River, NJ, USA, 1996.
- [61] J. Canny, "A Computational Approach to Edge Detection," *IEEE Pattern Anal. Mach. Intell. Trans.*, vol. PAMI-8, no. 6, 1986.
- [62] X. Wang, "Laplacian operator-based edge detectors," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007, vol. 29, no. 5, pp. 886–890.
- [63] F. Z. F. Zhao and C. J. S. DeSilva, "Use of the Laplacian of Gaussian operator in prostate ultrasound image processing," *Proc. 20th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, vol. 20, no. 2, pp. 812–815, 1998.
- [64] G. T. Shrivakshan and C. Chandrasekar, "A Comparison of various Edge Detection Techniques used in Image Processing," *Int. J. Comput. Sci. Issues*, vol. 9, no. 5, pp. 269–276, 2012.

- [65] J. Illingworth and J. Kittler, "A survey of the hough transform," *Comput. Vision, Graph. Image Process.*, vol. 44, no. 1, pp. 87–116, 1988.
- [66] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures.," vol. 15, no. April 1971, pp. 11–15, 1972.
- [67] R. Laganière and R. Lagani, *OpenCV2 Computer Vision Application Programming Cookbook*, no. 5. Birmingham, UK, 2011.
- [68] G. Bradski and A. Kaehler, *Learning OpenCV*, First. Gravenstein Highway North, Sebastopol: O'Reilly Media, Inc, 2008.
- [69] M. Asad and W. Ikram, "Smartphone based guidance system for visually impaired person," *2012 3rd Int. Conf. Image Process. Theory, Tools Appl. IPTA 2012*, pp. 442–447, 2012.
- [70] S. Dutta and B. B. Chaudhuri, "A Color Edge Detection Algorithm in RGB Color Space," in *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, 2009, pp. 337–340.
- [71] Y. Chen, K. M. Hou, H. Zhou, H. L. Shi, X. Liu, X. Diao, H. Ding, J. J. Li, and C. De Vault, "6LoWPAN stacks: A survey," in *7th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2011*, 2011, no. 0, pp. 1–4.
- [72] L. Z. Zhang, K. M. Hou, P. J. Ma, E. Pissaloux, M. Yusro, and H. L. Shi, "Smart Environment Explorer Stick (SEES): concept and design of its orientation and walking sub-system," in *Int Workshop of NICST (New Information Communication Science and Technology for Sustainable Development)*, 2013.
- [73] U. Iqbal, A. F. Okou, and A. Noureldin, "An integrated reduced inertial sensor system - RISS / GPS for land vehicle," *Rec. - IEEE PLANS, Position Locat. Navig. Symp.*, pp. 1014–1021, 2008.
- [74] Q. Ladetto, V. Gabaglio, and B. Merminod, "Combining Gyroscopes , Magnetic Compass and GPS for Pedestrian Navigation," in *International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation (KIS)*, 2001, pp. 205–212.
- [75] J. Y. Shi, "Design and implementation of embedded GPS system," in *Proceedings, 2012 IEEE International Conference on Computer Science and Automation Engineering*, 2012, vol. 1, pp. 311–314.
- [76] L. F. Motiwalla and J. Qin, "Enhancing Mobile Learning Using Speech Recognition Technologies: A Case Study," in *Eighth World Congress on the Management of eBusiness (WCM eB 2007)*, 2007, pp. 18–18.
- [77] H.-D. J. Jeong, S.-K. Ye, J. Lim, I. You, W. Hyun, and H.-K. Song, "A Remote Computer Control System Using Speech Recognition Technologies of Mobile Devices," *2013 Seventh Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput.*, pp. 595–600, 2013.
- [78] I. Zoratti, "MYSQL security best practices," *IET Conf. Crime Secur.*, pp.

183–198, 2006.

- [79] W. M. Chen and Y. C. Chen, “Web design and implementation for remote control,” in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, 2012, pp. 920–924.
- [80] J. Sobota, R. Písl, P. Balda, and M. Schlegel, “Raspberry pi and arduino boards in control education,” *IFAC Proc. Vol.*, vol. 10, no. PART 1, pp. 7–12, 2013.
- [81] Y. Chen, J. Joseph, and A. R. Pi, “Pi-IDS : Evaluation of Open-Source Intrusion Detection Systems on Raspberry Pi 2,” in *Second International Conference on Information Security and Cyber Forensics (InfoSec)*, 2015, pp. 165–170.
- [82] S. Gammeter, a. Gassmann, L. Bossard, T. Quack, and L. Van Gool, “Server-side object recognition and client-side object tracking for mobile augmented reality,” *Comput. Vis. Pattern Recognit. Work. (CVPRW), 2010 IEEE Comput. Soc. Conf.*, no. C, 2010.
- [83] Y.-S. Lee, H.-S. Koo, and C.-S. Jeong, “A straight line detection using principal component analysis,” *Pattern Recognit. Lett.*, vol. 27, no. 14, pp. 1744–1754, 2006.
- [84] S. Wang, H. Pan, C. Zhang, and Y. Tian, “RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs,” *J. Vis. Commun. Image Represent.*, vol. 25, no. 2, pp. 263–272, 2014.
- [85] M. A. Rabbani and C. Chellappan, “A Fast and New Approach to Gradient Edge Detection,” *Int. J. Soft Comput.*, vol. 2, no. 2, pp. 325–330, 2007.
- [86] O. R. Vincent and O. Folorunso, “A Descriptive Algorithm for Sobel Image Edge Detection,” *Proc. Informing Sci. IT Educ. Conf. 2009*, pp. 1–11, 2009.
- [87] M. Litvin and G. Litvin, *Getting Started with Eclipse for Java and GridWorld*. Skylight Publishing, 2011.
- [88] N. Makoto, Y. Nishiyama, C. Henmi, S. Iwanaga, H. Nakagawa, K. Yamaguchi, K. Akita, S. Mochizuki, and K. Takiura, “Ink Jet Three-Dimensional Digital Fabrication for Biological Tissue Manufacturing: Analysis of Alginate Microgel Beads Produced by Ink Jet Droplets for Three Dimensional Tissue Fabrication,” *J. Imaging Sci. Technol.*, vol. 52, no. 1, pp. 1–15, 2008.
- [89] A. Lannoy and H. Procaccia, “Expertise, safety, reliability, and decision making: practical industrial experience,” *Environ. Syst. Decis.*, vol. 34, no. 2, pp. 259–276, 2014.
- [90] T. Fawcett, “An Introduction to ROC Analysis,” *Pattern Recognit.*, vol. 27, pp. 861–874, 2006.
- [91] D. M. W. Powers, “Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation,” *J. Mach. Learn. Technol.*,

- vol. 2, no. 1, pp. 37–63, 2011.
- [92] M. Alsoos and a. Joukhadar, “Posture Independent Model for Hand Detection and Tracking,” *Adv. Intell. Syst. Comput.*, vol. 300, pp. 301–312, 2014.
- [93] P. Przystup, A. Bujnowski, J. Ruminski, and J. Wtorek, “A multisensor detector of a sleep apnea for using at home,” *2013 6th Int. Conf. Hum. Syst. Interact. HSI 2013*, pp. 513–517, 2013.
- [94] I. Giosan and S. Nedevschi, “Superpixel-based obstacle segmentation from dense stereo urban traffic scenarios using intensity, depth and optical flow information,” *17th Int. IEEE Conf. Intell. Transp. Syst.*, pp. 1662–1668, 2014.
- [95] G. B. Moody and R. G. Mark, “The impact of the MIT-BIH arrhythmia database,” *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, 2001.