Blind Navigation in Outdoor Environments: Head and Torso Level Thin-Structure Based Obstacle Detection

By

K.A.T Lakshan

15000753

This dissertation is submitted to the University of Colombo School of Computing
In partial fulfillment of the requirements for the
Degree of Bachelor of Science Honours in Computer Science

University of Colombo School of Computing
35, Reid Avenue, Colombo 07,
Sri Lanka
July 2020

Declaration

I, K.A.T Lakshan 2015/CS/075 hereby certify that this dissertation entitled "Blind Navigation in Outdoor Environments: Head and Torso Level Thin-Structure Based Obstacle Detection" is entirely my own work and it has never been submitted nor is currently been submitted for any other degree.

Date

Student's Signature

I, Dr. G.D.S.P Wimalaratne, certify that I supervised this dissertation entitled "Blind Navigation in Outdoor Environments: Head and Torso Level Thin-Structure Based Obstacle Detection" conducted by K.A.T Lakshan in partial fulfillment of the requirements for the degree of Bachelor of Science Honours in Computer Science.

Date

Supervisor's Signature

Abstract

Blind navigation in computer vision is a highly active research area because independent mobility is one of the essential needs of every human being. Among many blind navigation and obstacle detection systems, researchers have given limited attention to the detection of thin structured wires like obstacles even though blind people can be severely damaged by them. In this research domain, a thin-structured wire is defined as a wire with a maximum of 5mm diameter. For finding a solution for this growing problem of detecting thin-structured wires for blind navigation in outdoor spaces, an assistive system was implemented. The main objective of this research is to address this research gap using computer vision-based techniques.

The proposed approach is based on the Simultaneous Localization and Mapping algorithm(SLAM). SLAM is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of an agent's location within it.The suggested approach takes input as a video stream of the path of the user from a monocular camera and that video stream is processed by the system frame by frame. This process consists of three main stages named image information extraction stage, Tracking stage, and Mapping stage. In the information extraction stage Difference of Gaussian(DoG) based edge detector is used to extract image edges and the outputs of the edge extraction stage are called keylines. Based on these keylines, edge map is created for each frame. Then in the tracking stage, camera motion is tracked by fitting the previous edge map into the new edge map using a warping function. The outcome of this stage is a Special Euclidean group (SE(3)) transformation and it is used as an input to the mapping stage. In the mapping stage, each keyline in the new edge map is matched against the ones in the previous edge map in order to filter real obstacle edges from the noise edges.

Since there are no suitable benchmark datasets for evaluating the method, new datasets were created which consists of wires as obstacles in different environments. The system was evaluated using the Intel core I 3 machine without Graphics Processing Unit(GPU) support. According to the evaluation results, a maximum of 15 frames per second(fps) rate has been achieved with 75% accuracy for wire detection. In conclusion, the proposed algorithmic approach can be considered as a lightweight and accurate solution for the addressed research gap.

Keywords: SLAM, FPS, DoG, Thin-structured wires,

Preface

In this dissertation, novel wires detection algorithm was introduced to address the problem, lack of specifically designed SLAM based algorithms for detection of thin structured wires like obstacles for blind navigation in outdoor environments. State of art wires detection algorithms are based on either machine learning-based or image processing based. Design in Chapter 3 and implementation in Chapter 4 rely upon a semi-dense SLAM system found in the JuanTarrio/rebvo repository on Github. Extending this semi-dense SLAM system to detect wires by introducing an edge linking step and a small component filtering step is the author's work. Wires identifying algorithm which is presented in the implementation chapter is also a completely novel experiment by the authors. These approaches have not been proposed in any other work in the domain of computer vision. The real-world datasets were created to evaluate the proposed approach and evaluation of the proposed method was done by the authors against the state of art SLAM systems.

Acknowledgement

I would like to express my sincere respect and gratitude to my research supervisor, Dr. G.D.S.P Wimalaratne, Senior lecturer of University of Colombo School of Computing. I would also like to extend my respect and gratitude to Dr. Kasun Karunanayake and Dr. K.H.E.L.W Hettiarachchi, Senior lecturers of University of Colombo School of Computing for providing valuable feedback on the research proposal and interim evaluation to improve the results. I would like to express my gratitude for acknowledge the assistance provided by Dr. H. E. M. H. B. Ekanayake, senior lecturer of University of Colombo School of Computing as the final year research project coordinator.

I would also like to thank all of my batch mates for giving comments, motivation and their warm friendship. It is a great pleasure for me to acknowledge all the teachers, mentors and people for the immense support they have given. Finally to my parents and family for their immeasurable sacrifices and love they have given throughout this beautiful journey.

Contents

List of Figures

List of Tables

Acronyms

CNN	Convolution Neural Network
YOLO	You Only Look Once
GPS	Global Positioning System
GIS	Geographical Information System
DoG	Difference of Gaussian
PIC	peripheral Interface Controller
RFID	Radio Frequeny Identification
LDR	Long-Distance Relationship
MonoSLAM	Monocular SLAM
PTAM	Parallel Tracking and Mapping
ORB-SLAM	Oriented FAST and Rotated BRIEF SLAM
SVO	Monocular Visual Odometry
DSO	Direct Sparse Odometry
DTAM	Dense Tracking and Mapping
LSD-SLAM	Large-Scale Direct Monocular SLAM
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
SE	Special Euclidean group
POC	Proof of Concept
REBVO	Real-time Edge-Based Visual Odometry
SGBM	Semi-Global Block Matching
IMU	Inertial Measurement Unit
FPS	Frames Per Second

Chapter 1

Introduction

Globally, it is estimated that approximately 1.3 billion people live with some form of vision impairment. With regard to distance vision, 188.5 million people have a mild vision impairment, 217 million have moderate to severe vision impairment [1], and 36 million people are blind. With regards to near vision, 826 million people live with near vision impairment.Visually impaired people face difficulties in independent mobility and navigation, especially in unknown or dynamic environments. The White cane is one of the conventional navigation aids and a symbol of identification for blind people. There are some limitations have been identified in this mechanism [2] like unable to detect torso and head-level obstacles, other overhanging obstacles, etc.

There have been published a number of researches to assist blind people. Work has been done in the area of positioning, navigation and obstacle identification in both indoor and outdoor environments [3]. There are two main types of navigation systems can be seen in the literature named indoor blind navigation systems and outdoor blind navigation systems.

In indoor navigation systems researchers have used different types of channels to sense the environment. They are infrared sensors, ultrasonic sonar sensors, vision sensors, inertial sensors, etc [4]. An ultrasonic sensor is used as the obstacle detection and range finding the sensor in the literature. Priyadarshana and Wimalaratne [5] presented a mobile phone based personalized wearable obstacle detection approach which used ultrasonic sensors to detect obstacles coming from different directions. Some limitations have been identified of this approach such as main measurement lobe, specular reflections, outliers, crosstalk when using multiple sonar sensors etc [4].Vision sensors have used to identify the obstacles along the path. Kanwal et al [6] presented a blind navigation system using Windowing-Based Mean on Microsoft Kinect Camera. It consumed high computational power during the processing of the images which is a barrier in the real-time navigation approaches. Jiang et al [7] implemented a Real-Time visual Recognition system with 3d audio feedback. In this system, a Convolutional Neural Network(CNN) model called You Only Look Once(YOLO) has been used. Detection failure when objects are too close or too far and Overload of information when the system tries to notify users too many objects have been identified as issues in the system. Kumar and Chourasia [8] presented a blind navigation system using artificial intelligence. They have used a Raspberry Pi camera module to sense the environment. Then, the captured images have been processed by using a CNN image classifier to predict the environment details.

In Outdoor navigation systems, researchers have used Global Positioning System(GPS), Dead reckoning navigations, IR sensors, ultrasonic sensors, Lidar sensors, stereo cameras, monocular cameras, etc [9] to sense the outdoor environment. Velázquez et al [10] have presented a novel, wearable navigation system for visually impaired and blind pedestrians that combines GPS for user outdoor localization and tactile-foot stimulation for information presentation. The OpenStreetMap was used as the Geographic Information System(GIS) of the system. An open-source route planner has used to compute the shortest pedestrian route to a previously chosen destination. One limitation was GPS signal is weak for some areas and GPS locations were not accurate to the exact point. Tian et al [11] have implemented a model based on Microsoft hololens, a powerful head-mounted computer designed for augmented reality for blind navigation. The HoloLens scans all surfaces in the environment using video and infrared sensors, creates a 3D map of the space, and localizes itself within that volume to a precision of a few centimetres. In this research addresses the problem of detecting torso and head level obstacles like thin wires in the outdoor environments.

1.1 Background

Visually impaired people face difficulties in independent mobility and navigation in both indoor and outdoor environments. When it comes to outdoor environments researches have done a large number of researches. There is a lack of research about detecting thin-structure head and torso level hazards such as wires in the literature.

One of the most dangerous accidents visually impaired people face is head-level collisions with low-hanging thin obstacles. A technical report from the University of California, Santa Cruz [12, 13] determined that low-hanging obstacles present a significant risk for the visually impaired. Out of 307 blind or legally blind individuals, 54% reported having head-level accidents once a year or more frequently. Of these accidents, 23% required medical attention and 43% resulted in users changing their walking habits (walking more slowly or raising their arms to protect their heads whenever possible) [12].

Wang and Kuchenbecker [12] have presented an extended version of traditional white which was capable of alert users of low-hanging obstacles. Ultrasound range sensors with vibration feedback have been used. O'Keeffe et al [14] have presented a paper about the characterization result of the Gen1 long-range Lidar sensor. According to the paper, obstacles in the natural environment which are not easily detected or those not detected as early with a white cane were chosen to be detected. Identified limitations were lidar sensors need a huge amount of power and achieving portability will be a challenge. Using sensors like ultrasound, Lidar, IR is not accurate in outdoor environments. Especially these systems are failed when it comes to thin structured obstacles (tree branches, wires, cables)[15].

In this research, the above problem will be addressed using computer vision perspective. Edges will be extracted along the path of a blind person using a video stream and edge 3d reconstruction methodology will be used to calculate the distance to the low hanging obstacles from the user. The proposed framework will be explained in a more detailed manner in the following chapters.

1.2 Research Problem and Research Questions

Among many blind navigations and obstacle detection systems, researchers have given limited attention to the detection of thin structured wires like obstacles even though blind people can be severely damaged [13] by them. In this research domain, a thin-structured wire is defined as a wire with a maximum of 5mm diameter. Finding a solution for this growing problem of detecting thin-structured wires for blind navigation in outdoor spaces can be considered as the problem that is addressed by this research.

1. How to detect thin structured wires to aid blind navigation using Simultaneous Localization and Mapping(SLAM) based approach ?

   As mentioned in the earlier sections researchers have implemented blind navigation systems by following several approaches. Each method has different benefits and drawbacks.Existing edge detecting methodologies are being studied deeply and their limitations will be identified when detecting thin structured wires. Problem of detecting thin structured wires using computer vision-based approach is addressed.

2. How to extract information (depth, camera trajectory) from the environment while maintaining the simplicity of the algorithm ?

Estimating depth using computer vision technologies is a challenging task. Under this research question, accurate depth estimation techniques and camera trajectory finding techniques will be observed. Limitations and benefits of each method will be identified. Based on the results suitable approach will be selected and suitable modifications will be suggested. Here in this research, more concern is to warn blind people about the head level wires like obstacles, Because of that calculating exact distance to the obstacles is not concerned. Since the ultimate goal of this research is to use it in a embedded system, the Simplicity of the suggested system is highly concerned while achieving the highest accuracy.

1.3 Research Aim and Objectives

The ultimate purpose of this research is to develop a framework for the detection of head and torso level hazards like thin structured wires along the way of the blind pedestrians and alert them. The proposed framework will be able to estimate the distance to the hazard from the visually impaired person and alert them using an appropriate methodology. The main objective of this research are:

- Getting a better understanding of the current approaches in order to detection and depth estimation of thin structured wires.
- Understanding the advantages and disadvantages of the different approaches and various types of obstacle detection methods
- Studying obstacle detection approaches that are based on outdoor blind navigation systems
- Studying about thin structured obstacle detection methods based on computer vision technologies
- Exploring depth estimation techniques which are based on computer vision
- Developing a computer vision-based framework for the detection of thin structured obstacles and estimation of depth to these obstacles.
- Exploring methods and techniques which can be used to improve the accuracy of the edge-based obstacle detection methods and edge-based visual odometry methods.

1.4 Justification for the Research

Researches in the field of blind navigation have been popular among the research community especially because independent mobility is a life-critical requirement. Obstacle detection and identification is a vital component of a navigation system. As mentioned in the earlier sections, there are various approaches have been introduced. Even though hanging wires can cause severe damage to blind people [13] there are limited number of researches can be found in the literature. Systems that are capable of detecting these obstacles are cannot be run in an embedded systems due to limited resources. Suggested method can be extendable in all kinds of navigation systems. This attempt is to address this research gap and provide accurate thin-structured wire detection methodology using computer vision while maintaining the simplicity of the system.

1.5 Methodology

Since this study addresses a more practical based scenario with the underline theoretical knowledge, the constructive research method is best suited for this research. During the course of the research project following steps were followed to synthesize a novel thin-wire detection algorithm. First, a thorough literature review will be done to identify the current limitations of this research area. Identifying research scope, research aim, objectives will be considered as the next stage. In the next step derived research questions will be addressed based on the observations which are made from the literature review. Evaluation of the algorithm will be done using both generated datasets and benchmark datasets. Finally, the conclusion about the algorithm and evaluation results in terms of accuracy and simplicity will be derived.

1.6 Outline of the Dissertation

The first chapter of the dissertation will outline the background of the research area as well as research questions, aims, methodologies, and scope of the research. Chapter 2 of the dissertation will critically review the reach area and will identify the research gap clearly, further it will discuss the importance of the problem using literature as well as possible avenues for solution to the problem. Chapter 3 will focus on the research design. Further, it will describe the evolution of the design and design choices made during the research process. Chapter 4 will explain details about implementation and challenges occurred during the implementation. Chapter 5 will discuss the evaluation of the proposed model and the results. Finally, Chapter 6 will provide discussion and conclusion about the positive and negative outcomes of the results, as well as limitations and future avenues for the research.

1.7 Delimitations of Scope

A complete SLAM based thin-wire detection blind navigation system consists of many complicated factors, due to the time limitations scope is narrowed to an achievable scope. Features of thin-wire detection blind navigation system divided as in scope and out of scope are shown in the table 1.1

Table 1.1: Scope and out of scope features

Scope	**Out of scope**
Edge extraction	Usability Evaluation
Edge tracking	Detection of moving objects
Edge mapping	Fuse with navigation systems
Reducing computation cost	Building into a embedded system
Edge linking	Implementation of depth estimation method
Depth estimation	Detecting other types of obstacles
Defining thin-wires	-
Detecting thin-wires	-

The algorithm developed during this research can be used to only detect thin-wires like obstacles. In order to use in the blind navigation system or any other navigation system, there are some further improvements to be done. Since the proposed model only detects thin wires it is recommended to fuse this system with another obstacle detection system when using for blind navigation.

1.8 Conclusion

This chapter laid the foundations for the dissertation. It introduced the research problem, research questions, research aim and objectives. Then the research was justified, definitions were presented, the methodology was briefly described and justified, the dissertation was outlined, and the scope and limitations were given. Based on this foundation, the dissertation can proceed with a detailed description of the research and its contributions.

Chapter 2

Literature Review

2.1 Introduction

This review is conducted mainly to identify solutions for the research questions posed in Chapter 1. A brief introduction to the research problem in the context of Computer vision is included in this chapter. It will also provide an insight into the solutions by identifying the strengths and weaknesses of each method.

2.2 Blind Navigation Systems

Blind navigation is a very broad field and studied across a large number of application domains. Blind Navigation systems can be divided into two main sub-domains called indoor blind navigation and outdoor blind navigation. This research is focused on outdoor navigation. According to the literature, blind navigation systems can be divided into 5 categories [3]. Figure 2.1 depicts these five main categories.

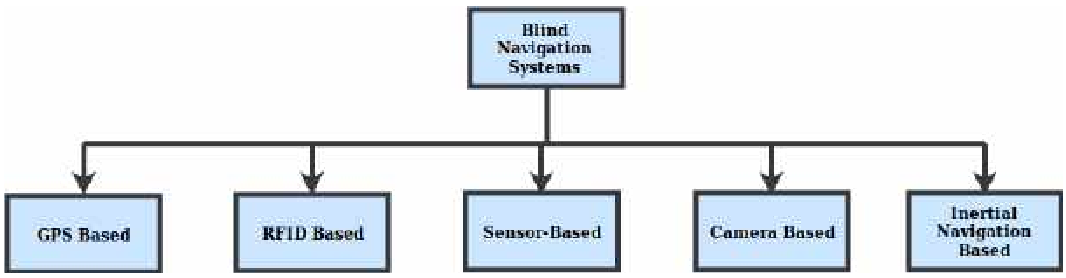


Figure 2.1: Classification of Blind Navigation Systems

In GPS based systems GPS receiver is used. According to the longitude and latitude values received from GPS-satellites users are navigated. These type of systems consists of GPS-receiver, Peripheral Interface Controller(PIC) microcontroller, and feedback module. These systems cannot be used in areas where GPS signals are weak. Radio Frequency Identification(RFID) based systems cosist of

RFID tags and RFID readers. RFID readers are required to be distributed over the navigational area, while RFID tags should attach to the blind person. RFID reader gets the signal from the RFID tag and calculates the position of the user. Sensor-based systems navigate blind people using sensor devices such as ultrasonic, infrared, light sensors or a combination of the aforementioned sensors. In Camera-based systems, camera devices are used to aid the blind navigation by capturing images along the path of the blind person. Inertial navigation systems consist of computer motion sensors and rotation sensors to continuously calculate via dead reckoning the position, orientation, and velocity (direction and speed of movement) of a moving blind person without the need for external references like GPS. Inertial navigation systems are used on vehicles such as ships, aircraft, submarines, guided missiles, and spacecraft.

Obstacle detection is a vital component for the above navigation systems. This task has been extensively studied in various communities including robotics, assistive, autonomous vehicles. Since the safety of blind people are paramount identifying and detecting obstacles in the blind navigation systems is an essential feature. These systems become more complex since they involve other measurements like calculating the distance to the obstacles, etc.

In the literature, researchers have tried various approaches to detect obstacles in blind navigation systems in outdoor environments. Mainly there are two approaches that can be seen such as sensor devices based approaches and computer vision-based approaches.

When considering sensor-based blind navigation systems white cane has become the basic platform for most of the researches. Alshajajeer et al [16] have implemented an enhanced white cane for visually impaired people which consists of an ultrasonic sensor module, raindrops sensor module and (Long-Distance Relationship(LDR) sensor module. Gbenga et al [17] proposed a Smart Walking Stick for Visually Impaired people. The proposed method consists of a simple walking stick equipped with sensors to give information about the environment. GPS technology integrated with pre-programmed locations allows the user to choose the optimal route to be taken. In the system, ultrasonic sensor, pit sensor, water sensor, GPS receiver, level converter, driver, vibrator, voice synthesizer, keypad, speaker or headphone, PIC16F877A microcontroller, and battery were used.

Different types of sensors work differently and they have their strengths and weaknesses. Since one sensor cannot provide all the necessary information in the environment some researchers have tried heterogeneous sensor fusion-based approaches. These type of systems can be more accurate when compared with homogeneous sensor-based systems. As a summary characteristics of often used sensors can be depicted in Table 2.1 as follows.

Table 2.1: Characteristics of sensors

Sonar	**Infrared**	**Lidar (TF mini)**
Range- 2cm to 4m	Range maximum 1m	Range- 30cm - 12m
Does not work accurately in outdoors	Not good for measure distance	Good for both indoor and outdoor use
Some objects absorb the sound	Uses as a proximity detector	Speed is high (100 readings per second)
Ultrasonic Response is subjected to change due to environmental effects such as temperature, pressure,humidity and air turbulence. Smooth surfaces reflect sound energy more efficiently than rough surfaces: foam and cloth tend to absorb more sound	Very sensitive to IR lights and sunlight, it has a weakness to darker colors such as black	Ineffective during heavy rain or low hanging clouds: LiDAR pulses may be affected by heavy rains or low hanging clouds because of the effects of refraction. However the data collected can still be used for analysis
Inexpensive	Inexpensive	Relatively expensive

Detecting thin structure wires using these active sensors can be challenging due to the following reasons. Ultrasonic range sensors have limited sensing range (typically, < 3 m) and difficulties of operating on highly reflective surfaces. Lidar sensors are expensive and have problems with portability when using embedded systems. Most of the sensor devices are sensitive to the light of the environment. In that case, there is a higher probability to miss thin structure obstacles. To tackle these difficulties this research proposed a computer vision-based approach to detect thin structure head and torso level obstacles for blind people.

2.3 Computer vision based obstacle detection

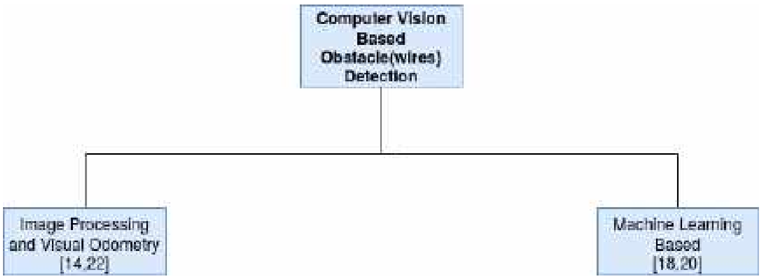


Figure 2.2: Classification of Computer vision-based obstacle detection systems

Figure 2.2 shows the two main approaches of computer vision-based obstacle detection that can be found in the literature. Apart from the two methods in the Figure 2.2, Computer vision-based obstacle detection methods can be grouped into two other groups called monocular camera-based or stereo camera-based. Monocular methods use single-camera and usually detect obstacles by image processing techniques and estimate the depth by using machine learning techniques (Supervised learning) or monocular depth cues. Stereo camera-based methods use two cameras and can have the ability to estimate the distance to the obstacles more accurately than monocular camera-based methods.

Michels et al [18] proposed obstacle avoidance navigation system for vehicles. Video frames were fed into the system and each frame was divided into vertical stripes. The depth of each image stripe was calculated using a regression model using texture features. Hedenberg and Astrand [19] have presented a system based on a trinocular camera system. Using the resulting images dense disparity map and sparse disparity map were created. Canny edge detector and Sobel filter were applied on these disparity maps to detect edges. According to the evaluation results hanging cables were detected properly but both detectors were not able to detect horizontal ladder obstacles due to it was nearly the same grey value as the background. Mancini et al [20] have presented a paper under the title of Fast Robust Monocular Depth Estimation for Obstacle Detection with Fully Convolutional Networks. In this paper, researchers have explored the architecture and performance of a depth estimation algorithm based on encoder-decoder convolutional neural network architecture. The encoder section was composed of a stack of convolutional layers which apply learned filters on their input and extracted relevant synthetic features. The decoder section is composed of three deconvolutional layers. Each deconvolutional layer learns to upsample encoder feature maps by, respectively, a factor of 2 for the first two layers and a factor of 4 for the final layer. One limitation identified was hard to generalize the model since the environment has huge diversity. Mori and Scherer [21] have implemented a system that consists of detecting surf features and checking the scale ratio using template matching techniques. Based on this scale difference depth has been calculated. But when using this approach obstacles should have good texture otherwise surf features are not detected them as obstacles. Because of the above limitations, this research is trying to address this problem using image processing and visual odometry approach.

While the above methods show accurate results for large and medium-sized regular-shaped obstacles little attention has been given to the thin obstacles detection such as tree branches, cables in the literature. Zhou et al [15] have presented a system to detect thin structure obstacle for UAVs such as drones. Video sequence was used in the prototype. Obstacles were represented with edges in the video frames and reconstructed them in 3D using efficient edge-based visual odometry to calculate the distance to the obstacles. Visual odometry is the process of determining the position and orientation of a robot by analyzing the associated camera images. It has been used in a wide variety of navigation applications. In visual odometry based systems information is obtained by extracting image feature points and track them in an image sequence. This system consists of both monocular and stereo based approaches. In this proposed system Difference of Guassian(DoG) based detector was used in combination with a consequent Canny-Style hypothesis to enhance the weak edges. DoG detector has been chosen because of its good receptivity. After extracting image edges visual odometry algorithm was used to estimate the depth. This algorithm was proposed by Tarrio and Pedre [22]. Since this algorithm only stores local inverse depth maps of the current map it was considered as light-weight. When a new frame comes camera motion is tracked by fitting the edge map from the current frame to the edge map of the new frame.

2.4 Visual Odometry and Simultaneous Localization and Mapping

Visual Odometry is defined as the process of estimating the agent's(robot,UAV)motion (translation and rotation with respect to a reference frame) by observing a sequence of images of its environment [23]. Whereas Simultaneous Localization and Mapping(SLAM) is a process in which a robot is required to localize itself in an unknown environment and build a map of this environment at the same time without any prior information with the aid of external sensors.

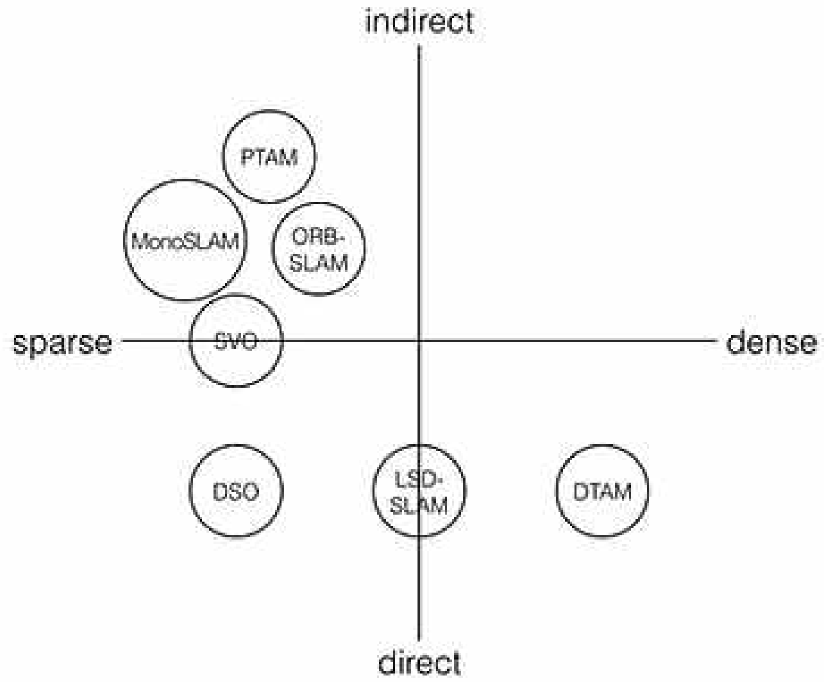


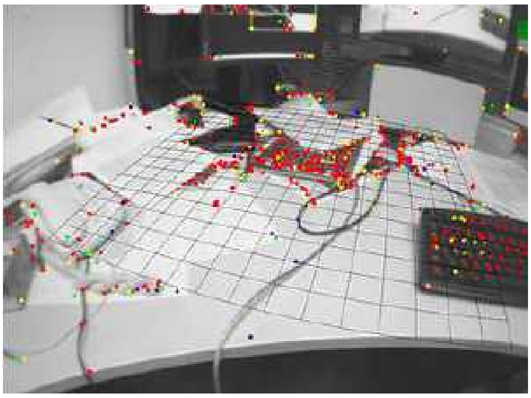
Figure 2.3: Classification of SLAM systems

There are four types of SLAM systems can be found in the literature. Parallel Tracking and Mapping for Small AR Workspaces(PTAM) [24], Monocular SLAM(MonoSLAM) [25], Oriented FAST and Rotated BRIEF SLAM(ORB-SLAM) [26], Fast Semi-Direct Monocular Visual Odometry (SVO) [27], Direct Sparse Odometry (DSO) [28], Dense Tracking and Mapping in real-time (DTAM) [29], Large-Scale Direct Monocular SLAM (LSD-SLAM) [30] algorithms are shown in figure 2.3 based on the type.

SLAM systems can be classified as either sparse or dense based on how the area of the image is used. In the sparse SLAM systems, only a small selected number of pixels are used in an image frame. Since sparse SLAM systems generate point cloud maps, very fewer details about the environments are given but it can be runnable with limited resources. These systems can only be used for tracking the camera pose. In the dense SLAM systems, all pixels are used and gives more information about the environments. Dense SLAM systems required more powerful hardware when compared with sparse SLAM systems.

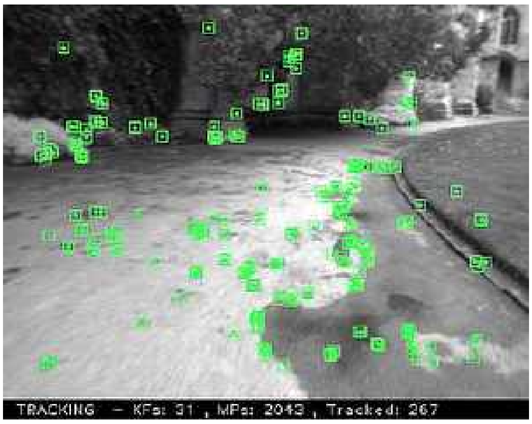
Another way to classify SLAM systems as either direct or indirect based on how the information is extracted. Indirect SLAM systems get information by extracting features from the image and make use of these features to locate the camera and build the map. These features can be image edges, image corners, Scale-Invariant

Feature Transform(SIFT) features, speeded Up Robust Features(SURF) features, ORB, FAST, etc. Since this method doesn't deal with image intensities, it is reliable with different lighting conditions. These Systems are quite slow due to the feature extraction procedure. Direct methods make use of pixel intensities directly, rather than extracting intermediate features. These direct methods have the capability to recover both environment depth and camera pose. Visualizations of each type of SLAM systems are shown in Figure 2.4

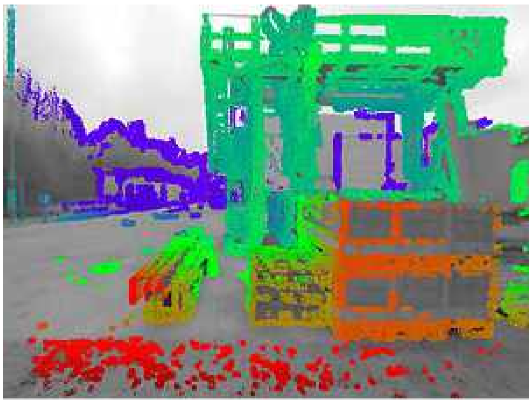


(a) The sparse map created by PTAM, where the coloured points are map points

(b) The dense map generated by the DTAM system. All points on the surface are part of the map



(c) ORB feature points in the ORB-SLAM system



(d) The semi-dense map in the LSD-SLAM, where the coloured points are map points

Figure 2.4: Visualizations of four types of SLAM systems

2.5 Conclusion of the Literature Review

Blind navigation systems are classified into five main categories called GPS-based, sensors-based, camera-based, RFID-based and Inertial measurement-based systems. Furthermore camera-based systems are divided into two main categories named image processing based and machine learning-based systems. Due to several drawbacks that are mentioned in the literature image processing based approach is chosen. Four types of SLAM systems can be found in the literature called sparse, dense, direct and indirect SLAM.

Chapter 3

Design

3.1 Introduction

This chapter explains how research questions are addressed. Further, it describes the design choices and evolution of the model throughout the research and describes the design evolution and analysis of each design approach. Apart from that this chapter provides a higher-level architecture of the proposed approach and definitions and assumptions made during the research.

3.2 Design Evolution

There are three main approaches that can be found in the recent literature to detect thin-wires. The first method was to detect wires using active sensors. Due to limitations that are mentioned in Section 2.1, this research is trying to address the problem using computer vision. Computer vision-based approaches can be divided into two main categories called image-processing approaches, visual odometry approaches and machine learning approaches. In order to avoid limitations that were identified in Section 2.2, this research address the research problem using image processing and visual odometry approach. As mentioned the Chapter 2 image processing approaches are based on either the monocular camera or stereo camera versions. Since one of the goals of this research is to maintain the simplicity of the system monocular camera-based approach was chosen. SLAM approach was chosen since the problem consists of tracking and detecting the obstacle's(wires) and depth estimation. Based on the areas used in the image SLAM systems are classified into two types called sparse and dense. Table 3.1 depicts the key features of Sparse and Dense SLAM systems.

Table 3.1: features of Sparse and Dense SLAM systems

Sparse SLAM	• Uses only a small selected subset of the pixels in the image frame • The map contains points cloud • Mainly use to track the camera motion • Provide fewer details about the environment
Dense SLAM	• Use all the pixel in the image frame • Provide more details about the environment • More powerful hardware is needed

Based on the information utilize method of the received image SLAM systems can be classified into categories called direct SLAM and indirect SLAM. Key features of Direct and Indirect SLAM systems are shown in Table 3.2.

Table 3.2: Features of Direct and Indirect SLAM systems

Direct SLAM	• Make use of pixel intensities directly • Gives environment depth, camera pose more accurately
Indirect SLAM	• Use features to locate the camera and build the map • Features can be simple geometric features or edges or corners (SIFT, SURF, ORB, FAST)

According to Table 3.1 and Table 3.2 each type of SLAM system has both advantages and disadvantages. The main idea of the proposed approach is to develop an algorithm between classical feature-based visual odometry systems and modern direct dense/semi-dense methods to avoid those disadvantages and limitations.The main idea of the proposed methodology is to find a fast and simple thin-wire detection algorithm to work on an embedded blind navigation system by finding the middle point of above SLAM systems.

3.2.1 Defining Thin-wires

This research mainly focuses on detecting thin-structured wires that can be strangled on blind people's head and torso level. Based on the observations that were done in the outdoor environment several types of lower hanging obstacles were identified and among them, thin-structured wires were chosen. In this research domain, a thin-structured wire is defined as a wire with a maximum of 5mm diameter.

3.3 Research Design

The proposed algorithm was designed based on the SLAM algorithm which was developed by Tarrio and Pedre [22]. The proposed algorithm consists of three main steps which are common to most of the SLAM systems named information extraction, tracking stage and mapping stage Figure 3.1 depicts the higher-level view of the proposed algorithm.

Figure 3.1: High level architecture of the proposed algorithm

3.3.1 Edge Extraction

Edge detection algorithms can be basically classified on the behavioral study of edges [31, 32] with respect to the operators . Table 3.3 shows the summary of main edge detection algorithms and their characteristics.

Table 3.3: Characteristics of key edge detection algorithms

Method	**Features**
Gradient methods(Roberts, Sobel, Prewits)	• Uses first derivative operation • Sensitive to noise • Simplicity • Capability of detecting edge orientations
Zero crossings (DoG, LoG)	• Uses second derivative operation • Sensitive to noise • Simplicity • Can find positions of the edges more accurately
Gaussian methods (Canny)	• Uses probability for finding error rate • Consists of complex computations • time consuming • Uses Gaussian filter operation

Since this research focuses on finding a fast, simple and accurate wire detection system choice of the edge detection algorithm has an impact on the final performance of the system. When choosing an edge detection algorithm following characteristics were considered.

- Repetition of the same edge in consecutive frames
- Precise localization of edge pixels
- Low running time

Based on the above requirements and based on the conclusion obtained from the literature, Zero crossing of DoG based detector was chosen.

Gray scale images that are smoothed with two different sigmas are used for the DoG calculation. The gradient is calculated using the lowest sigma image. Norm of the gradient is used to threshold edge pixels and eliminate false edges. A subpixel level of positions is obtained by fitting a plane to DoG. Then, edge thinning is obtained by discarding those pixels whose subpixel position lies more than half-pixel away from the center of the pixel. Finally, edge linking step is applied by doing a simple search on 20 pixels that surround each edge point. The output of this step are the edge pixels, subpixel position of the image, estimated inverse depth, estimated uncertainty, references to next and previous edge points. Figure 3.2 depicts the main steps of the edge extraction stage.

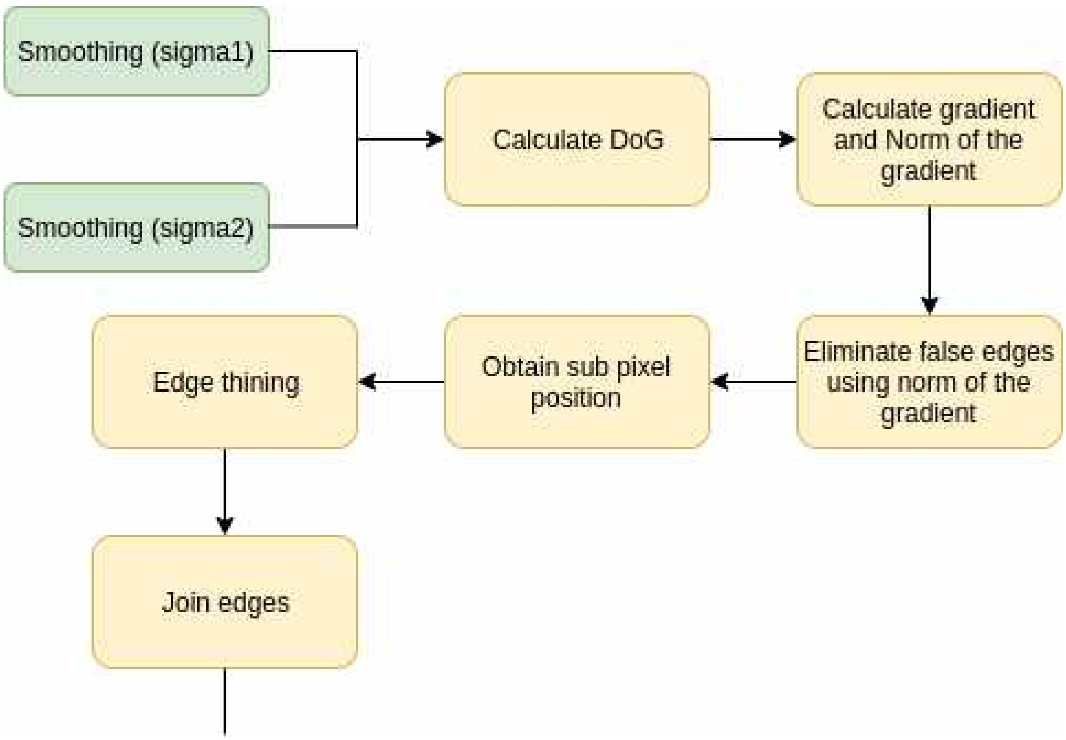


Figure 3.2: Key steps of edge extraction stage

3.3.2 Edge Tracking

In the tracking stage main objective is to find a SE(3) transformation that maps the previous edge map into the new edge map. The rigid body transformation consists of three main steps called reflection, translation, and rotation. Excluding reflection, any proper rigid transformation can be decomposed as rotation followed by the translation. The proper rigid transformation in a 3-dimentaional Euclidean space denoted as SE(3). This tracking stage is completely built upon the algorithm proposed by Tarrio and Pedre [22]. According to that algorithm, only the previous edge map and current edge map are stored instead of using a global edge map. When a new frame receives, camera motion is tracked by e fitting the edge map from the previous frame to the edge map of the new frame using a warping function $W(q,\rho,x) : R^2 \times R \times R^6 \to R^2$ and geometric error is minimized using the minimization function $E_0(w,v) = \rho((W(q,\rho,x) - q_i) \times g_i)$ where previous frame's image coordinates q, inverse depth ρ, SE(3) transformation x, corresponding edge point in the new frame q_i, gradient direction of q_i is g_i. Using this method, complexity of the system can be reduced. Figure 3.3 illustrates how edge tracking works using warping function,points $(\overline{q_0}, \rho_0)$ from the previous edge map are projected into point $(\overline{q_t}, \rho_t)$ warping function

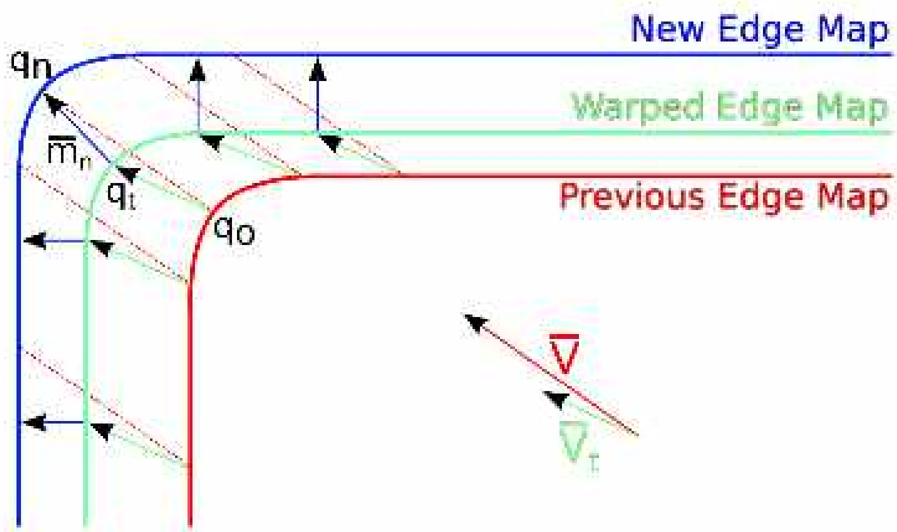


Figure 3.3: Edge warping and searching in the tracking stage. Rotation has been ignored for the sake of simplicity

3.3.3 Edge Mapping

In the mapping stage, it was done by epipolar search from the new edge map to the previous edge map instead of mapping from the previous edge map to a new edge map. Because of that possibility of finding a match for every edge point in the new edge map is increased. Each edge point in the new edge is back-rotated using the SE(3) transformations which are obtained in the previous tracking stage to get the transformed image point. The inverse depth of the previous frame is propagated to the new frame using a camera motion. The inverse depth map of the current frame is propagated to the new frame using camera motion and updated with new observations obtained during the epipolar search. During this epipolar search only up to given maximum number of pixels(d) are searched. When a possible match is found, gradient Vectors of previous and the new edge point are compared to test for compatibility as done in the tracking stage. Then the depth estimation process is applied followed by a regularization step as described by the Tarrio and Pedre [22].

3.4 Summary

A wire with a minimum of 5mm diameter is considered as a thin-structured wire. A hybrid SLAM based approach is proposed to detect thin-structured wires in this research. The design of the proposed solution is divided into three main stages called edge extraction, edge tracking and edge mapping. Moreover the proposed approach is based on the work which was done by Tarrio and Pedre [22].

Chapter 4

Implementation

4.1 Introduction

This chapter explains the implementation of the proposed model. Further this will describe the evolution of the Proof of Concept Prototype(PoF) implementation and it describes the technologies and software tools used in the implementation and analysis of those tools. Section 4.3 explains how datasets were created and preprocessing which are done before input to the algorithm. In the next section, proposed algorithms are explained in a detailed manner. Section 4.4 discusses the final research design implementation details explaining core modules such as edge labeling and edge linking steps. Section 4.5 discusses the challenges faced during the prototyping and implementation process. Finally, Section 4.6 provides a summary of the implementation process.

4.2 Tools and Technologies

The proposed solution was built upon the framework called Real-time edge-based visual odometry(REBVO) that was developed by Tarrio and Pedre [22]. REBVO has been built using the C++ language. In this REBVO system, there are several libraries have used including OpenGL for visualizations, TooN mathematical library for matrix creation and operations, LibAV for video codecs and LibGD for image management. During the different stages of the research Python, Java, OpenCV, pandas, numpy like languages and libraries were used. Two android applications were built for getting IMU data from the camera and calculating the focal length and intrinsic parameters of the mobile phone camera using Android studio. The PC used in this research has the following specifications, as it has to have the processing power to accommodate the requirements of all resources.

- Processor: Intel(R) core i7
- RAM: 8GB
- Operating System: Ubuntu 16.04LTS
- Graphics: 2GB onboard VGA

The current prototype can only be run on Linux platforms.

4.3 Dataset Preparation

Since there are no suitable benchmark datasets that can be used for thin-wire like obstacle detection appropriate dataset was created. When creating a dataset different places like roads, grounds, parks, etc were chosen with different lighting conditions and different angles. Some environments were created manually by putting thin wires and some datasets created by choosing the places with thin wires. Figure 4.1 depicts the image frame before applying the processing steps.

Figure 4.1: Image frame before applying preprocessing steps

The video stream is taken from the android mobile phone which has a 12Mp camera and 500mm focal length. After creating video sequences following preprocessing steps were applied before input to the proposed system.

- Converting every frame to Grayscale
- Converting every frame to the size of (752,480)
- Naming each frame according to the timestamp

Figure 4.2 illustrates the preprocessed image frames

Figure 4.2: Image frame after applying preprocessing steps

4.4 Implementation Details

Since the proposed solution required camera focal length and required to capture accelerometer and gyroscope data while recording the video there were two open source android apps were modified and build. Figure 4.3, figure 4.4 illustrates the main interfaces of those two apps.

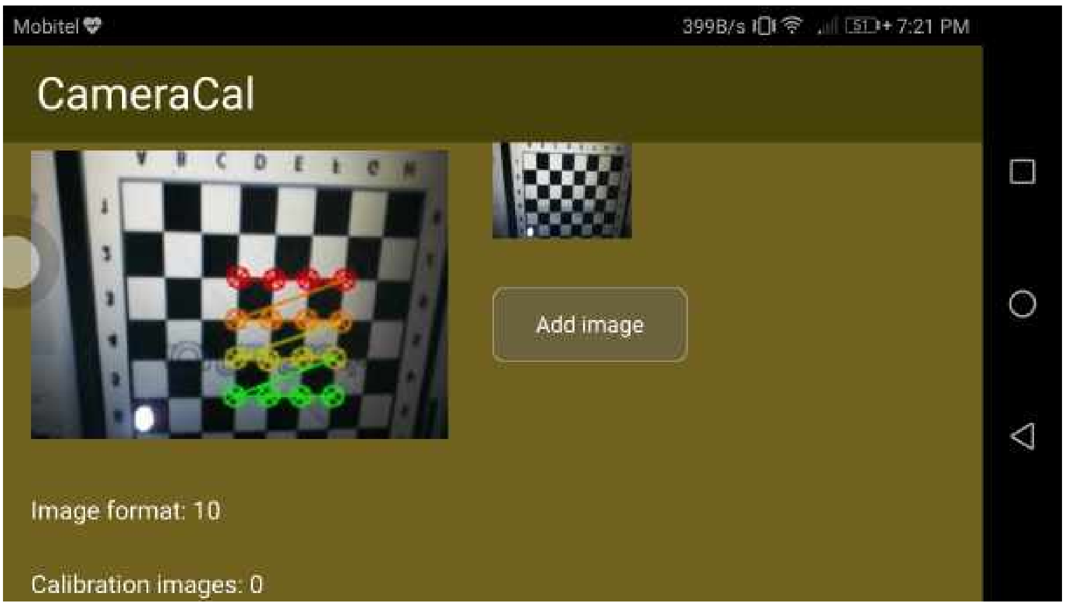


Figure 4.3: CameraCal - Application used for calculate camera focal length

Figure 4.4: PilotGuru - Application used for record IMU data

As explained in the previous sections work the proposed solution based on the REBVO system. There are three main modifications have been proposed.

- Edge Linking
- Small connected component filtering
- Obstacle labelling

4.4.1 Edge Linking

As explained in the Section 3.3.3 in the edge map is created based on the extracted edges considering repetition of the edges and their localization. In order to enhance the detection of edges, edge linking step is proposed. As shown in Figure 4.5 if x is an edge pixel search for the next edge is performed every eight directions up to max d pixels. d is determined based on the environment condition. When the lighting condition is good, better to use a smaller value for d (2-3) [22] and when the lighting condition is bad bit larger value for d is recommended. Code level details about the algorithm is included in appendix A.

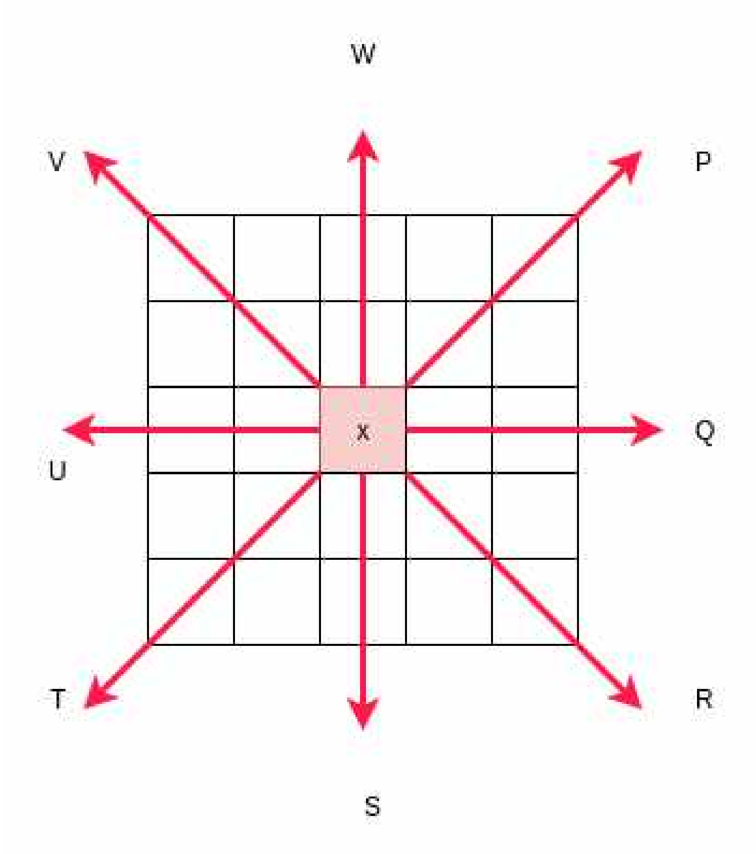


Figure 4.5: Edge linking

4.4.2 Small Connected Components Filtering

When detecting wires from the edge map removing unnecessary noise is very important. As shown in figure 4.6 there is a large number of connected components can be found in an image frame. In order to detect wires accurately among these noise edges, a noise filtering algorithm is applied.

Figure 4.6: Edge extraction with noisy edges

Figure 4.7 depicts the code level implementation for the above-mentioned filtering algorithm. Algorithm search for a connected component of a given length. Since every edge point is consists of two pointers to the previous and the next edge point connected component length is determined by traversing to the next and previous edge points and one hop from current edge point to next or previous edge point is considered as size of one length. If the calculated length is less than the given maximum length then all the edge points are neglected.

```
for (int fil = 0; fil < kn; fil++){
    int maximumLength = 10;
    if (kl[fil].checked == 0){
        int length = 0;
        int nextKl = fil;
        int prevKl = fil;
        while (length <= maximumLength && (kl[prevKl].p_id != -1 || kl[nextKl].n_id != -1)) {
            if (kl[nextKl].n_id != -1) {
                kl[nextKl].checked = 1;
                length++;
                nextKl = kl[nextKl].n_id;
            }
            if (kl[prevKl].p_id != -1) {
                kl[prevKl].checked = 1;
                length++;
                prevKl = kl[prevKl].p_id;
            }
        }
        if (length <= maximumLength) {
            int nextKl_r = fil;
            int prevKl_r = fil;
            int p, q, r, s, imageIndex1, imageIndex2;
            while ((kl[prevKl_r].p_id != -1 || kl[nextKl_r].n_id != -1)) {
                if (kl[nextKl_r].n_id != -1){
                    kl[nextKl_r].checked = -1;
                    p = util::round2int_positive(kl[nextKl_r].c_p.x);
                    q = util::round2int_positive(kl[nextKl_r].c_p.y);
                    imageIndex1 = img_mask_kl.GetIndex(p, q);
                    img_mask_kl[imageIndex1] = -1;
                    nextKl_r = kl[nextKl_r].n_id;
                }
                if (kl[prevKl_r].p_id != -1){
                    kl[prevKl_r].checked = -1;
                    r = util::round2int_positive(kl[prevKl_r].c_p.x);
                    s = util::round2int_positive(kl[prevKl_r].c_p.y);
                    imageIndex2 = img_mask_kl.GetIndex(r, s);
                    img_mask_kl[imageIndex2] = -1;
                    prevKl_r = kl[prevKl_r].p_id;
                }
            }
        }
    }
}
```

Figure 4.7: Small connected component filtering algorithm

4.4.3 Wires Labelling

In order to do robust obstacle labeling, here only considered the edge pixels with stable inverse depth estimations that have been observed and matched across multiple frames.

```cpp
struct KeyLine{

    int  p_inx;          //KeyLine linear index of image position

    Point2DF m_m;        //KeyLine's gradient vector
    Point2DF u_m;        //Normalized m_m
    float n_m;           //Norm of m_m

    float score;         //Final score given to the keyline

    Point2DF c_p;        //KeyLine's image position

    double rho;          //Estimated Inverse Depth
    double s_rho;        //Estimated Inverse Depth Uncertainty

    double rho_nr;           //Estimated Inverse Depth Non-Regularized
    double s_rho_nr;         //Estimated Inverse Depth Non-Regularized


    double rho0;         //Predicted Inverse Depth in EKF (use only if reescaling)
    double s_rho0;       //Predicted Inverse Depth Uncertainty

    Point2DF p_m;        //KL position in homogeneous coordinates (plane on focal length zf)
    Point2DF p_m_0;      //matched KL position in homogeneous coordinates

    int m_id;            //Id of the matching keyline
    int m_id_f;          //Id of the matching keyline by fordward matching
    int m_id_kf;         //Id of the matching keyline in the last keyframe

    int m_num;           //number of consecutive matches

    Point2DF m_m0;       //Grandient of matched KeyLine
    double n_m0;         //Norm of m_m0

    int p_id;            //Id of previous consecutive KeyLine
    int n_id;            //Id of next consecutive KeyLine
    int net_id;          //Network ID of the KeyLine
    int checked;

    int stereo_m_id;
    double stereo_rho;          //Estimated Inverse Depth from stereo
    double stereo_s_rho;        //Estimated Inverse Depth Uncertainty from stereo

};
```

Figure 4.8: Struct of the Edge point

As shown in Figure 4.8 beside image coordinates, each edge point consists of inverse depth, variance, the number of frames it has been successfully matched. Based on these features edge points with stable inverse depth are selected as edge points.

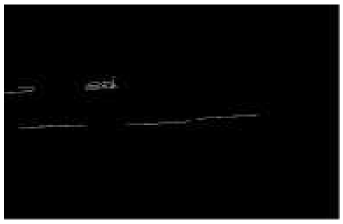
```
[I, map] = imread('/Users/lakshan/Desktop/edge.png');

% Original Image
subplot(2,3,1);
imshow(I);

% Extract blue color
I = I-x;
I = I(:,:,3);
subplot(2,3,2);
imshow(I)

% Remove small regions
I = bwareaopen(I, 100);
I = uint8(I); I(I==1) = 255;
subplot(2,3,3);
imshow(I);

% Dilate
I = imdilate(I, strel('rectangle', [20, 20]));
I = uint8(I); I(I==1) = 255;
subplot(2,3,4);
imshow(I);

% Remove small regions
I = bwareaopen(I, 5000);
I = uint8(I); I(I==1) = 255;
subplot(2,3,5);
imshow(I);

% Erode
se = strel('cube',10);
I = imerode(I, se);
subplot(2,3,6);
imshow(I);
```

Figure 4.9: Wires labelling algorithm

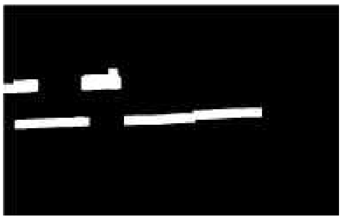
Due to limited time constraints, this algorithm was implemented using Matlab and not combined with the original REBVO algorithm. According to the algorithm, the converged edge map is inserted as the input. In the next step blue color is extracted from the image. In order to remove unwanted noise edges small regions filtering function is applied. In the next step, a dilation step is applied to wide the remaining edge distribution. Small regions filtering step applied again and finally, the erosion step is applied to sharp the remaining edges. Basic steps of the proposed algorithm are shown in figure 4.10.

(a) Original image

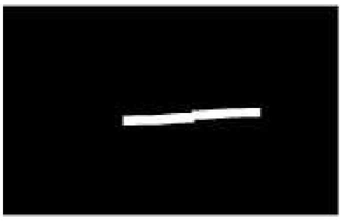
(b) After extracting blue color



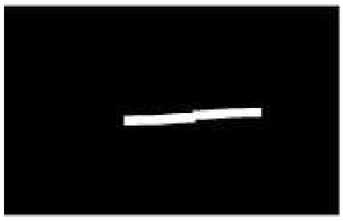
(c) After removing small regions



(d) After applying dilation



(e) After applying small region filtering for the second time



(f) After applying erosion

Figure 4.10: Stages of wires identification algorithm from (a) to (f)

4.5 Implementation Challenges

At the initial stage of the project, several SLAM implementations were deployed including PTAM, LSD SLAM, and Semi-Global Block Matching(SGBM). The major issue encountered was the lack of proper documentation. Due to the above problem, lots of time was spent when deploying and debugging the systems. When deploying the REBVO system there were lots of version conflict issues were arisen. Lack of suitable benchmark datasets was the next major problem faced. Because of that suitable datasets were created. When creating a dataset recording Inertial Measurement Unit(IMU) data was another challenge and an opensource android application was modified and built for resolve that problem.

4.6 Summary

The original REBVO algorithm is extended by introducing an edge linking step and small connected components filtering algorithm in order to detect thin-structured wire edges. Separate algortihm is developed for the identification of the wires using Matlab.

Chapter 5

Results and Evaluation

5.1 Introduction

This chapter explains how the proposed algorithm is evaluated and the results of the evaluation. The algorithm was tested using datasets generated by the author. This section consists of the results of the proposed algorithm and comparison to baseline methods. Furthermore, the proposed algorithm was evaluated based on the lighting condition, camera movements, and motion blur. Performance evaluation was done by profiling the algorithm. Figure 5.1 depicts the evaluation plan for the proposed algorithm.

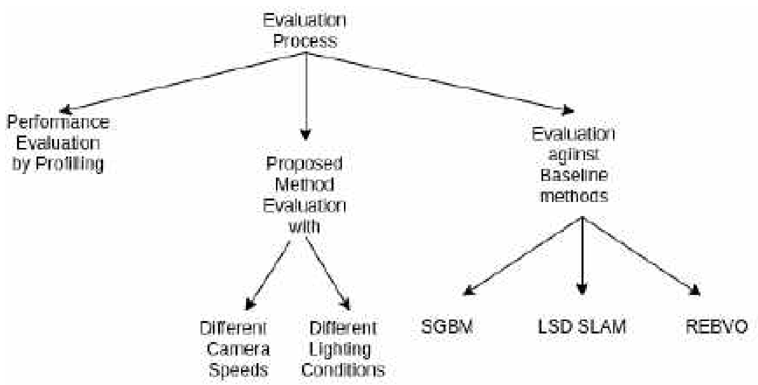
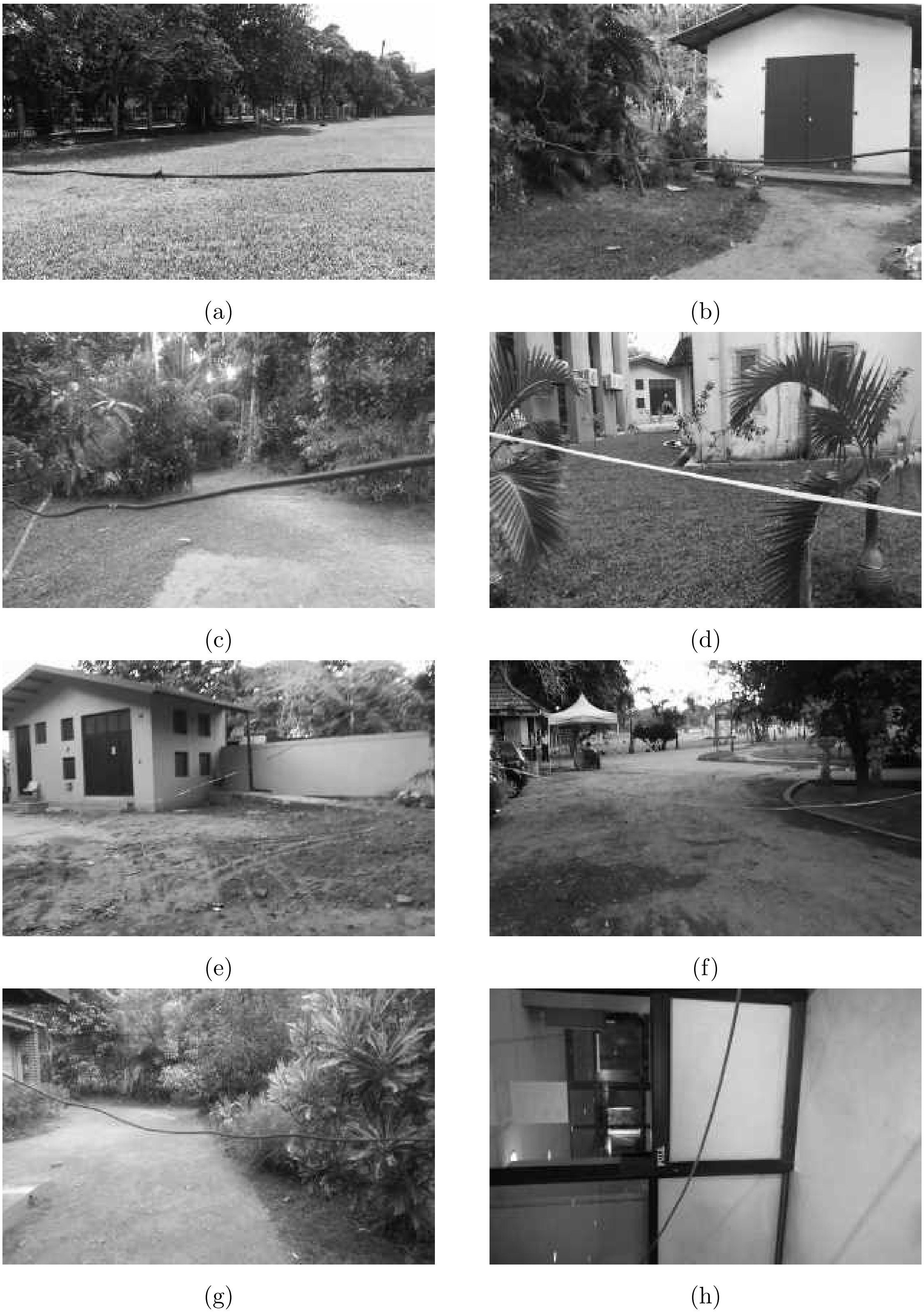


Figure 5.1: Evaluation Plan

5.2 Dataset

When considering the nature of this problem there are no suitable datasets that can be used to evaluate the proposed algorithm. Hence fifty different datasets were created using Huawei GR5 2017 and HTC M8 mobile phone cameras. When creating datasets camera motion, lighting conditions and motion blur were considered. Then preprocessing steps were applied as mentioned in section 4.3. Following figure 5.2 illustrates different datasets are created for the evaluation of the proposed algorithm.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5.2: Datasets which are used for evaluation

5.3 Qualitative Results and Comparison to Baseline Methods

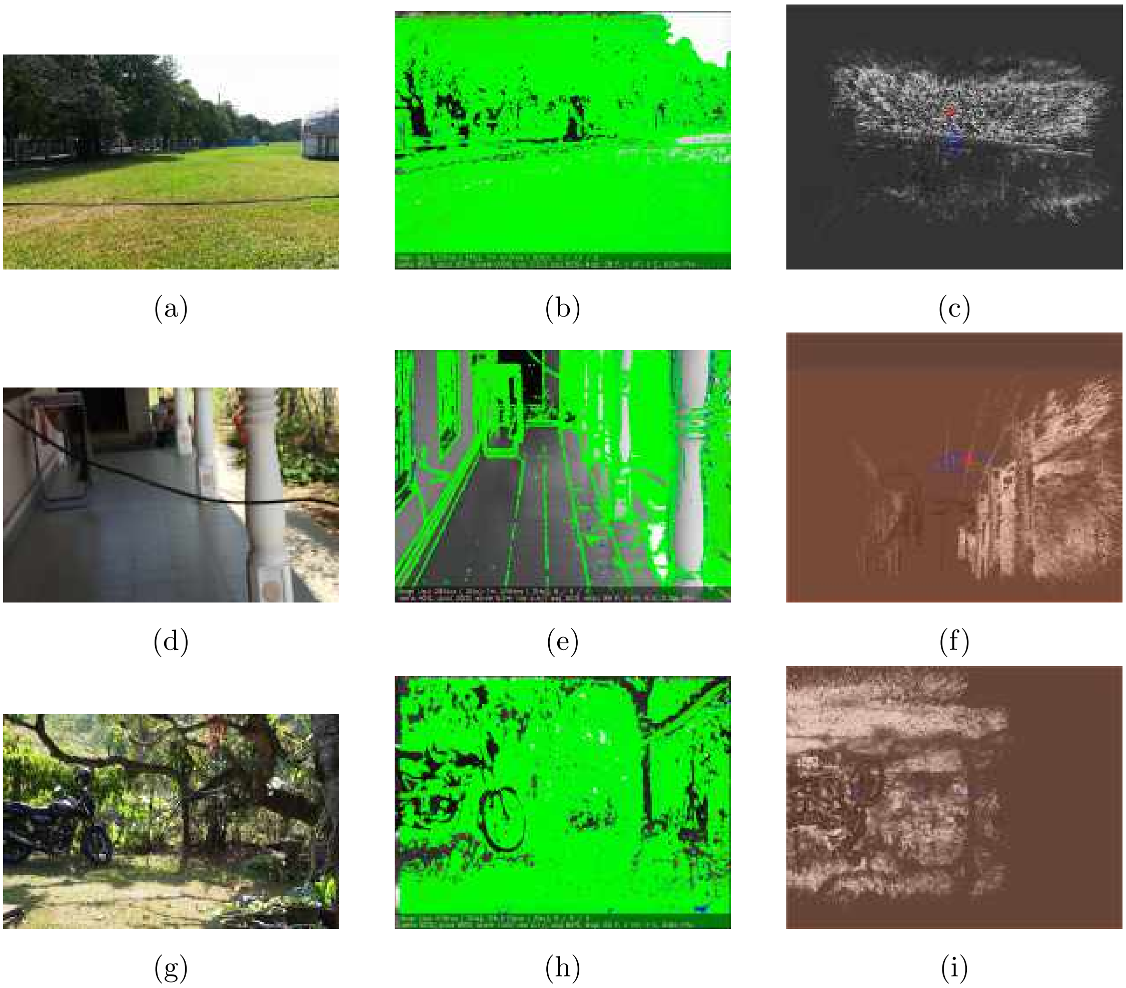
There are two baseline methods selected from the literature called Stereo SGBM and LSD SLAM in order to evaluate and compare the results of the proposed algorithm. SGBM can be considered as a disparity estimation algorithm which was originally founded by Hirschmuller [33]. SGBM algorithm takes a pair of left and right images as input. Both images should be rectified in order to make all epi-polar lines parallel to the horizontal axis. In the SGBM algorithm disparity of a pixel is calculated by considering the block of pixels instead of using the whole image. Furthermore in the original SGBM algorithm, information from the five neighboring pixels were used to calculate the disparity of a pixel. Since the Stereo SGBM is an OpenCV implementation of original SGBM it uses all the eight neighboring pixels for the disparity calculation. Thus, this algorithm uses block-based cost matching technique.

LSD SLAM is also chosen for the evaluation which has been developed by Engel et al [30] in 2014. As mentioned in section 2.4 LSD SLAM is a semi-dense SLAM system that only deal with pixel intensities instead of key points or features. Global Semi dense depth map is created to estimate the geometry. The camera is tracked by using direct image alignment.

As stated in the previous sections the proposed algorithm was based on the REBVO algorithm which was presented by Tarrio and Pedre [22]. When evaluating the algorithm original REBVO algorithm was also compared with the modified REBVO algorithm. There were fifty created datasets used for the evaluation and following figures depict the obtained results only for five different datasets.

5.3.1 LSD-SLAM Evaluation

Figure 5.3 depicts the results that are obtained for the created datasets. According to the results, there is no distinction can be found between wire edges and other edges. When the complexity of the environment is high, the quality of the edge map also reduced. This method deals with image intensities and it builds a global pose graph map, therefore it requires powerful hardware resources.



(a) (b) (c)
(d) (e) (f)
(g) (h) (i)

Figure 5.3: Original images are shown in (a),(d),(g) and (b),(e),(h) are the depth maps created and (c),(f),(i) are the point clouds created for corresponding original images

5.3.2 SGBM Evaluation

Figure 5.4 interprets the selected results that are obtained from the created datasets for the SGBM algorithm. This method requires a pair of left and right images for the creation of the disparity map. Disparity maps that were obtained for the generated datasets were not clear enough for the detection of wires. Furthermore, this algorithm is slower than the other baseline algorithm so that this algorithm can not use with real-time video sequences.

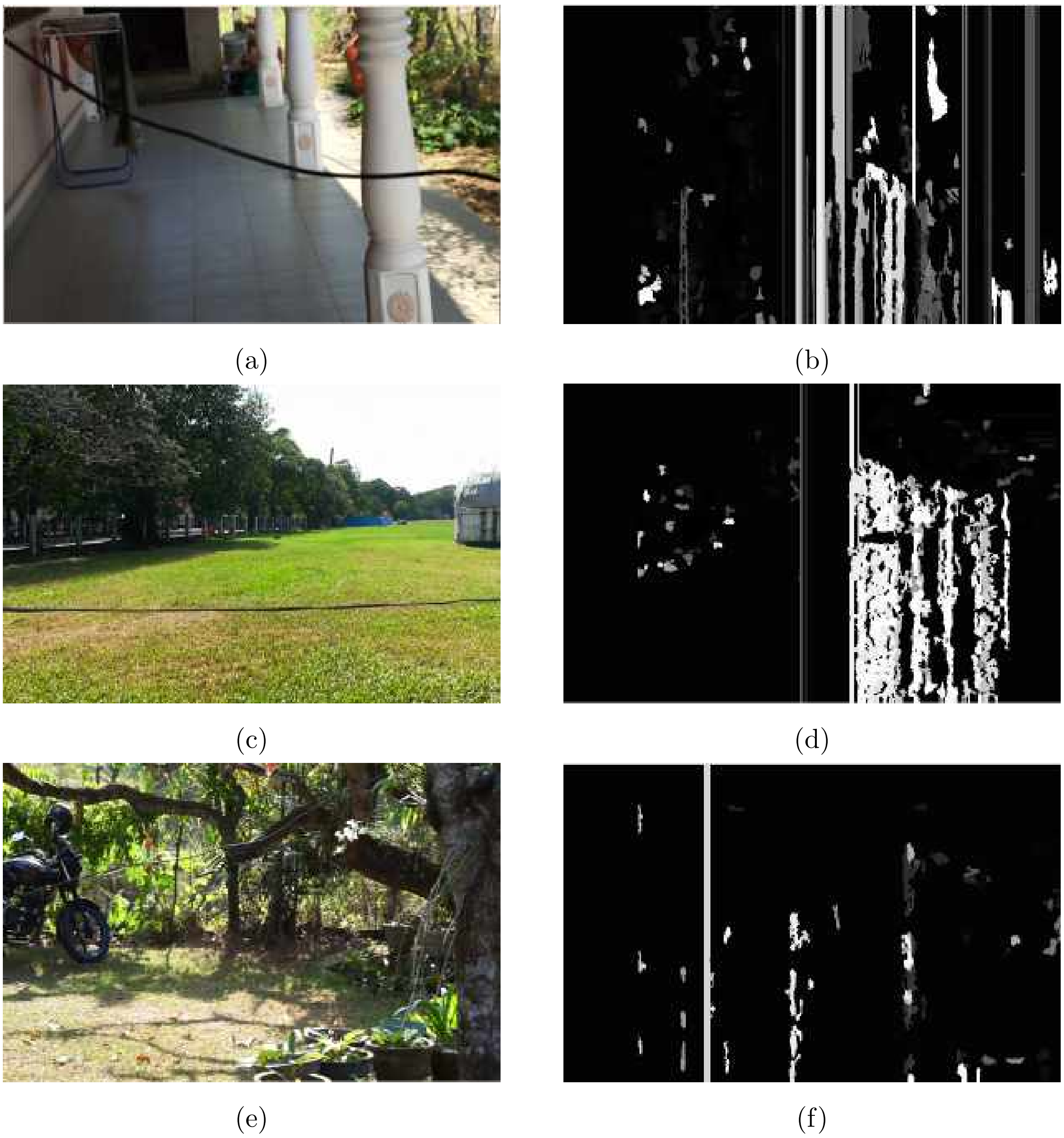


Figure 5.4: Original images of datasets are shown in (a),(c),(e) and (b),(d),(f) shows corresponding disparity maps for each image

5.3.3 Proposed Approach Evaluation

Figure 5.5, 5.6 and 5.7 visualizes the results that are obtained from the original REBVO algorithm and proposed algorithm.

When comparing the results there is a positive improvement that can be seen in the detection of the edges in the proposed modified algorithm.

(a) Original image

(b) REBVO edge detection output

(c) REBVO edge map

(d) Original image

(e) Proposed method edge detection output

(f) Proposed method edge map

Figure 5.5: Evaluation results of proposed approach for created dataset 01 over the original REBVO algorithm

(a) Original image

(b) REBVO edge detection output

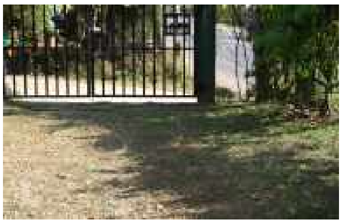
(c) REBVO edge map

(d) Original image

(e) Proposed approach edge detection output

(f) Proposed approach edge map

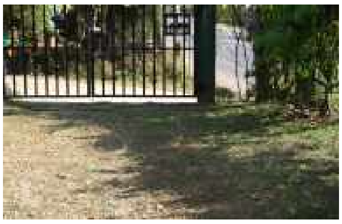
Figure 5.6: Evaluation results of proposed approach for created dataset 02 over the original REBVO algorithm



(a) Original image

(b) REBVO edge detection output

(c) REBVO edge map



(d) Original image

(e) Proposed approach edge detection output

(f) Proposed approach edge map

Figure 5.7: Evaluation results of proposed approach for created dataset 03 over the original REBVO algorithm

5.4 Evaluation with Different Lighting Conditions

The proposed approach was evaluated in different lighting conditions. Morning, Afternoon, and Evening time periods were chosen and light intensity was measured by using a light meter mobile app. As seen in figure 5.8(a) and figure 5.8(g), when the light intensity is high the results are more accurate and depth maps are clearer. Furthermore, when the light intensity is too high and too low(figure 5.8(d), figure5.8(j) depth maps were not converged

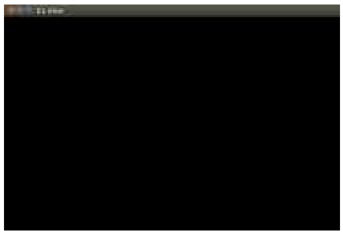
(a) Original image(Light intensity 800LUX)

(b) REBVO edge detection output

(c) REBVO edge map

(d) Original image(Light intensity 2000LUX

(e) Proposed approach edge detection output



(f) Proposed approach edge map

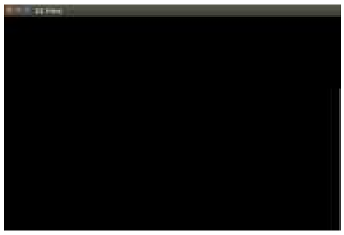
(g) Original image(Light intensity 1500LUX

(h) Proposed approach edge detection output

(i) Proposed approach edge map

(j) Original image(Light intensity 170LUX)

(k) Proposed approach edge detection output



(l) Proposed approach edge map

Figure 5.8: Evaluation with different lighting conditions

5.5 Evaluation with Different Camera Moving Speeds

As seen in the Figure 5.9 when the camera moving speed is low the results are more accurate and depth maps are converged clearly. When thehttps://www.overleaf.com/project/5de6 camera moving speed is high there is no time for convergence and edge maps are not created.

(a) Original image(speed 5km per hour

(b) Edge detection output

(c) Edge map

(d) Original image(speed 7km per hour

(e) Edge detection output

(f) Edge map

Figure 5.9: Evaluation with different camera moving speeds

5.6 Performance Evaluation by Profiling

Performance was evaluated by profiling all baseline algorithms with the proposed algorithm. The FPS rate for each algorithm is shown in the following Table 5.1. All the evaluations were done in an Intel core I3 computer with no Graphics Processing Unit(GPU) support. When the image quality is high it required more processing time. Based on the results, proposed algorithm performs better than all the baseline methods. Since new modifications were applied, FPS rate has been dropped when compared with the original REBVO algorithm.

Table 5.1: Scope and out of scope features

Image size	**LSD-SLAM**	**SGBM**	**REBVO**	**Proposed Method**
320 x 240	4	3	11	8
752 x 480	3	2	9	5
1280 x 720	3	1	7	4

Chapter 6

Conclusion

6.1 Introduction

This chapter focuses on the conclusions drawn upon the completion of the research. The research aim stated in Section 1.3 has been accomplished by using the technologies and tools that are mentioned in Section 4.2.

6.2 Conclusions about Research Questions

The main focus of this research is to detect wires for blind navigation in outdoor environments. This research gap was addressed by extending the REBVO algorithm by introducing an edge linking step and small connected component filtering steps. Results that are presented in Chapter 4 show the fact that the performance of the suggested approach is better than the existing baseline methods. This answers the first research question, "How to detect wires to aid blind navigation using Image processing and SLAM based approach?".

Camera trajectory and depth estimations were obtained by tracking the edge point's deviations from the previous frame to the current frame. Since there is no global edge map, less memory is needed when compared to other SLAM systems. Since the proposed method obtained the highest FPS rates it can be considered as the simplest algorithm among state of art methods. This answers the second research question, "How to extract information (depth, camera trajectory) from the environment while maintaining the simplicity of the algorithm?". [34]

6.3 Conclusions about research problem

As mentioned in section 1.2 the problem of lack of specifically defined accurate systems for the detection of thin-structured wires was addressed in this study. The goals of the research have been successfully achieved with the modification of the REBVO algorithm. Furthermore, the evaluation results show that the performance of the suggested algorithm is better than the existing state of art methods.

In this research, three algorithms were introduced to the scientific community and new research areas have been identified for future researchers. Apart from addressing the research problem, as a result of this research new specific datasets for thin wire detection was created. For further research studies and experiments, these datasets can be used easily.

6.4 Limitations

There are some limitation can be seen in the current developed solution. The proposed approach only works accurately in the light intensity between 1000LUX and 1500LUX. In other lighting intensities, the depth map was not converged. Integrating IMU data will be a good solution to this problem. Camera moving speed (walking speed) is another limitation of this research. As stated in Chapter 5 when the camera speed is high the accuracy of the system is reduced. So that camera speed inversely proportional to the accuracy of the system and it directly affects the convergence of the depth map. Currently, the suggested approach does not accurately detect the edges of moving objects and it only detects the edges of stable objects.

6.5 Implications for Future Research

There are several aspects can be considered for future researches based on this research. One aspect is to extend the current model to detect dynamic or moving wires. In real life, wires tend to have oscillations and shakings. Therefore, doing researches towards that direction would be valuable. The proposed system should be fused with anther obstacle detection and navigation system in order to use for blind navigation. Determining the fusion criteria will be complex and can be considered as a future aspect of this research. As explained in Section 5.5 accuracy of the system is highly dependent on the moving speed of the user. Therefore new methods and modifications can be introduced in the future in order used in speed moving users. Since this system is used by blind people, a thorough usability evaluation is required. In there future researchers can consider areas such as finding an optimal place for mounting the camera, accurate method of giving feedback about the obstacles to the users, finding the optimal camera angles to mount and etc. Since the current system is built using a monocular camera the depth estimation is not accurate so that stereo vision techniques can be introduced in the future for more accurate depth estimation.

References

[1] Who.int, "Vision impairment and blindness." https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment, 2019. [Online; accessed 20- Mar- 2019].

[2] I. Khan, S. Khusro, and I. Ullah, "Technology-assisted white cane: Evaluation and future directions," *PeerJ*, 2018.

[3] T. Alhmiedat, A. A. Taleb, and G. Samaraz, "A prototype navigation system for guiding blind people indoors using NXT mindstorms," *International Journal of Online Engineering*, 2013.

[4] C. S. Silva and S. P. Wimalaratne, "Towards Intelligent Sensor Fusion based Visually Impaired Navigation: An Assistive Technology Framework," in *Proceeding of the Inernational Conference on Emerging Trends in Artificial Intelligence*, 2015.

[5] Y. Priyadarshana and G. Wimalaratne, "Sensing Environment through Mobile: A Personalized Wearable Obstacle Detection System for Visually Impaired People," *Internatonal Journal of Soft Computing and Engineering*, vol. 3, no. 6, pp. 81–88, 2014.

[6] N. Kanwal, E. Bostanci, K. Currie, and A. F. Clark, "A Navigation System for the Visually Impaired: A Fusion of Vision and Depth Sensor," *Applied Bionics and Biomechanics*, 2015.

[7] R. Jiang, L. Qian, and S. Qu, "Let Blind People See: Real-Time Visual Recognition with Results Converted to 3D Audio," 2016.

[8] A. Kumar and A. Chourasia, "Blind Navigation System Using Artificial Intelligence," pp. 601–605, 2018.

[9] A. Bhowmick and S. M. Hazarika, "An insight into assistive technology for the visually impaired and blind people: state-of-the-art and future trends," 2017.

[10] R. Velázquez, E. Pissaloux, P. Rodrigo, M. Carrasco, N. I. Giannoccaro, and A. Lay-Ekuakille, "An outdoor navigation system for blind pedestrians using GPS and tactile-foot feedback," *Applied Sciences (Switzerland)*, 2018.

[11] Y. Tian, X. Yang, and A. Arditi, "Computer vision-based door detection for accessibility of unfamiliar environments to blind persons," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010.

[12] Y. Wang and K. J. Kuchenbecker, "HALO: Haptic Alerts for Low-hanging Obstacles in white cane navigation," in *Haptics Symposium 2012, HAPTICS 2012 - Proceedings*, 2012.

[13] R. Manduchi and S. Kurniawan, "Mobility-related accidents experienced by people with visual impairment," *Insight: Research and Practice in Visual Impairment and Blindness*, 2011.

[14] R. O'Keeffe, S. Gnecchi, S. Buckley, C. O'Murchu, A. Mathewson, S. Lesecq, and J. Foucault, "Long Range LiDAR Characterisation for Obstacle Detection for use by the Visually Impaired and Blind," in *Proceedings - Electronic Components and Technology Conference*, 2018.

[15] C. Zhou, J. Yang, C. Zhao, and G. Hua, "Fast, Accurate Thin-Structure Obstacle Detection for Autonomous Mobile Robots," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[16] M. Alshajajeer, M. Almousa, and Q. Abu Al-Haija, "Enhanced white cane for visually impaired people," *Journal of Applied Computer Science and Mathematics*, vol. 12, pp. 9–13, 01 2018.

[17] D. E. Gbenga, A. I. Shani, and A. L. Adekunle, "Smart Walking Stick for Visually Impaired People Using Ultrasonic Sensors and Arduino," *International Journal of Engineering and Technology*, 2017.

[18] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*, 2005.

[19] K. Hedenberg and B. Astrand, "A trinocular stereo system for detection of thin horizontal structures," in *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, pp. 211–218, 2008.

[20] M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "Fast robust monocular depth estimation for Obstacle Detection with fully convolutional networks," in *IEEE International Conference on Intelligent Robots and Systems*, 2016.

[21] T. Mori and S. Scherer, "First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013.

[22] J. J. Tarrio and S. Pedre, "Realtime edge-based visual odometry for a monocular camera," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 702–710, 2015.

[23] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics," *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.

[24] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, Nov 2007.

[25] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, pp. 1052–67, 07 2007.

[26] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, p. 1147–1163, Oct 2015.

[27] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15–22, 2014.

[28] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[29] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 International Conference on Computer Vision*, pp. 2320–2327, Nov 2011.

[30] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), (Cham), pp. 834–849, Springer International Publishing, 2014.

[31] S. Saluja, A. K. Singh, and S. Agrawal, "A Study of Edge-Detection Methods," vol. 2, no. 1, p. 6, 2013.

[32] M. Sharifi, M. Fathy, and M. Mahmoudi, "A classified and comparative study of edge detection algorithms," in *Proceedings. International Conference on Information Technology: Coding and Computing*, pp. 117–120, IEEE Comput. Soc.

[33] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[34] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision – ECCV 2006* (A. Leonardis, H. Bischof, and A. Pinz, eds.), (Berlin, Heidelberg), pp. 404–417, Springer Berlin Heidelberg, 2006.

Appendices

Appendix A

Overall Process Flow

Figure A.1 shows the higher level diagram of suggested SLAM system

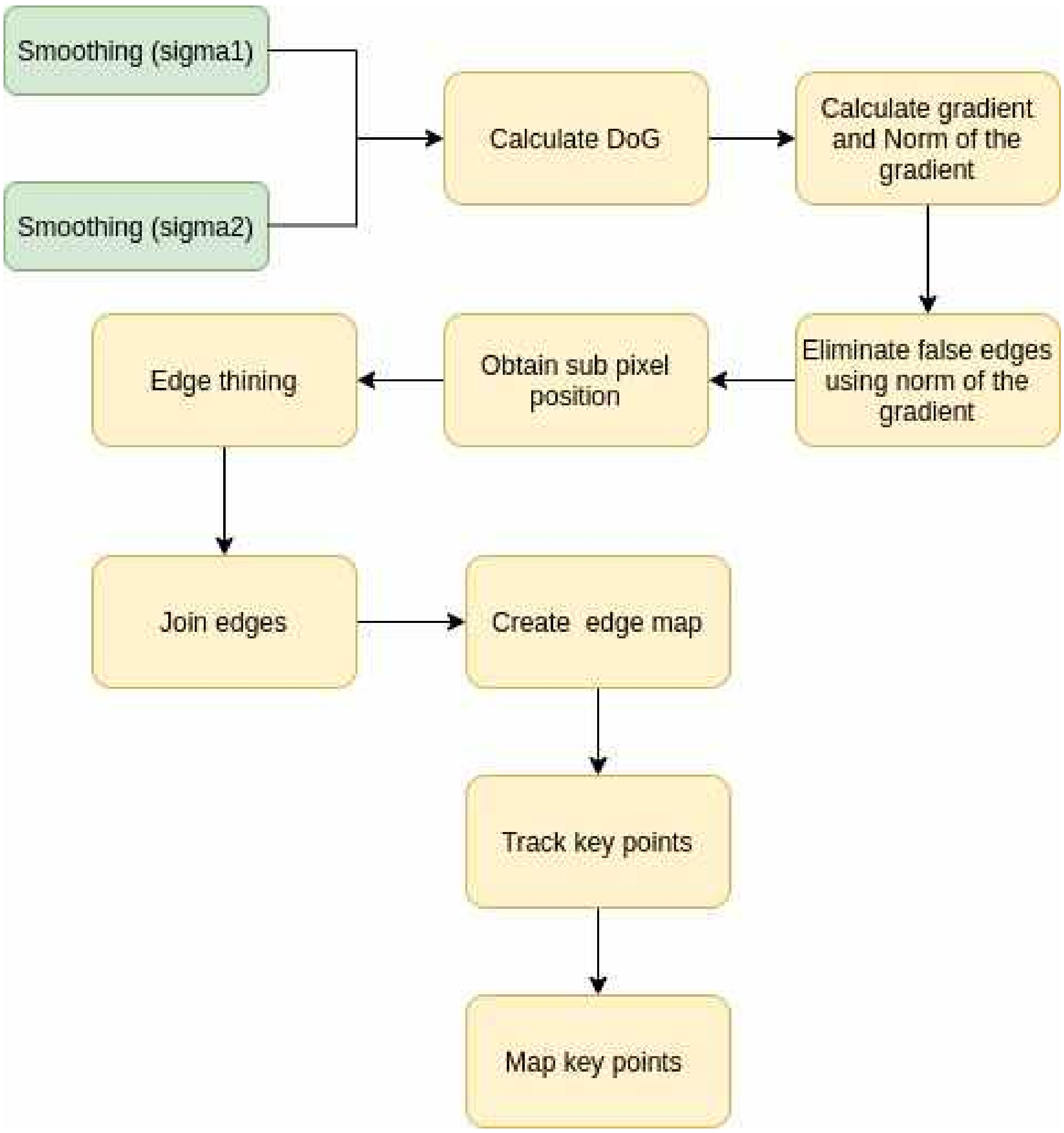


Figure A.1: Overall process flow diagram

Appendix B

Edge Linking Algorithm

Searching for the next edge was performed on the gradient directions. ty is the y directional gradient and tx is the x directional gradient. Figure B.1 shows the searching criteria for finding the neighbouring edges up to a maximum of d pixels.

```
if (ty > 0) {
    if (tx > 0) {
        for (int i = 1; i < searchForMaxD; i++) {
            if ((kl_inx = img_mask(x + i, y + 0)) >= 0) {
                return kl_inx;
            }
        }
        for (int i = 1; i < searchForMaxD; i++){
            if ((kl_inx = img_mask(x, y + i)) >= 0){
                return kl_inx;
            }
        }
        for (int i = 1; i < searchForMaxD; i++){
            if ((kl_inx = img_mask(x + i, y + i)) >= 0){
                return kl_inx;
            }
        }
    } else {
        for (int i = 1; i < searchForMaxD; i++) {
            if ((kl_inx = img_mask(x - i, y)) >= 0) {
                return kl_inx;
            }
        }
        for (int i = 1; i < searchForMaxD; i++) {
            if ((kl_inx = img_mask(x, y + i)) >= 0) {
                return kl_inx;
            }
        }
        for (int i = 1; i < searchForMaxD; i++){
            if ((kl_inx = img_mask(x - i, y + i)) >= 0) {
                return kl_inx;
            }
        }
    }
```

Figure B.1: Searching criteria if gradient Y is positive

```
        }
} else {
    if (tx < 0){
        for (int i = 1; i < searchForMaxD; i++) {
            if ((kl_inx = img_mask(x - i, y)) >= 0){
                return kl_inx;
            }
        }
        for (int i = 1; i < searchForMaxD; i++}{
            if ((kl_inx = img_mask(x, y - i)) >= 0)
            {
                return kl_inx;
            }
        }
        for (int i = 1; i < searchForMaxD; i++) {
            if ((kl_inx = img_mask(x - i, y - i)) >= 0) {
                return kl_inx;
            }
        }
    } else  {
        for (int i = 1; i < searchForMaxD; i++){
            if ((kl_inx = img_mask(x + i, y)) >= 0) {
                return kl_inx;
            }
        }
        for (int i = 1; i < searchForMaxD; i++) {
            if ((kl_inx = img_mask(x, y - i)) >= 0) {
                return kl_inx;
            }
        } for (int i = 1; i < searchForMaxD; i++) {
            if ((kl_inx = img_mask(x + i, y - i)) >= 0) {
                return kl_inx;
            }
        }
    }
}
return -1; //-1 if no match
```

Figure B.2: Searching criteria if gradient Y is negative