DEVELOPMENT OF AN IMPROVED ALGORITHM FOR POWER LINE DETECTION IN OPTICAL IMAGES USING FRANGI FILTER AND FIRST ORDER DERIVATIVE OF GAUSSIAN

BY

Prosper Obaro OYIBO

Department of Electrical and Computer Engineering,

Faculty of Engineering

Ahmadu Bello University,

Zaria.

June, 2017

DEVELOPMENT OF AN IMPROVED ALGORITHM FOR POWER LINE DETECTION IN OPTICAL IMAGES USING FRANGI FILTER AND FIRST ORDER DERIVATIVE OF GAUSSIAN

BY

Prosper Obaro OYIBO, B.Eng. (ABU, 2014)

P14EGEE8076

A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATESTUDIES, AHMADU BELLO UNIVERSITY, ZARIA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF A MASTER OF SCIENCE (M.Sc.) DEGREE IN CONTROL ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING,
FACULTY OF ENGINEERING
AHMADU BELLO UNIVERSITY,
ZARIA

June, 2017

DECLARATION

I Prosper ObaroOYIBO, hereby declare that the work in this dissertation entitled Development of an Improved Algorithm for Power Line Detection in Optical Images using Frangi Filter and First Order Derivative of Gaussianhas been carried out by me in the Department of Electrical and Computer Engineering. The information derived from literature has been duly acknowledged in the text and a list of references provided. No part of this dissertation was previously presented for another degree or diploma at this or any other institution.

Prosper ObaroOYIBO _____ _____

Signature **Date**

CERTIFICATION

This Dissertation entitled DEVELOPMENT OF AN IMPROVED ALGORITHM FOR POWER LINE DETECTION IN OPTICAL IMAGESUSING FRANGI FILTER AND FIRST ORDER DERIVATIVE OF GAUSSIAN by Prosper Obaro OYIBO meets the regulations governing the award of degree of Master of Science (M.Sc.) in Control Engineering of the Ahmadu Bello University, and is approved for its contribution to knowledge and literary presentation.

Prof. Boyi Jimoh	_____	_____
Chairman, Supervisory Committee	Signature	Date
Prof. M. B. Mu'azu	_____	_____
Member, Supervisory Committee	Signature	Date
Dr. Y. Jibril	_____	_____
Head of Department	Signature	Date
Prof. S. Z. Abubakar	_____	_____
Dean. School of Postgraduate Studies	Signature	Date

DEDICATION

This research work is dedicated to my Parents and Siblings.

ACKNOWLEDGEMENT

I express my unreserved gratitude andappreciation to God for seeing me through the completion of this research work.

I wish to express my deepest gratitude to my supervisory team and mentors:Prof. Boyi Jimoh and Prof. M. B. Mu'azu, for their tireless efforts, valuable guidance and constant supervision towards the success of this research. The completion of this research could not have been possible without their constant participation and assistance. Thank you very much Sirs. My deep appreciation also goes to the Computer & Control Research Group for their keen interest, suggestions and constructive criticisms during all the stages of this research.

I acknowledge with thanks all the lecturers of Electrical and Computer Engineering, Ahmadu Bello University, namely: Prof.B. G. Bajoga, Dr. E.A. Adedokun, Dr. A. M. S. Tekanyi, Dr. S. M. Sani, Dr. A. D. Usman and Dr. S. Garba, Dr. Y. Jibril, Dr. T.H Sikiru, Dr. K.A. Abu-Bilal, Dr. E. A. Gbenga,Engr. M. J. Mu'azu, Engr. A. I. Abdullahi, Engr. E. Okafor, Engr. Bashir Sadiq, Engr.Salawudeen A. Tijjani, Engr.Olaniyan Abdulrahman, Engr. Yusuf and most especially, those whose names could not be mentioned. My deepest appreciation also goes to Abdullahi Tukur for all the administrative support towards me and the entire students of Department of Electrical and Computer Engineering.

I am also thankful to my special friends, Esther Ejiofor,Mr. & Mrs G. U. Igelige, Abdulmalik Ibrahim, Abdumalik Fahd, Alhassan, Engr. Suleiman Hussein, Ocholi Haruna, Ibrahim Abdulwahab, Magaji Suleiman, Wasiu Ajadi, Malachi Egbuga, Oga Abubakar, Shittu Danraka, Engr. Ovie, Bukola Ishola, Salamatu Bello, Oziabor Loveth, Anuoluwapo Babarinde, Blessing Oghenekowhegba, Abu Samuel, Alao Micheal, Abdu-Samaila Thomas and Yusuf Muktar. Their continuous support and contributions toward the success of this researchwould never be forgotten, may God reward them and strengthen our friendship. I am very much thankful to all my course mates, Atuman, Muktar, Zaharadeen, Kachalla, Kabir, Muhideen, Ajayi, Usman, Akilu and Balarabe, may God bless and reward you all. Amen.

Above all, I am forever indebted to my dear parents Engr & Mrs J.E. Oyibo, your endless love and support, kind and understanding personality will always be appreciated. Thank you very much. To my siblings, Engr. V. E. Oyibo, Engr. T. O. Oyibo, Dr. S. T. Oyibo, Dr. K. O. Oyibo. I owe you all a big thank you. I am highly grateful for your constant support. May God bless you all.

Prosper ObaroOYIBO
June, 2017.

ABSTRACT

This research presents the development of a Frangi filter and first-order derivative of Gaussian (FF-FDOG) based power line detection (PLD) algorithm as an improvement to the standard PLD algorithm. Vision-based PLD is important in obstacle avoidance in low-altitude flight and also in the surveillance and maintenance of electrical infrastructure. The need for high and real-time detection rates as well as low false alarm in noisy and cluttered images makes it a challenging task. Matched filterand first-order derivative of Gaussian (MF-FDOG) based PLD algorithmwas developed to handle limitations associated with the standard PLD algorithm in terms of its ability to automatically select a problem specific threshold in its edge detection. The MF-FDOG based PLD, however, returned a high false positive rate and a detection rate insufficient for real time processing.The FF-FDOG based threshold is developed in this work using frangi filter (which detects vessel based on the eigen value analysis of the second order structure of an image) and FDOG filter. Images from the University of South Florida computer vision and pattern recognition group wire database were used to evaluate the performance of the developed FF-FDOG method. In the results obtained, the true positive rate of the developed FF-FDOG based PLD algorithm was 86.39%, which is a 2.64% improvement over MF-FDOG's 84.16%, while the false positive rate of the developed FF-FDOG based PLD algorithm was 11.45%, which is a 36.06% improvement over MF-FDOG's 17.91%.

TABLE OF CONTENTS

DEVELOPMENT OF AN IMPROVED POWER LINE DETECTION ALGORITHM FOR OPTICAL IMAGES USING FRANGI FILTER AND FIRST ORDER DERIVATIVE OF GAUSSIAN

DECLARATION ii

CERTIFICATION iii

DEDICATION iv

ACKNOWLEDGEMENT v

ABSTRACT vi

TABLE OF CONTENTS vii

LIST OF FIGURES x

LIST OF PLATES xi

LIST OF TABLES xiii

ABBREVIATIONS xiv

CHAPTER ONE 1

INTRODUCTION 1

1.1 Background 1

1.2 Statement of the Problem 3

1.3 Significance of the Study 3

1.4 Aim and Objectives of the Study 3

1.5 Methodology 4

1.6 Dissertation Organization 5

CHAPTER TWO 7

LITERATURE REVIEW 7

2.1 Introduction 7

2.2 Review of Fundamental Concepts 7

2.2.1 Power line detection 7

2.2.2 Power line detection algorithm 9

2.2.3 MF-FDOG power line segment detection 10

2.2.4 Multiscale vessel enhancement filter: Frangi filter 15

2.2.5 Graph-cut model based line detection 22

2.2.6 Morphological operations 25
2.2.7 Performance metrics 28
2.3 Review of Similar Works 30
CHAPTER THREE 37
MATERIALS AND METHODS 37
3.1 Introduction 37
3.2 Experimental data. 37
3.2.1 Image Acquisition. 37
3.2.2 Clutter measure 40
3.2.3 Parameters 41
3.3 Replication of the MF-FDOG based PLD algorithm 42
3.3.1 Pre-processing the Image 42
3.3.2 Matched filtering 42
3.3.3 First Order derivative of Gaussian filtering 43
3.3.4 MF-FDOG thresholding 45
3.3.5 Morphological filtering 45
3.3.6 Line segment clustering 46
3.4 Development of the FF-FDOG based PLD algorithm 47
3.4.1 Frangi filtering 48
3.4.2 FF-FDOG thresholding 49
3.5 Performance measures 50
CHAPTER FOUR 52
RESULTS AND DISCUSSION 52
4.1 Introduction 52
4.2 Clutter Measure 52
4.3 Experimental Results 53
4.3.1 Visualized results 53
4.3.2 Quantified results 64
4.4 Discussion 65
4.5 Application of FF-FDOG on captured images 65

CHAPTER FIVE 70
SUMMARY, CONCLUSION AND RECOMMENDATIONS 70
5.1 Summary 70
5.2 Conclusion 70
5.3 Significant Contributions 71
5.4 Limitations 71
5.5 Recommendations for Further Work 71
REFERENCES 72
APPENDIX A1 77
MATLAB code for MF-FDOG PLD algorithm 77
APPENDIX A2 81
MATLAB code for FF-FDOG PLD algorithm 81
APPENDIX B1 85
MATLAB code for matched filtering 85
APPENDIX B2 86
MATLAB code for frangi filtering 86
APPENDIX B3 89
MATLAB code for first order derivative of Gaussian filtering 89
APPENDIX C1 90
MATLAB code for performance measure 90
APPENDIX C2 92
Data for MF-FDOG performance measure 92
APPENDIX C3 94
Data for FF-FDOG performance measure 94
APPENDIX D1 96
MATLAB code for clutter measure 96
APPENDIX D2 97
Data for clutter measure 97

LIST OF FIGURES

Figure 2.1: Overview of MF-FDOG Power Line Detection Algorithm 11

Figure 2.2: The Profiles of Two Kinds of Edges 12

Figure 2.3: Responses of MF and FDOG to the Two Kinds of Edges 13

Figure 2.4: The Second Order Information of a Gaussian Kernel 17

Figure 2.5: Hit-miss Operation 26

Figure 3.1: Matched Filter Kernel 42

Figure 3.2: FDOG Filter Kernel 44

Figure 3.3: Overview of FF-FDOG Power Line Detection Algorithm 48

Figure 4.1: ClutterStatistical Histogramof Power line Image Dataset 52

LIST OF PLATES

Plate 2.1: Challenging Sample Images from Dataset 8

Plate 3.1: Sample Images of Power Lines downloaded from USF Wire Database 38

Plate 3.2: Captured Images of Power Lines in Samaru Zaria 39

Plate 4.1. Grayscale of Sample Images 54

Plate 4.2: Matched Filter Response of Sample Images 55

Plate 4.3: Frangi Filter Response of Sample Images 56

Plate 4.4: First Order Derivative of Gaussian Filter Response of Sample Images 57

Plate 4.5: Edgemap of Sample Image 1 58

Plate 4.6:Edgemap of Sample Image 2 58

Plate 4.7: Edgemap of Sample Image 3 58

Plate 4.8: Edgemap of Sample Image 4 59

Plate 4.9: Edgemap of Sample Image 5 59

Plate 4.10: Edgemap of Sample Image 6 59

Plate 4.11:Line Segment Pool of Sample Image 1 60

Plate 4.12: Line Segment Pool of Sample Image 2 60

Plate 4.13: Line Segment Pool of Sample Image 3 60

Plate 4.14: Line Segment Pool of Sample Image 4 61

Plate 4.15: Line Segment Pool of Sample Image 5 61

Plate 4.16: Line Segment Pool of Sample Image 6 61

Plate 4.17: PLD Result of Sample Image 1 62

Plate 4.18: PLD Result of Sample Image 2 62

Plate 4.19: PLD Result of Sample Image 3 63

Plate 4.20: PLD Result of Sample Image 4 63

Plate 4.21: PLD Result of Sample Image 5 63

Plate 4.22: PLD Result of Sample Image 6 64

Plate 4.23: Captured Image 1 66

Plate 4.24: Captured Image 2 66

Plate 4.25: Captured Image 3 66

Plate 4.26: Captured Image 4 67

Plate 4.27: Captured Image 5 67

Plate 4.28: Captured Image 6 67

Plate 4.29: Captured Image 7 68

Plate 4.30: Captured Image 8 68

Plate 4.31: Captured Image 9 68

Plate 4.32: Captured Image 10 69

LIST OF TABLES

Table 2.1: Possible Patterns of theEigenvalues λ_k in 2D and 3D 18

Table 3.1: The Contingency Table for PLD 50

Table 4.1: Power Line Detection Result of USF Dataset 64

ABBREVIATIONS

Acronyms	Definition
Acc	accuracy
AUC	area under the curve
CBS	Circle Based Search
DSA	Digital Subtraction Angiography
FCM	Fuzzy C-means
FDOG	first order derivative of Gaussian
FF	Frangi filter
FF-FDOG	Frangi filter and first order derivative of Gaussian
FN	false negative
FOV	field of view
FP	false positive
FPR	false positive rate
HIS	hue-intensity-saturation
HSV	hue-saturation-value
HT	Hough transform
IVRT	Integrated Vector Radon Transform
LSD	line segment detector
MF	matched filter
MF-FDOG	matched filter and first order derivative of Gaussian
MRA	Magnetic Resonance Angiogram
N-cut	Normalised cut
NIR	near-infrared
NTSB	National Transportation Safety Board
PLD	power line detection
PPV	positive predictive value
RGB	red-green-blue
ROC	receiver operating characteristic
SIFT	Scale-invariant feature transform

SN	Sensitivity
SP	Specificity
TN	true negative
TP	true positive
TPR	true positive rate
UAV	Unmanned Ariel Vehicle
U.S.	United State
USF	University of South Florida

CHAPTER ONE

INTRODUCTION

1.1 Background

Electricity is vital for the activities of modern-day societies and effective monitoring and maintenance of power lines are needed to secure uninterrupted distribution of electricity, (Matikainen *et al.*, 2016). Electricity companies spend a significant budget on power line inspections, and continuously pursue new approaches to reduce inspection cost (Martinez *et al.*, 2014). For example, Ergon Energy, one of the top electricity companies in Australia, spends $80 million a year inspecting and managing vegetation that encroaches on power line assets (Li *et al.*, 2010). Inspection of high voltage power lines can be very dangerous if performed by humans, very expensive if performed by helicopters and damaging on the cable if performed by the roll robot. Unmanned aerial vehicle (UAV) is, therefore, one of the best instruments for detailed power line inspection tasks (Zhou *et al.*, 2016).

Thin objects such as cables, power lines, and wires (whose position cannot be guaranteed as known before flight) are not easily perceived by helicopters and small UAV pilots over heavily cluttered backgrounds, or when the contrast between the object and the background is low (Byrne *et al.*, 2006; Candamo *et al.*, 2009; Luo *et al.*, 2014). Power line detection is a widely sought environmental awareness techniques and it is of great significance in ensuring the safety of low altitude flight which is essential in a variety of applications (such as traffic and power infrastructure monitoring, border patrol, search and rescue, and surveillance), as even a low-speed collision with power lines can be fatal (Gandhi *et al.*, 2003; McGee *et al.*, 2005; Scherer *et al.*, 2008).

In particular, comprehensive real data suggests that more than half of the low altitude aerial collision accidents are caused by power lines(Shan *et al.*, 2015). The United State (U.S.)

Army reports that more helicopters have been lost due to hitting power lines than are to combat(Avizonis & Barron, 1999). Between 1997 and 2006 the U.S. Army recorded 54 so-called “wire strikes” in which 13 military personnel died and which caused $224 million in damages. Also in the same period there were 102 civilian wire strikes, killing 33 people, according to data from the National Transportation Safety Board (NTSB)(Safe Flight, 2009).

All these highlightthe need forenhanced capabilities in power line detection(PLD) techniques that should provide a high probability of timely detection while maintaining a low probability of false alarm in noisy, cluttered images of power lines exhibiting a wide range of sizes and complexities (Kasturi & Camps, 2002).

Continued advances in the fields of image-processing and computer vision have raised interest in their suitability in PLD for collision avoidance and also for power line tracking during inspection with UAVs (Gandhi *et al.*, 2003; Seibold *et al.*, 2013). Optical based power line detection algorithms generally capture two criteria(local and global criterion)in which a threshold is set manually for its edge detection in the local criterion (Bhujade *et al.*, 2013; Cao *et al.*, 2013; Liu *et al.*, 2012; Seibold *et al.*, 2013; Yang *et al.*, 2012; Zhang *et al.*, 2012; Zhu *et al.*, 2013). However, a fixed threshold are only effective when the background is relatively monotonic thus they may cause these algorithms to fail with complex and changing backgrounds (Zhou *et al.*, 2016).A problem-specific threshold schemewas developed bySong & Li (2014),to overly detect power line segments which was a variant of a retinal blood vessel detection method based on the extension of matched filter (MF), proposed by Zhang *et al.* (2010). The thresholding scheme was able to detect symmetrical edges and suppress step edges in images of powerline but it still produced a high false alarm.

To this end, Frangi filter (FF), a vessel enhancement filter, in which vesselness measure is obtained on the basis of all eigenvalues of the multiscale second order local structure of an

image (Frangi *et al.*, 1998), and first order derivative of Gaussian(FDOG) is employed in the local criterion of the PLD algorithm in this research.

1.2 Statement of the Problem

Optical based power line detection algorithms generally capture two criteria (local and global criterion) in which a threshold is set manually for its edge detection in the local criterion. However, a fixed threshold are only effective when the background is relatively monotonic thus they may cause these algorithms to fail with complex and changing backgrounds. Anadaptive threshold schemebased on the image response to MF and FDOG was introduced to mitigate the problem of fixed threshold. However, it is not without its own drawbacks as it returned a high false positive rate and a detection rate insufficient for real time processing.This research is aimed at developing an adaptive threshold scheme based on the image response to FF and FDOG which will maximise the true positive detection rate and minimise the false positive detection rate and runtime.

1.3 Significance of the Study

Power line detection is a widely sought environmental awareness techniques which reduce significantly, the budget Electricity companies spend on power line inspections, and the occurrence of “wire strikes” in low altitude flight, thus preventing loss of lives and properties. This Study offers a FF and FDOG based power line detection algorithm whichprovides a high probability of timely detection while maintaining a low probability of false alarm in noisy, cluttered images of power lines exhibiting a wide range of sizes and complexities

1.4 Aim and Objectives of the Study

The aim of this research work is the development of an improvedalgorithm for PLD in optical images using the Frangi filter and first order derivative of Gaussian (FF-FDOG).

To achieve the stated aim, a number of objectives have been identified as follows:

1. To replicate the standard matched filter and first order derivative of Gaussian (MF-FDOG)and develop the FF-FDOG based power line detection algorithms
2. To compare the performance of the FF-FDOG and MF-FDOG based PLD algorithmsusing true positive rate, false positive rate and run time as metrics using the work of Song & Li (2014).
3. To apply the FF-FDOG based PLD algorithm on select dataset of 10 images of power lines obtained under different conditions from Samaru, Sabon-Gari Local Government Area of Kaduna State.

1.5 Methodology

The methodology adopted in this research is described as follows:

1. Replication and implementation of the MF-FDOG based PLD algorithm by adopting the following:
    a) An edge map based on MF and FDOG is applied to detect all the line segments with symmetrical edges in the image.
    b) A morphological filter is designed specifically to filter out the non-power line candidates.
    c) The graph-cut model based on the graph theory is exploited to group the line segment pool into whole line pool
    d) The “true” power lines is picked-up by morphology properties.
2. Development and implementation of the FF-FDOG based PLD algorithm by adopting the following:
    a) An edge map based on FF and FDOG is applied to detect all the line segments with symmetrical edges in the image
    b) Step 1 (b - d) is repeated.

3. Validation by comparing the performance of both algorithm by adopting the following:
    a) A dataset of 50 images of power lines is obtained from the University of South Florida, computer vision and pattern recognition wire database.
    b) For each image, clutter measure on the visual complexity of the background is calculated to quantitatively show the richness of the dataset
    c) For each power line, the ground truth is obtained by first manually labelling several points, and then approximating with a straight line or a quadratic polynomial.
    d) The performance of both algorithms are measured by true positive rate (TPR) and false positive rate (FPR) and run-time at line-level.
    e) Comparison based on TPR, FPR and run time of both algorithms.
4. Application of the FF-FDOG based PLD algorithm on selected images under different conditions by adopting the following:
    a) A dataset of 10 images of power lines having resolution of 720×480 pixels, with different noise level, ambiguity, weather conditions, backgrounds, is obtained from Samaru, Sabon-Gari Local Government Area of Kaduna State using a Digital Camera.
    b) The FF-FDOG based PLD algorithms is applied on the images.

1.6 Dissertation Organization

The general introduction has been presented in Chapter One. The rest of the chapters are structured as follows: Firstly, detailed review of related literature and relevant fundamental concepts ofPLD, MF-FDOG based PLD algorithm,multiscale vessel enhancement filter,graph-cut model based line detection, and morphological operations are discussed in Chapter Two. Secondly, an in-depth approach and relevant mathematical models describing

the acquired image data, the replication of the MF-FDOG based PLD algorithm and the development of the FF-FDOG based PLD algorithm are presented in Chapter Three. Next, the analysis, performance and discussion of the result are shown in Chapter Four. Finally, summary, conclusion and recommendations for further work are presented Chapter Five. The list of cited references and MATLAB codes are provided in the appendices at the end of this dissertation.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The literature review comprises of the review of fundamental concepts and the review of similar works. In the review, most of the fundamental theories and existing works relevant to the success of this research are discussed.

2.2 Review of Fundamental Concepts

In this section, concepts fundamental to the research such as power line detection (PLD), PLD algorithm, matched filter and first order derivative of Gaussian (MF-FDOG)edge detection, Frangi filter (FF), graph-cut model based line detection, morphological operations and performance measures amongst others are reviewed.

2.2.1 Power line detection

Unlike other ground obstacles such as trees and human buildings, power lines appear extremely small from a distance, which results in great difficulties for them to being sensed(Candamo *et al.*, 2009)**.** Viewing from this standpoint, the PLD problem turns into a small targets detection problem, where the power lines, as the small targets, are very thin and possess few visual features. Moreover, the power lines always keep a still stance, so the motion features are not prominent, which may even complicate the PLD problem (Shan *et al.*, 2015).

Correspondingly, many sensor-based strategies for automatic PLD have been developed. Popular among these systems are infrared system (Yamamoto & Yamada, 1997), laser radar system (Essen *et al.,* 2002; Garcia-Pardo *et al.,* 2002) and electromagnetic system(Greene *et al.,* 2013).In infrared system, low contrast between the background and power lines may hinder automatic detection even if high-resolution infrared camera is used; moreover, power

failure or environments with some heat sources may also impede this technology. Laser radar system is highly sensitive to atmospheric attenuation effects,thus easily affected by weather conditions. Also, electromagnetic system may be dysfunctional in the case of outage. Beyond that, the three kinds of emissive devices used in these systems are most energy consuming, big size and heavy(Song & Li, 2014).

Over the years, the optical-based system has received more and more attention due to their low energy consumption, small size and light weight, which enable small aircraft to overcome the significant limitations in electrical power supply and payload (Song & Li, 2014). However, perceiving power lines from optical image is a challenging problem due to heavily cluttered background, low contrast between the object and the background and the tiny sizes and thin structures of the power lines as shown by the sample images in Plate. 2.1.



Plate 2.1: Challenging Sample Images from Dataset.

In resolving these general challenges, computer vision plays a major role as different aspects of UAV-based obstacle avoidance and electrical infrastructure monitoring can benefit greatly from appropriate vision based PLD algorithms(Martinez *et al.*, 2014).

2.2.2 Power line detection algorithm

The existing PLD algorithms for optical image can be classified into two kinds by their applications. One is for power line tracking in automatic surveillance and inspection of the electrical power infrastructure, and the other is for obstacle avoidance in UAV navigation. PLD for obstacle avoidance in low altitudeflight is much more difficult and challenging than for electrical power infrastructure surveillance because the aerial vehicles are operated in unknown scenario and little information about the threatening power lines may be known beforehand. Besides, the PLD algorithms should be fast to leave enough time for avoiding the obstacle (Song & Li, 2014).

A state-of-the-art review of vision based PLD algorithms for electrical infrastructure monitoring was presented by Mirallès *et al.* (2014)and Matikainen *et al.* (2016).The algorithms depended on some prior knowledge of the power lines. For example, the number of power lines were known. Also,they assumed a downward view of the power lines making them approximately straight lines and parallel to each other.

In spite of huge challenges in PLD for threatening obstacle avoidance, there have been a few of researchers involved in related investigation. In this case, power lines look more like arc curves than straight lines, since the power lines usually hang down for force of gravity.To the best of our knowledge, the premier work was introduced byKasturi & Camps (2002). Line segments were extracted using Steger's detector of curvilinear structures (Steger, 1998) prior to line detection by Hough transform (HT). Although, this algorithm was with sub-pixel accuracy it was however time consuming.Candamo *et al.* (2009),exploited the temporal

information carried between different frames in the low altitude videos. Firstly, they obtained a feature map image by estimating the relative motion of each pixel utilising an optical flow method from consecutive frames. Next, HT was performed in each of sub window of the feature map image, prior to predicting the locations of the previously detected lines in the next frame by tracking the parameter space of HT over time using a linear motion model. However this did not benefit real time processing as the data capacity of the video was huge and the computation complexity of the optical flow algorithm was high.Song & Li (2014), utilized the matched filter (MF) and first-order derivative of Gaussian (FDOG) to detect power line segments with symmetric properties prior to grouping the line segments into whole power lines based on a graph-cut model. The algorithm, though was able to detect curved power lines but returned high false positive rate. More recently Shan *et al.* (2015) made use of a pylon-line spatial context to link lines from a line segment pool extracted utilizing the line segment detector (LSD) algorithm, with power line pylons detected with aScale-invariant feature transform(SIFT) based approach, thus forming a power line seed segment for an incremental search based line-line spatial context to detect whole power lines. However the algorithm was time consuming and depended highly on the presence of power pylon in the image view. The above reported schemes generally capture two criteria.

1. Local criterion: creating a line segment pool involving local operator or pixel-wise manipulation mainly based on the gradient image.
2. Global criterion: grouping “true” candidates in pool into whole power lines by incorporating additional knowledge about their structures, such as the smoothness.

2.2.3 MF-FDOG power line segment detection

Matched filter and first order derivative of Gaussianpower-line segments detection method was introduced bySong & Li (2014) and is based on the variant of MF andFDOG. MF has been used in the detection of various line features in the retinal blood vessels extraction

(Chaudhuri *et al.*, 1989). It is an effective yet simple method, which detects vessel-like feature by filtering image and thresholding on its response image, exploiting the characteristics that the crosssection of a vessel is Gaussian shaped and symmetric about its peak position(Song & Li, 2014). However due to the fact that MF may respond to not only vessel edges but also non-vessel edges,Zhang *et al.* (2010) proposed a retinal blood vessel detection method based on the extension of MF, which is called MF-FDOG. The MF-FDOG is the same as MF approach detecting vessels by thresholding on the response image to MF, but with a self-adjusting threshold based on the image responseto FDOG as shown in Figure 2.1.

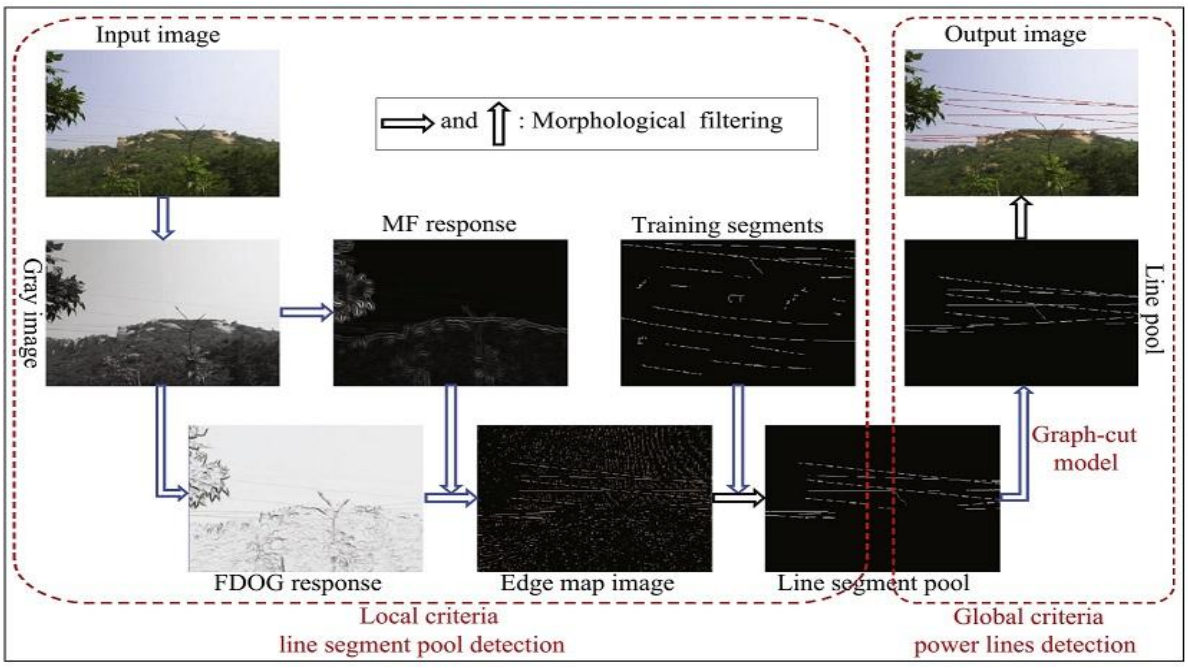


Figure 2.1: Overview of MF-FDOG Power Line Detection Algorithm (Song & Li, 2014)

The MF-FDOG method achieved competitive detection results when being compared with those state of the art schemes but with much lower complexity.

2.2.3.1 MF-FDOG definition

Matched filter is a Gaussian-shaped filter defined as(Song & Li, 2014)

$$F(x,y)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{x^2}{2\sigma^2}\right)-s, \qquad \text{for } |x|\le t\cdot\sigma, \quad |y|\le L/2 \tag{2.1}$$

Where (x,y) are the pixel coordinate; σ denotes the scale of filters; t is a constant and is usually set to 3 because larger than 99% confidence interval is $[-3\sigma,3\sigma]$; L represents the length of the neighbourhood along the y-axis to smooth noise; In order to remove the smooth background after filtering by normalizing the mean value of the filter to 0, s is introduced as(Song & Li, 2014):

$$s=\left(\int_{-t\sigma}^{t\sigma}\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)dx\right)\Big/(2t\cdot\sigma) \tag{2.2}$$

It can be derived without much exertion that the first order derivative of the Gaussian (FDOG) function is as follows(Song & Li, 2014):

$$f(x,y)=-\frac{x}{\sqrt{2\pi}\sigma^3}\exp\left(-\frac{x^2}{2\sigma^2}\right), \quad \text{for } |x|\le t\cdot\sigma, \quad |y|\le L/2 \tag{2.3}$$

Given two synthetic signals representing the profiles of two kinds of edges (a symmetrical and an ideal step edge), shown in Figure 2.2 (a) and (b),

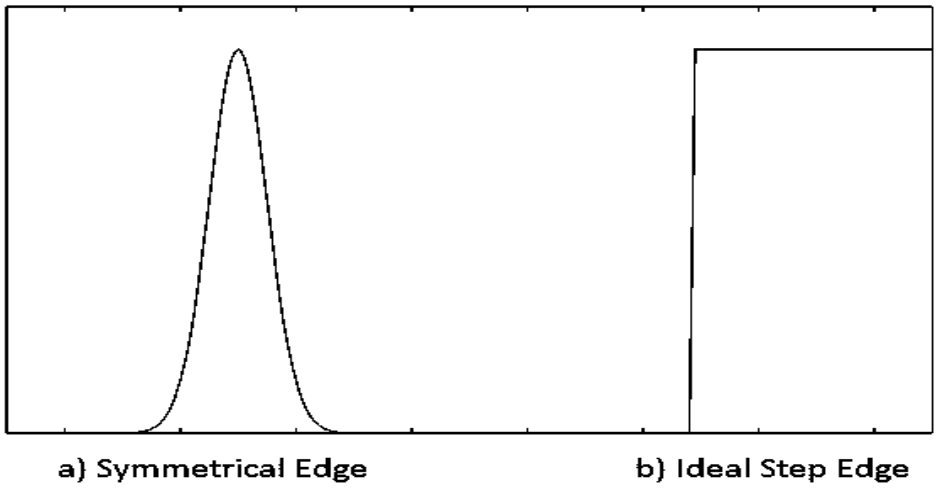


Figure 2.2: The Profiles of Two Kinds of Edges (Song & Li, 2014)

The corresponding responses of MF and FDOG are given in Figure 2.3. It demonstrates that for MF, the response of the symmetrical edge is still symmetrical and strong positive in the peak area of the edge, while the response of the step edge is anti-symmetrical and very low in the jump area (Figure 2.3-b); for FDOG, the opposite is true (Figure 2.3-d) (Song & Li, 2014).

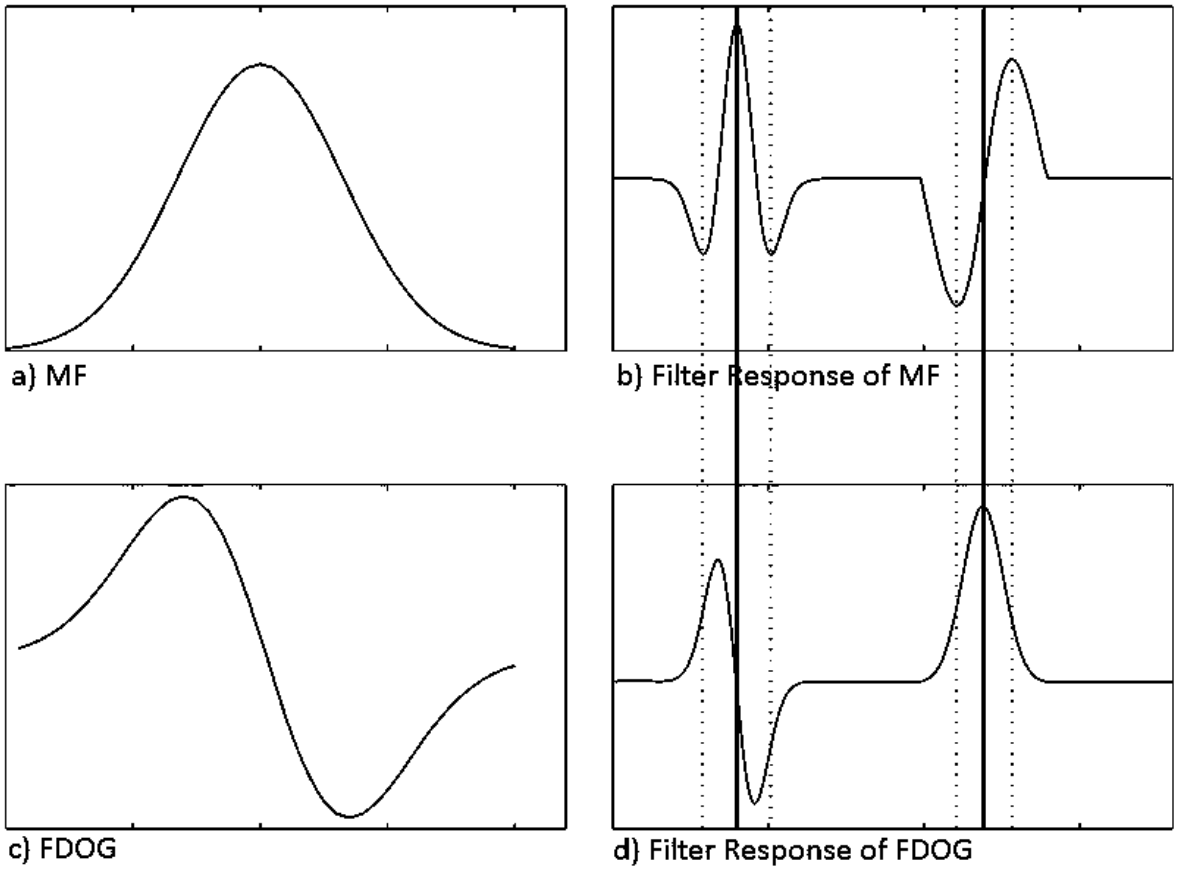


Figure 2.3: Responses of MF and FDOG to the Two Kinds of Edges(Song & Li, 2014)

2.2.3.2 MF-FDOG thresh-holding scheme

The MF-FDOG thresh-holding scheme, proposed by Zhang *et al.* (2010), for retinal blood vessel detection was based on the idea that the retinal vessels belong to symmetrical edges. The edge map image of the retinal image is obtained when the response to MF is larger than a given threshold in the original MF method. However, it can also match the non-vessel edges

to some extent. In Zhang *et al.* (2010), the threshold is adjusted by the response to FDOG: if there is a vessel in the image, the threshold will be lowered as a result of a weak FDOG response at the corresponding area, thus detecting vessel; otherwise, the threshold will be raised by the strong FDOG response, thus suppressing non-vessel structures.

The quality of the images used for PLD may seriously degrade compared to the retinal images due to the motion of the sensor and all kinds of noises. Thus, MF-FDOG method may not be robust enough for PLD. Song & Li (2014), then proposeda problem-specific design threshold scheme to overly detect power line segments. In Song & Li (2014), the thresholding scheme is to set a reference threshold on the response to FDOG firstly, and then adjusted by the response to MF. The response images are obtained by convolving the input image with MF kernel and FDOG filter kernel in N orientations prior to maximizing and minimizing all the N filtered images at each pixel. They are denoted by M and G respectively. The reference threshold is denoted by T_G and set as follows(Song & Li, 2014):

$$T_G = c \cdot \mu_G \tag{2.4}$$

Where c is a constant and μ_G is the mean value of the FDOG's response, G. Next the reference threshold is adjusted by the following equation(Song & Li, 2014):

$$T = \left(1 + \bar{M}_n\right) \cdot T_G \tag{2.5}$$

Where $\bar{M}_n$ is calculated by normalizing $\bar{M}$, which is the local mean of the response image M as shown in the following equation(Song & Li, 2014):

$$\bar{M} = M * R \tag{2.6}$$

where R is a mean filter with size of $r \times r$ whose individual elements are $1/r^2$. This mean filter is used to decrease the bad effects of the noise or outliers and gets more robust detection results.

By thresholding T on the response image G, the final edge map image D is created as (Song & Li, 2014):

$$D = \begin{cases} 0 & \text{if } G(x,y) \le T(x,y); \\ 1 & \text{otherwise.} \end{cases} \tag{2.7}$$

where (x,y) is the index of the pixel. It can be seen from the Figure2.4 and equations (2.5)–(2.7) that if there is a symmetrical edge in the image, the magnitude in M_n will be high at the corresponding area and the threshold T will become larger by equation (2.5), so the edge can be easily detected by equation (2.7); if there is a non-symmetrical edge, the corresponding magnitude in M_n will be small and the threshold T will become smaller, so it can be suppressed adaptively(Song & Li, 2014).

2.2.4 Multiscale vessel enhancement filter: Frangi filter

The multiscale second order local structure of an image (Hessian) was examined by Frangi *et al.* (1998),with the purpose of developing a vessel enhancement filter. A vesselness measure was obtained on the basis of all eigenvalues of the Hessian. This clinical utility of this method was shown by the simultaneous noise and background suppression and vessel enhancement in maximum intensity projections and volumetric displays, whentested on two dimensionaldigital subtraction angiography (DSA) and three dimensional aortoiliac and cerebral magnetic resonance angiogram (MRA) data.This approach conceives vessel enhancement as a filtering process that searches for geometrical structures which can be regarded as tubular. Since vessels appear in different sizes, a measurement scale is introduced which varies within a certain range(Frangi *et al.*, 1998).

2.2.4.1 Second order local structure of an image

The local behaviour of an image, L, is analysed by considering its Taylor series expansion in the neighbourhood of a point X_o (Frangi *et al.*, 1998),

$$L(\mathrm{X}_o + \delta\mathrm{X}_o, s) \approx L(\mathrm{X}_o, s) + \delta\mathrm{X}_o{}^{\mathrm{T}}\nabla_{o,s} + \delta\mathrm{X}_o{}^{\mathrm{T}}\mathrm{H}_{o,s}\delta\mathrm{X}_o \tag{2.8}$$

This expansion approximates the structure of the image up to second order. $\nabla_{o,s}$ and $\mathrm{H}_{o,s}$ are the gradient vector and Hessian matrix of the image computed in X_o at scale s. The concepts of linear scale space theory (Florack *et al.*, 1992; Koenderink, 1984) is used to calculate these differential operators of L in a well-posed fashion. The differentiation is defined as a convolution with derivatives of Gaussians(Frangi *et al.*, 1998):

$$\frac{\partial}{\partial \mathrm{X}} L(\mathrm{X}, s) = s^{\gamma} L(\mathrm{X}) * \frac{\partial}{\partial x} G(\mathrm{X}, s) \tag{2.9}$$

where the D-dimensional Gaussian is defined as(Frangi *et al.*, 1998):

$$\mathrm{G}(\mathrm{X}, s) = \frac{1}{\sqrt{2\pi s^2}^{D}} e^{-\frac{\|\mathrm{X}\|^2}{2s^2}} \tag{2.10}$$

The parameter γ was introduced by Lindeberg (1996) to define a family of normalized derivatives. This normalization is particularly important for a fair comparison of the response of differential operators at multiple scales. When no scale is preferred, γ should be set to unity.

Analysing the second order information (Hessian), the third term in equation (2.8), has an intuitive justification in the context of vessel detection(Frangi *et al.*, 1998). The second order derivative of a Gaussian kernel at scale s generates aprobe kernel that measures the contrast between the region inside and outside the range $(-s, s)$ in the direction of the derivative while the second order ellipsoid describes the local principal directions of curvature as shown in Figure 2.4.

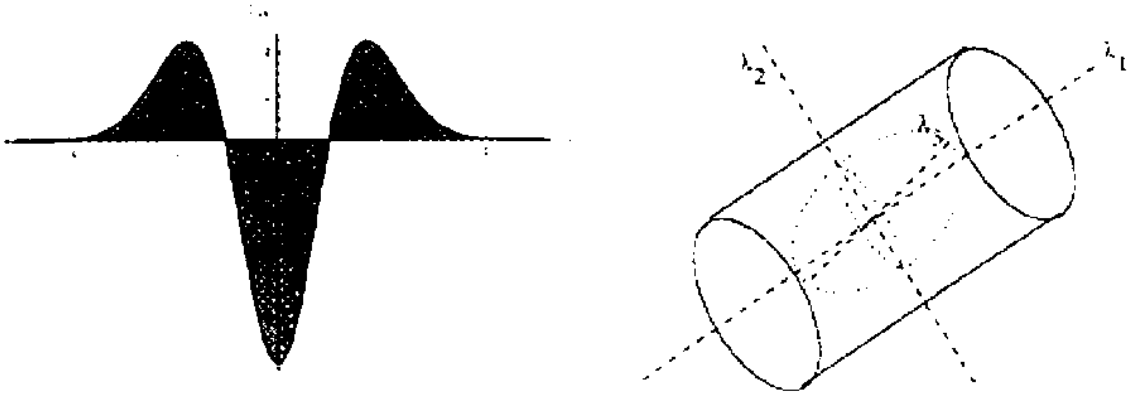


Figure 2.4: The Second Order Information of a Gaussian kernel (Frangi *et al.*, 1998).

2.2.4.2 Eigen value analysis of the Hessian matrix

The idea behind eigenvalue analysis of the Hessian is to extractthe principal directions in which the local second order structure of the image can be decomposed. Since this directly gives the direction of smallest curvature (along the vessel), hence theapplication of several filters in multiple orientations is avoided. This latter approach is computationally more expensive and requires a discretization of the orientation space(Frangi *et al.*, 1998). Let $\lambda_{s,k}$ denote the eigenvalue corresponding to the k-*th* normalized eigenvector $\hat{u}_{s,k}$ of the Hessian $\mathrm{H}_{o,s}$, all computed at scale s. From the definition of eigenvalues (Frangi *et al.*, 1998):

$$\delta x_o{}^{\mathrm{T}} \mathrm{H}_{o,s} \delta x_o = \left(\frac{\partial}{\partial \delta x_o} \right) \left(\frac{\partial}{\partial \delta x_o} \right) L(x_o, s) \tag{2.11}$$

$$\mathrm{H}_{o,s} \hat{u}_{s,k} = \lambda_{s,k} \hat{u}_{s,k} \tag{2.12}$$

and it follows that

$$\hat{u}_{s,k}^{\mathrm{T}} \mathrm{H}_{o,s} \hat{u}_{s,k} = \lambda_{s,k} \tag{2.13}$$

By analysing equations (2.11)-(2.13) a nice geometric interpretation arises. The eigenvalue decomposition extracts three orthonormal directions which are invariant up to a scaling factor

when mapped by the Hessian matrix. In particular, a spherical neighbourhoodcentredat X_o with radius 1, N_{X_o}, will be mapped by X_o onto an ellipsoid whose axes are along the directions given by the eigenvectors of the Hessian and the corresponding axis' semi-lengthsare the magnitudes of the respective eigenvalues(Frangi *et al.*, 1998).

This ellipsoid locally describes the second order structure of the image (as shown in Figure 2.5b) and can be used as an intuitive tool for the design of geometric similarity measures.

In this research, λ_k will be the eigenvalue with the k-*th* smallest magnitude $\left(|\lambda_1| \le |\lambda_2| \le |\lambda_3|\right)$. Under this assumption Table 2.1 summarizes the relations that must hold between the eigenvalues of the Hessian for the detection of different structureswhere H=high, L=low, N=noisy, usually small, +/- indicate the sign of the eigenvalue (Frangi *et al.*, 1998).

Table 2.1: Possible Patterns of the Eigenvalues λ_k in 2D and 3D(Frangi *et al.*, 1998).

2D		**3D**			**Orientation pattern**
λ_1	λ_2	λ_1	λ_2	λ_3	
N	N	N	N	N	Noisy, no preferred direction
		L	L	H-	Plate-like structure (bright)
		L	L	H+	Plate-like structure (dark)
L	H-	L	H-	H-	Tabular structure (bright)
L	H+	L	H+	H+	Tabular structure (dark)
H-	H-	H-	H-	H-	Blob-like structure (bright)
H+	H+	H+	H+	H+	Blob-like structure (dark)

In particular, a pixel belonging to a vessel region will be signalled by λ_1 being small (ideally zero), and λ_2 and λ_3 of a large magnitude and equal sign (the sign is an indicator of brightness/darkness). The respective eigenvectors point out singular directions: $\hat{u}_1$ indicates the direction along the vessel (minimum intensity variation) and $\hat{u}_2$ and $\hat{u}_3$ form a base for the orthogonal plane. This prior information related to the imaging modality can be used as a consistency check to discard structures present in the dataset with a polarity different than the onesought (Frangi *et al.*, 1998)

The eigenvaluesfor an ideal tubular structure in a 3D image is summarized as (Frangi *et al.*, 1998):

$$|\lambda_1| \approx 0 \tag{2.14}$$

$$|\lambda_1| \square |\lambda_2| \tag{2.15}$$

$$\lambda_2 \approx \lambda_3 \tag{2.16}$$

and the sign of λ_2 and λ_3 indicate its polarity.

It is emphasized that all three eigenvalues play an important role in the discrimination of the local orientation pattern. The dissimilarity measure takes into account two geometric ratios based on the second order ellipsoid. The first ratio, $\Re_B$, accounts for the deviation from a blob-like structure but cannot distinguish between a line and a plate-like pattern(Frangi *et al.*, 1998):

$$\Re_B = \frac{\text{Volume}/(4\pi/3)}{(\text{Largest Cross Section Area}/\pi)^{3/2}} = \frac{|\lambda_1|}{\sqrt{|\lambda_2\lambda_3|}} \tag{2.17}$$

This ratio attains its maximum for a blob-like structure and is zero whenever $\lambda_1 \approx 0$, or λ_1 and λ_2 tend to vanish. Also, λ_1/λ_2 remains bounded even when the second eigenvalue is very small since its magnitude is always larger than the first (Frangi *et al.*, 1998).

The second ratio, $\Re_A$, refers to the largest area cross section of the ellipsoid (in the plane orthogonal to $\hat{u}_1$) and accounts for the aspect ratio of the two largest second order derivatives. This ratio is essential for distinguishing between plate-like and line-like structures since only in the latter case it will be zero(Frangi *et al.*, 1998),

$$\Re_A = \frac{\left(\text{Largest Cross Section Area}\right)/\pi}{\left(\text{Largest Axis Semi-Length}\right)^2} = \frac{|\lambda_2|}{|\lambda_3|} \tag{2.18}$$

The twogeometric ratios introduced so far are grey-level invariant (i.e., they remain constant under intensity re-scalings). This ensures that the measures only capture the geometric information of the image. However, vessel structures are brighter than the background and occupy a (relatively) small volume of the whole dataset. If this information is not incorporated, background pixels would produce an unpredictable filter response due to random noise fluctuations. However, a distinguishing property of background pixels is that the magnitude of the derivatives (and thus the eigenvalues) is small, at least for typical signal-to-noise ratios present in acquired datasets. To quantify this, the Frobenius matrix norm of the Hessian is used, since it has a simple expression in terms of the eigenvalues when the matrix is real and symmetric. Hence the following measureof second order structureness is defined as (Frangi *et al.*, 1998),

$$s = \left\|\mathrm{H}\right\|_F = \sqrt{\sum_{j \le D} \lambda_j^2} \tag{2.19}$$

where D is tile dimension of the image.

This measure will be low in the background where no structure is present and the eigenvalues are small for the lack of contrast. In regions with high contrast compared to the background, the norm will become larger since at least one of the eigenvalues will be large(Frangi *et al.*, 1998).

2.2.4.3 Vesselness function

The vesselness function is defined as the following combination of the components(Frangi *et al.*, 1998),

$$v_o(\mathrm{s}) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \text{ or } \lambda_3 > 0 \text{ ,} \\ \left(1-\exp\left(-\frac{\Re_{\mathrm{A}}{}^2}{2\alpha^2}\right)\right)\exp\left(-\frac{\Re_{\mathrm{B}}{}^2}{2\beta^2}\right)\left(1-\exp\left(-\frac{s^2}{2c^2}\right)\right) & \text{otherwise} \end{cases} \qquad (2.20)$$

where α , β and c are thresholds which control the sensitivity of the line filter to the measures $\Re_{\mathrm{A}}$, $\Re_{\mathrm{B}}$ and s. The idea behind this expression is to map the features in Equations (2.17)-(2.19) into probability-like estimates of vesselness according to different criteria. The different criteria is combined using their product to ensure that the response of the filter is maximal only if all three criteria are fulfilled(Frangi *et al.*, 1998).

In all the results presented in Frangi *et al.* (1998) showed α and β were fixed to 0.5. However, the value of the threshold c depends on the grey-scale range of the image and half the value of the maximum Hessian norm has proven to work in most cases.

The vesselness measure in equation (2.20) is analysed at different scales, s. The response of the line filter will be maximum at a scale that approximately matches the size of the vessel to detect. The vesselness measure provided by the filter response at different scales is integrated to obtain a final estimate of vesselness(Frangi *et al.*, 1998):

$$v_o(\gamma) = \max_{s_{\min} \le s \le s_{\max}} v_o(s, \gamma) \qquad (2.21)$$

where $s_{\min}$ and $s_{\max}$ are the maximum and minimum scales at which relevant structures are expected to be found. They can be chosen so that they will cover the range of vessel widths.

For 2D images , the following vesselness function which follows from the same reasoning as in 3D is given as (Frangi *et al.*, 1998):

$$v_o(\mathrm{s}) = \begin{cases} 0 & \text{if } \lambda_2 > 0, \\ \exp\left(-\dfrac{\mathfrak{R}_{\mathrm{B}}{}^2}{2\beta^2}\right)\left(1-\exp\left(-\dfrac{s^2}{2c^2}\right)\right) & \text{otherwise} \end{cases} \tag{2.22}$$

Here, $\mathfrak{R}_{\mathrm{B}} = \lambda_1/\lambda_2$ is the blobness measure in 2D and accounts for the eccentricity of the second order ellipse.

2.2.5 Graph-cut model based line detection

There are three questions which must be answered when grouping line segments in the pool into whole power lines:

1. Which elements in the pool are the useful compositions?
2. Which ones are the background noises?
3. Which elements are each whole power line composed of?

The classical answers are HT and morphological methods. But these methods make an assumption on line model that power lines are straight, and ignore curve situation in which power lines usually hang down for force of gravity(Song & Li, 2014). Mathematically, straight and curve lines can be described by first-order and high-order polynomial respectively. As a matter of fact, it is enough to represent power line by quadratic polynomial, which curves down and shapes like parabola. Hence, a quadratic polynomial is used to fit every power line instead of straight line assumption. This enables detection of not

only the straight power lines but also the curve ones. The problem is formulated as a graph cut model by Song & Li (2014).

2.2.5.1 Definition of graph cut terminologies

Let $G=(V,E)$ be a connected undirected graph where $v_i \in V$ is a vertex corresponding to an element of the line segment pool V. $e_{ij} \in E$ is the edge between the line segments v_i and v_j, and each edge has a weight $w(v_i, v_j)$ which is defined to quantitatively measure the probability of collinearity between the two line segments connected by the edge. The greater the probability, the bigger the weight. The line segment pool is wished to be partitioned into mutually exclusive groups, such that each group is a connected $\tilde{G}=(\tilde{V},\tilde{E})$ representing a complete line, where $\tilde{V} \subseteq V$, $\tilde{E} \subseteq E$. Taking binary partition as an example, nonempty sets V_1 and V_2 form a partition of the graph G if $V_1 \cap V_2 = \varnothing$ and $V_1 \cup V_2 = V$.

A cut is related to a set of edges which are removed from the graph G and break the graph into two disjoint sets V_1 and V_2. So the grouping problem can be solved by minimizing the cut cost, which is defined as (Song & Li, 2014):

$$\min cut(V_1,V_2) = \min\left(\sum_{v_1 \in V_1, v_2 \in V_2} w(v_1, v_2) \right) \tag{2.23}$$

Because the number of line segments belonging to each group may be varied a lot, this can cause bias toward small group. The objective function in equation(2.23) is transferred to the normalised cut - Ncut-(Shi & Malik, 2000) to alleviate the unnatural bias is as follows (Song & Li, 2014):

$$\min Ncut(V_1,V_2) = \min\left(\frac{cut(V_1,V_2)}{vol(V_1)} + \frac{cut(V_1,V_2)}{vol(V_2)}\right) \tag{2.24}$$

where $vol(V_1) = \sum_{v_1 \in V_1, v \in V} w(v_1, v)$ Hence the minimal cut bias is circumvented because the small group will not have small Ncut cost in equation (2.24). The Ncut optimization problem is well studied in spectral graph theory and can be modelled as ageneralized eigenvalue problem.According to Rayleigh–Ritz theorem (Lütkepohl, 1996), the second smallest eigenvector of the graph Laplacian matrix, $L = D - W$ is the real valued solution to the Ncut problem as shown by the following equation(Song & Li, 2014) :

$$\min_{Y \perp Y_1} \frac{Y^{\mathrm{T}}(D-W)Y}{Y^{\mathrm{T}}DY} \quad \text{subject to } Y(i) \in \{1, -\beta\} \text{ and } Y^{\mathrm{T}}D1 = 0 \tag{2.25}$$

where W is the weight matrix with $W_{ij} = w(v_i, v_j)$, D is a diagonal matrix with element $d(i) = \sum_{j=1}^{n} W_{ij}$, and $\beta = \sum_{v_i \in V_1} d(i) \Big/ \sum_{v_j \in V_2} d(j)$. Finally, partitioning the graph can be performed by thresholding on the eigenvalue of Laplacian Matrix.

2.2.5.1 Design of weights of the edges

The weight $w(v_i, v_j)$ of the edge e_{ij} , which measures the probability of collinearity between any two line segments is defined as(Song & Li, 2014)

$$w(v_i, v_j) = a \cdot \exp\left(-b \cdot \frac{err_{ij}}{ecc_{ij}}\right) \tag{2.26}$$

where a and b are constants, ecc_{ij} is the eccentricity of the minimal ellipse which can capture the region of the line segments v_i and v_j , err_{ij} is the error of quadratic polynomial fitting them.

By the above design of the weight matrix and the optimization method in literature (Shi & Malik, 2000), it is found from the experimental experience that when a and b are all set to 1, the global optimal solution can be achieved by thresholding on the eigenvalue of Laplacian

matrix with 0. That is to say just the eigenvectors corresponding to the positive eigenvalues contribute to the solution. So the groups of line segments are got by partitioning the graph, and each group is on behalf of a line(Song & Li, 2014)

2.2.6 Morphological operations

Morphological operations are applied on the edge-map image to eliminate all objects except those which are long and thin. With the help of several morphological operators this goal can be approached(Gerke & Seibold, 2014).

2.2.6.1 Area opening

Applying an 'area opening' algorithm, small connected components with a pixel count less than are removed while elongated objects tend to be preserved(Gao *et al.*, 2003; Jähne *et al.*, 1999).

The size s_a of the objects to be deleted depends on the image size. Therefore an empiric factor f is defined as:

$$f = \max\left(1, \text{round}\left(0.0007 * \left(\text{image width} + \text{image height}\right)\right)\right) \tag{2.27}$$

The size s_a for the area opening is

$$s_a = 50 * f \tag{2.28}$$

2.2.6.2 Hit-miss operation

A 'Hit-miss' filter is the general case of the morphological neighbourhood operators(Russ, 2016). Three states (must be white, must be black or don't care) at specific positions must match to turn the pixel at the centre position to the state "hit" (=white).

In the case shown in Figure 2.6 the 'hit-miss' filter removes all objects which are smaller than or equal to 2x2 pixel.

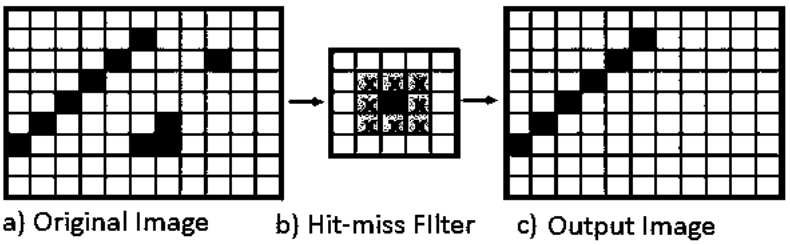


Figure 2.5: Hit-miss Operation(Gerke & Seibold, 2014)

Principally, the same filter as shown in Figure 2.6 but with a much larger "don't care" area will be used to remove non power line objects. The size s_h of the 'hit-miss' filter is:

$$s_h = 20 * f \tag{2.29}$$

2.2.6.3 Eccentricity

Compact objects are not considered to be part of a power line. As an indicator for the compactness of a pixel area, the shape of an ellipse which best fits an area of connected pixels will be used. The parameters of this ellipse are calculated by the second order moments (Jähne, 2005; Mukundan & Ramakrishnan, 1998). The order is defined by $p+q$(Gerke & Seibold, 2014):

$$M_{p,q} = \sum_x \sum_y x^p y^q I(\mathrm{x}, \mathrm{y}) \tag{2.30}$$

$$\mu_{p,q} = \sum_x \sum_y \left(x - \overline{x}\right)^p \left(y - \overline{y}\right)^q \tag{2.31}$$

$$\overline{x} = \frac{M_{10}}{M_{00}} \text{ and } \overline{y} = \frac{M_{01}}{M_{00}} \tag{2.32}$$

$$a = \sqrt{\frac{\mu_{20} + \mu_{02} + \sqrt{\left(\mu_{20} - \mu_{02}\right)^2 + 4\mu_{11}^2}}{\sqrt{\mu_{00}}}} \tag{2.33}$$

$$b = \sqrt{\frac{\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{\sqrt{\mu_{00}}}} \tag{2.34}$$

$$\theta = \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \tag{2.35}$$

$$\varepsilon = \frac{(\mu_{20} - \mu_{02})^2 - 4\mu_{11}^2}{(\mu_{20} + \mu_{02})^2} \tag{2.36}$$

where:

M = moment

p,q = parameter for order of moment

x,y = position of a pixel

I = pixel intensity

μ = central moment

$\bar{x},\bar{y}$ = center of gravity

a,b = long and short half axes

θ = angle of axes a

ε = eccentricity

In the case of power line detection, only the eccentricity ε is taken into account. The eccentricity is zero for a circle and one for a line. Therefore all objects with an eccentricity smaller than 0.94 are deleted(Gerke & Seibold, 2014).

2.2.6.4 Smoothness measure

A measurement was proposed in literature (Song & Li, 2014) to weigh the smoothness of the detected edges before filtering simply by thresholding the proposed smoothness measurement. Noticing that any one of the power lines can be fitted by a quadratic polynomial whose second order derivative is a constant, the smoothness measurement denoted by S for an edge is defined as the variance of the second order difference of each point's tangential direction(Song & Li, 2014):

$$S = \frac{1}{l-1}\left(\sum_{(x,y)\in\bar{e}} (D(x,y) - \mu_D)^2\right) \tag{2.37}$$

where (x,y) is the index of a point on the edge e, l is the length of the major axis of the ellipse which can capture the edge, $D(x, y)$ is the second order difference of the tangential direction at the point (x,y) and μ_D is the meanvalue of the second orderdifferences of all points belonged to $\bar{e}$. The range of the tangential direction is (-π, π]. From the above definition, it can be seen that the edge is smoother, the measurement value S is smaller. The edge map is filtered by thresholding on the smoothness measurement defined as (Song & Li, 2014):

$$\bar{e} \text{ is } \begin{cases} \text{kept} & \text{if } S \le t_u \\ \text{filtered out} & \text{otherwise} \end{cases} \tag{2.38}$$

The upper bound t_u can remove the non-smooth edge, which is determined by the smoothness measurements of the training line segments.

2.2.7 Performance metrics

In power line detection process, the outcome is a pixel-based classification result. Any pixel is classified either a power line or not. Consequently, there are four possibilities; two classifications and two misclassifications.

The classifications are the true positive (TP) where a pixel is identified as power line in both the ground truth and segmented image, and the true negative (TN) where a pixel is classified as a non-power-line in the ground truth and the segmented image. The two misclassifications are the false negative (FN) where a pixel is classified as non-power-line in the segmented image but as a power-line pixel in the ground truth image, and the false positive (FP) where a pixel is marked as power-line in the segmented image but non-power-line in the ground truth image.

2.2.7.1 True positive rate

The true positive rate (TPR) represents the fraction of pixels correctly detected as power-line pixels.

$$TPR = TP/power\text{-}line\ pixel\ count \tag{2.39}$$

2.2.7.2 False positive rate

The false positive rate (FPR) is the fraction of pixels erroneously detected as power-line pixels.

$$FPR = FP/non\text{-}power\text{-}line\ pixel\ count \tag{2.40}$$

2.2.7.3 Accuracy

The accuracy (Acc) is measured by the ratio of the total number of correctly classified pixels (sum of true positives and true negatives) to the number of pixels in the image field of view (FOV).

$$Acc = (TP+TN)/FOV\ pixel\ count \tag{2.41}$$

2.2.7.4 Sensitivity

Sensitivity (SN) reflects the ability of the algorithm to detect the power-line pixels.

$$SN = TP/(TP+FN) \tag{2.42}$$

2.2.7.5 Specificity

Specificity (SP) is the ability to detect non-power-line pixels. It can be expressed as $1 - FPR$

$$SP = TN/(TN+FP) \tag{2.43}$$

2.2.7.6 The positive predictive value

The positive predictive value (PPV) or precession rate gives the proportion of identified power-line pixels which are true power-line pixels. The PPV is the probability that an identified power-line pixel is a true positive.

2.2.7.7 Receiver operating characteristic curve

A receiver operating characteristic (ROC) curve plots the fraction of power-line pixels correctly classified as power-line, namely the TPR, versus the fraction of non-power-line

pixels wrongly classified as power-line, namely the FPR). The closer the curve approaches the top left corner; the better is the performance of the system. The most frequently used performance measure extracted from the ROC curve is the value of the area under the curve (AUC) which is 1 for an optimal system.

For power line images, the TPR and FPR are computed considering only pixels inside the FOV andit will be performance metrics used by power-line detection algorithms in this research.

2.3 Review of Similar Works

Detection of power lines is a highly researched topic as two distinct communities deal with it: the first for aircraft navigation safety and the second for power line inspection(Mirallès *et al.*, 2014). Over the years, scientists have proposed a wide range vision-based PLD algorithms (Chen *et al.*, 2016; Luo *et al.*, 2014) and a large number of patents can also be found in the field of power line monitoring (Matikainen *et al.*, 2016).

Liu *et al.* (2012)developed a fast power line detection and localisation algorithm. Firstly, ridge pointswere obtained based on the energy functions of the steerable oriented filter, and then linear features extracted by grouping the collinear ridge points. In addition, knowledge of the characteristics were used to distinguish power lines from surrounding linear objects. Finally, a line fitting algorithm, based on the idea of region growing and connected component analysis, was adopted to refine candidate power lines in images. Overall, the developed algorithm reduced FPR and was computationally efficient. However, it requires images to be taken at altitude ranging from 20 to 60 meters above power lines which makes it not applicable for obstacle avoidance in low altitude flight

Yang *et al.* (2012)proposed a real-time image processing algorithms to detect power lines from UAV video images. First,HT was used to detect line candidates in the binary images,

thenFuzzy C-means (FCM) clustering algorithm was used to discriminate the power lines from these detected line candidates, where the properties of power lines were used to remove the spurious lines and the length and slope of lines were used as the feature vector to create the clustering data set. The algorithm proposed was effective for automatic PLD. However the algorithm cannot be applicable in obstacle avoidance in low altitude flight because power line information must be obtained apriori to control the UAV flight.

Zhang *et al.* (2012)presented a novel method of power lines detection and tracking. Firstly, the image was transformed from the red-green-blue (RGB) space to the hue-intensity-saturation (HIS) colour space prior to extracting a gradient image from the I-channel and using Ostu method to extract pixels on power lines. Next, an improved K-means algorithm is used in the Hough space to cluster and filter the straight lines, and then detect power lines. Finally, a Kalman filter was used to track the power lines in the Hough space, based on the continuity of a video sequence. Although power lines were detected, however the algorithm was inefficient for real time processing as the data capacity of the video was huge hence increased computational complexity.

Bhujade *et al.* (2013)developed an approach for PLDfrom colour video frames obtained by a front looking camera. Firstly, The RGB image was converted to hue-saturation-value(HSV) image prior to sky and greenery suppression of pixel values in the image having saturation greater than 39% irrespective of hue and value. Next, the HSV was converted to a binary image, before the morphological erosion operator was applied to suppress small and large patches in the threshold binary image. Finally, HT was used to connect power line pixels and to detect the power line. However there werediscontinuities in the detected lines as parts of thepower lines were suppressed as background.

Cao *et al.* (2013)proposed a simple but effectiveimage prior-symmetry partial derivative distribution to detectpower lines in aerial image for UAVs. Firstly, radon transform was implemented on partial derivative image. Next, the power lines were found in the radon matrix by finding corresponding lines to thepeak points as the candidates prior to finding the anglecorresponding to these power line candidates by k-mean method. Next, the lines not parallel with power lines were removed and the lines close to each other were merged. Finally, the detected results were marked in the original image. Only straight lines were detected by the method as they assumed a downward view of the power lines with approximately straight lines cross in the image and parallel to each other.

Seibold *et al.* (2013)suggested a method to extract the lines even in front of cluttered background like trees by means of range filters, some morphological operations, Canny operator and HT. The procedure was tested on many images and extracted most of the lines with only a few non lines. The orientation of the power lines or the structure of the background had little effect on the result, howeverbackgrounds like long edges on buildings led to some false detection and real time detection of the power lines was not achieved.

Zhu *et al.* (2013)developed an automatic power line recognition method for UAV inspection vision system. Firstly, a novel double-side filter method was used to enhance the power lines in images, then radon transform was employed to detect the straight lines. Finally, the power lines was recognized by parallel lines constraint to avoid the misidentification caused by road or farmland. The proposed double-side filter could reduce the background noise effectively and did a good job in enhancing the power lines in images. However the method only detected the power lines in aerial images having some characteristics: The images are shot from a downward view and power lines are approximately straight lines across the images and are parallel to each other.

Cerón & Prieto (2014) came up with a new method for PLD (Circle Based Search, CBS) which searches for lines between two opposite points using computer graphic primitives, based on geometric relationship that are inherent in the circle symmetry. Canny and Steerable filters were used for edge detection, prior to the detection of line segments by CBS. The performance in non-vertical or horizontal lines depends on the radius and the length of the line which needs to be long enough. This made the method only suitable in scenes with longer lines and power line detection in images of smaller size.

Gerke & Seibold (2014)proposed a method for vision system to extract power lines from cluttered natural background. Firstly, the grey scale image was filtered in 4 directions by a range filter and then converted to binary image prior to the application of morphological operations on them before then superimposed. Next, edges were detected using Canny edge detector and finally grouped into whole power lines by HT. The algorithm proved reliable in terms of low false positive, false negative detection rates and real time processing. However, a few false detection of non-existing power lines occurred.

Luo *et al.* (2014) used a novel definition of power line to present its object aware properties (the thin line structure property *Ps*, the special material property *Pm*, and the flat colour property *Pc*.) and then a cascaded PLD scheme was devised based on corresponding image cues in joint colour (RGB) and near-infrared (NIR) images. The algorithm began with the pairing of line segments, detected by a joint LSD method for the joint RGB-NIR images, according to the property descriptor Ps. Next, the property Pm was used to filter out non power line regions. Finally, potential power line objects were validated by using the Pc property. The object aware PLD method had superiority in higher detection rates, lower false alarm and robustness. However, it took more time than traditional line detection.

Sharma *et al.* (2014)recommended a three stage solution for locating and complete extraction of power lines based on a novel morphological operator and a robust image space heuristics. Firstly, power lines in varying light conditions were isolated by adaptive thresholding, followed by the selection of candidate set of edges using a new erosion operator and finally, the detection of "stripes" in image matrix corresponding to power lines by applying robust heuristics on pixel intensity. Overall, the algorithm minimised the false negatives and false positives but only has its application in overhead monitoring of power lines as the images needs to be taken at a height above the power lines and in relative proximity.

Song & Li (2014)proposed a sequential local to global PLD algorithm, in which line segments with symmetrical edges were detected by morphologically filtering an edge map image obtained based on MF and FDOG. This resulted in over detection which guaranteed low missing detection rate. A novel Graph-cut model based on graph theory was then developed to group the line segment pool into whole line pools prior to picking up the 'true' power lines by morphological properties again. The algorithm, although detected both straight and curved power lines, it had better performance in images with simple background with high contrast as compared to images with dense power lines, seriously cluttered background or poor visibility condition. This resulted in a FPR that is far from ideal.

Shan *et al.* (2015) put forward a novel spatial context based PLD method in which the pylon-line spatial context was used to link lines from a line segment pool (extracted utilizing the LSD algorithm) with power line pylons (detected with a SIFT based approach), thus forming power line segment seeds which found the nearby power line segments utilizing an incremental search algorithm with the assistance of line-line spatial context. Although, the speed of the algorithm wasn't taken into consideration, the method obtained good results with

little false alarms. However same can't be said for images with heavily cluttered background, such as urban settings, with the presence of edges of buildings and without power pylons.

Sharma *et al.* (2015)presented a five-stage algorithmic framework that can be used for automatic, real-time detection of different linear infrastructural objects in outdoor aerial images.The first stage involved Mean shift filtering to find the peak of a confidence map using the colour histogram of the image, resulting in background suppression then gradient magnitudes for all edges of the segmented image was estimated as first feature, using Sobel function**.** In the next stage, linear features were detected by tracking the prominent boundary of such objects in the gradient image using a novel method for boundary growing. Detection of contour of infrastructural linear features in Image Space was done using contour growing approach, prior to the final stage being **r**emoval of false positives using rigidity feature, as represented by total sum of gradient orientations. The algorithm is however unsuitable for obstacle avoidance in low altitude flight asprior knowledge of the relative position of the power-line is required for efficient detection of seed points.

Alpatov *et al.* (2016)suggestedan approach to computational effective power line detection by the Integrated Vector Radon Transform (IVRT) based approach using gradient direction information. The method was effective in power line detection as the IVRT based algorithm uses image subdivision on 64x64 pixel squares thus avoiding the wire bend problem and the unification of separate segments was performed using a multi-agent approach however this increased the computational complexity of the algorithmmaking it inefficient for real time processing.

Zhou *et al.* (2016)made use of power line specific knowledge to build a model which achieves predictive and continuous parameter selection so that the best thresholds are selected for changing scenarios. Due to the benefit of the power line model and the improved

mapping/prediction scheme, the algorithm outperformed all of the traditional power line detection methods, especially in complex backgrounds, however their model assumed that the power lines are approximately straight lines and parallel to each other as they only considered images captured by the top view camera which makes it not applicable for obstacle avoidance in low altitude flight were power lines appear as curved lines as captured from front view camera.

From the review of similar works and literature, it is evident that fixed threshold are only effective when the background is relatively monotonic thus they may cause these algorithms to fail with complex and changing backgrounds (Zhou *et al.*, 2016). MF-FDOG is also not without its own drawbacks as it returned a high false positive rate and a detection rate insufficient for real time processing.FFremedies this problem since it directly gives the direction of smallest curvature (along the vessel) hence the application of several filters in multiple orientations,which is computationally more expensive and requires a discretization of the orientation space, is avoided(Frangi *et al.*, 1998).

CHAPTER THREE

MATERIALS AND METHODS

3.1 Introduction

In this chapter, the methods, materials and procedures employed for the successful completion of this research are discussed which involves the replication of the Matched Filter and First Order Derivative of Gaussian (MF-FDOG) and the development of the Frangi Filter and First Order Derivative of Gaussian (FF-FDOG) Power Line Detection (PLD) algorithms.The steps of the methodology adopted for this research are as highlighted in section 1.5.

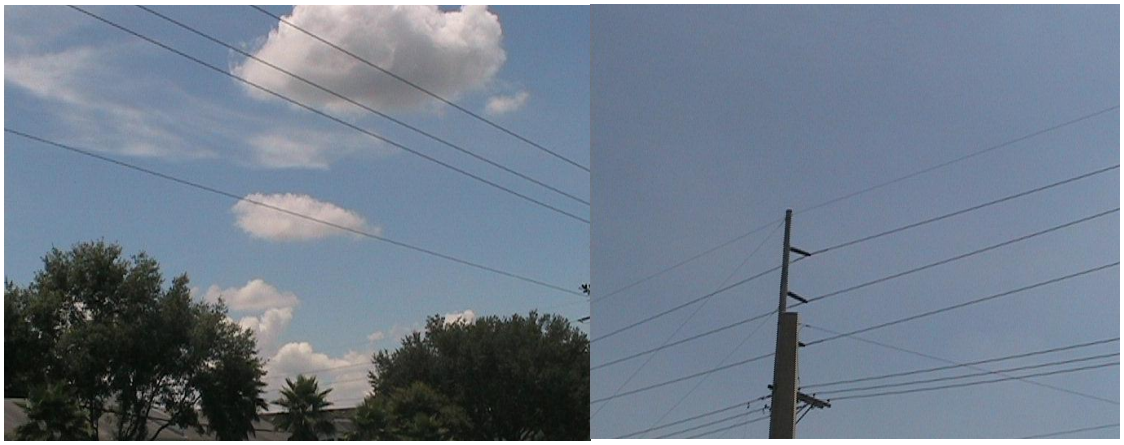
3.2 Experimental data.

The processes involved in obtaining the power line images and experimental parameters used for this research are discussed in details in the following sub-sections

3.2.1 Image Acquisition.

The images used to evaluate the performance of the proposed algorithm are classified into two sets. The dataset with power lines includes 50 picturesdownloaded fromUniversity of South Florida (USF), computer vision and pattern recognition wire database(Candamo *et al.*, 2010) and 10 images which were taken by a Nicon D60 digital camera in Samaru Zaria, in January, 2017. The resolution of all these images is720×480 pixels. The datasets are enriched in the following four aspects as shown by the sample images in Plates 3.1 and 3.2:

1. Different noise level
2. Different ambiguity by varying the distances of range perspective and the motion of the camera
3. Different weather conditions including cloudy, sunshine, fog.
4. Different backgrounds, such as clouds, mountains, buildings, trees and poles.



a) Image 1 b) Image 2



c) Image 3 d) Image 4



e) Image 5 f) Image 6

Plate 3.1: Sample Images of Power Lines downloaded from USF WireDatabase



a) Image 1



b) Image 2



c) Image 3



d) Image 4



e) Image 5



f) Image 6

Plate 3.2: Captured Images of Power Lines in Samaru, Zaria.

3.2.2 Clutter measure

The images used in the dataset contain urban characteristics (buildings, cables, trees, etc.), which make the detection process much more complex, due to the many linear patterns that are formed by the clutter and objects in the background. A clutter measurement (Candamo *et al.*, 2009; Schmieder & Weathersby, 1983)on the visual complexity of the background is employed to quantitatively show the richness of the datasets. For a given image I, the clutter measurement is defined as the averaging variances of the intensity distribution at the sub-images level(Candamo *et al.*, 2009):

$$\text{clutter(I)} = \sqrt{\frac{1}{K}\sum_{i=1}^{K}\sigma_i^2} \qquad (3.1)$$

where σ_i^2 is the variance of the intensity in the i-th sub-image, K is the number of the sub-images. In this research, the sub-window size was set to 25% of the original image with reference to literature (Candamo *et al.*, 2009), which means that the image I is divided into 4×4 sub-window and K=16.Algorithm 3.1 illustrates the program code for clutter measure.

Algorithm 3.1: Clutter Measure MATLAB Code

```
function clutterI = clutterMeasure(image)
% read and convert image to grayscale
I = rgb2gray(imread(image));
% obtain the variance for each sub-image window
iVar = [];
for i=1:120:480
    for j=1:160:640
        w = I(i:i+119, j:j+159);
        x = reshape(w, numel(w), 1);
        iVar = [iVar var(double(x))];
        size(x);
    end
end
% calculate the image clutter
k =16;
clutterI = sqrt(sum(iVar)/k);
```

3.2.3 Parameters

The value of the various parameters used in this research as adopted from literature (Song & Li, 2014) are defined in this section.

3.2.3.1 The parameters in obtaining the edge map

The parameters used in the design of the matched filters (MF) and first order derivative of Gaussian filter (FDOG) are as follows(Song & Li, 2014)

1. s=1, the scale of MF and FDOG filters. The power line is thinner, the scale should be smaller. On the contrary, the power line is thicker, the scale should be larger.
2. L=10, the length of the kernel for MF and FDOG filters.
3. N=8, the number of orientations of MF and FDOG filters kernels.
4. r=10, the scale of the mean filter.
5. c= 3, the most important parameter in determining the reference threshold. The larger the c, the higher the reference threshold, so the edge can be easily detected which may cause high false positives. Inversely, it is smaller, the reference threshold is lower, so the edge will be suppressedand which may cause high false negatives(Song & Li, 2014).

3.2.3.2 The parameters in filtering the edge map image

The non-smooth edge samples are filtered out by thresholding on the smoothness measurement S in equation (2.37), so a proper choice of the upper bound t_u is very important. Under the inspiration of the definition of S and the experiments carried out in literature (Song & Li, 2014) that the major factor of controlling S is the curvature of the edge and secondarily length, t_u is determined by the smoothness measurements of the training line segments.Based on the analysis carried out in literature(Song & Li, 2014) , t_u was set to 45.

3.3 Replication of the MF-FDOG based PLD algorithm

The processes involved in the replication of the MF-FDOG based PLD algorithm (as highlighted in the flowchart in Figure 2.2) are discussed in details in the following sub-sections:

3.3.1 Pre-processing the Image

'imread' is the command used in MATLAB to read images from graphic file. The file part is given as the input to the command and it returns pixel intensity values in RGB scale. Next, rgb2gray' MATLAB command used to convert coloured images to gray scale images. This MATLAB command eliminates the hue and saturation information in the image while retaining its luminance. It takes the 3 dimensional RGB image as input and outputs a 2 dimensional grayscale image with intensity values ranging for 0 to 265.

3.3.2 Matched filtering

A matched filter kernel (Figure 3.1) is implemented in MATLAB using equation (2.1) and the parameters in section 3.2.3.

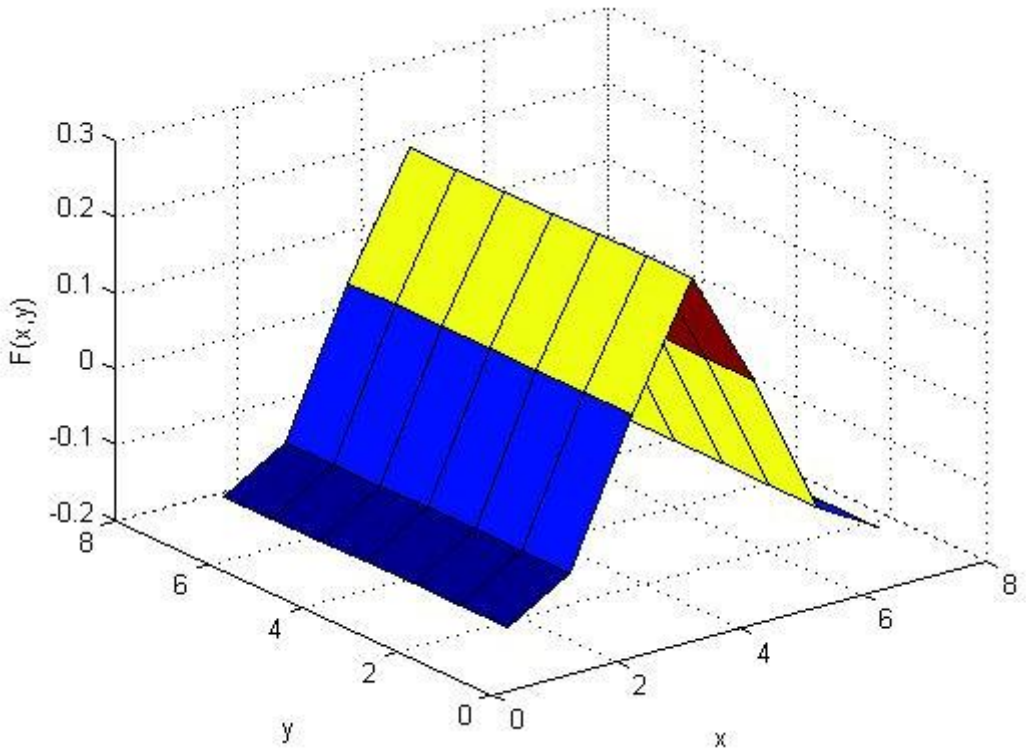


Figure 3.1: Matched Filter Kernel

The MF response image is then obtained from the convolution MF kernel and the grayscale image obtained in section 3.3.1. Algorithm 3.2 illustrates the program code for obtaining the MF response image.

Algorithm 3.2: MF Filtering MATLAB Code

```
% % Matched Filter
function [gx,gy]=matchedfilter(IM,sigma)

epsilon=1e-2;
halfsize =  ceil(sigma*sqrt(-2*log(sqrt(2*pi)*sigma*epsilon)));
size= 2*halfsize+1;
t=3;
p = normcdf([-3*sigma 3*sigma]);
s = (diff(p)/(2*t*sigma));
%generate a 2-D Gaussian kernel along x direction
for i=1:size
    for j=1:size
        u=[i-halfsize-1 j-halfsize-1];
        hx(i,j)=gauss(u(1),sigma)- s;
    end
end
hx=hx/sqrt(sum(sum(abs(hx).*abs(hx))));
%generate a 2-D Gaussian kernel along y direction
hy=hx';
figure; surf(hy)
%2-D filtering
gx=imfilter(IM,hx,'replicate','conv');
gy=imfilter(IM,hy,'replicate','conv');
```

3.3.3 First Order derivative of Gaussian filtering

A First Order Derivative of Gaussian filter kernel (Figure 3.2) is implemented in MATLAB using equation (2.3) and the parameters in section 3.2.3.

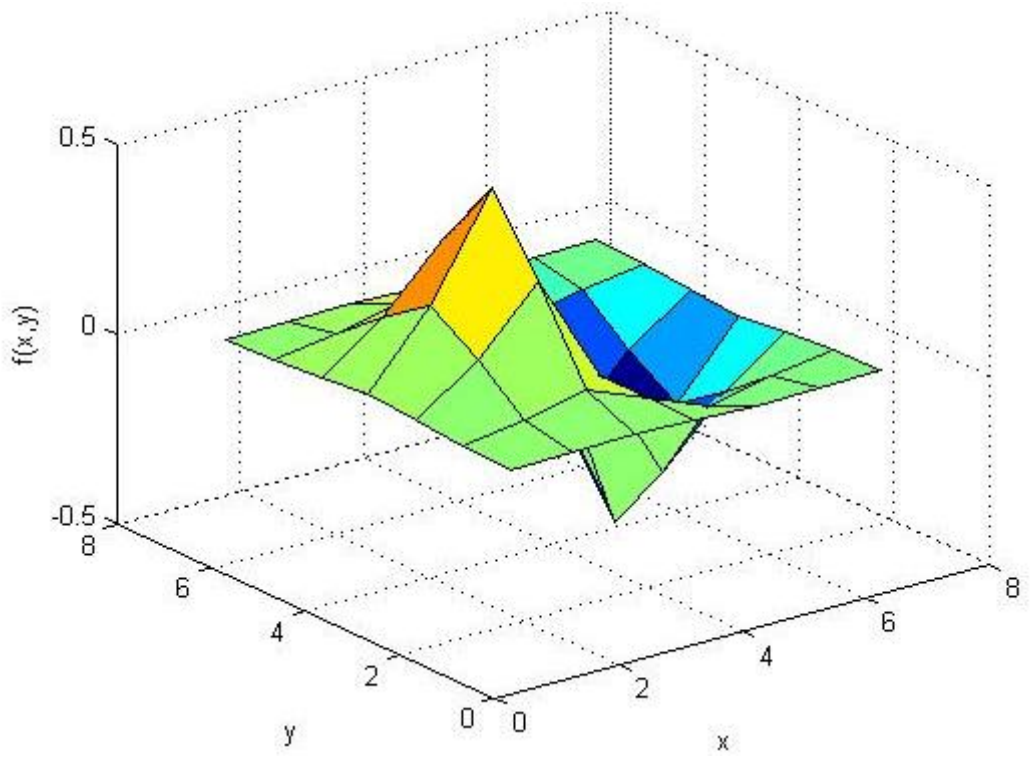


Figure 3.2: FDOG Filter Kernel

The FDOG response image is then obtained from the convolution FDOG kernel and the grayscale image obtained in section 3.3.1. Algorithm 3.3 illustrates the program code for obtaining the FDOG response image.

Algorithm 3.3: FDOG Filtering MATLAB Code

```
% % First Order Derivative of Gaussian Filter
function [gx,gy]=gaussgradient(IM,sigma)

epsilon=1e-2;
halfsize = ceil(sigma*sqrt(-2*log(sqrt(2*pi)*sigma*epsilon)));
size= 2*halfsize+1;
%generate a 2-D Gaussian kernel along x direction
for i=1:size
    for j=1:size
        u=[i-halfsize-1 j-halfsize-1];
        hx(i,j)=gauss(u(1),sigma)*dgauss(u(2),sigma);
    end
end
hx=hx/sqrt(sum(sum(abs(hx).*abs(hx))));
%generate a 2-D Gaussian kernel along y direction
hy=hx';

%2-D filtering
gx=imfilter(IM,hx,'replicate','conv');
gy=imfilter(IM,hy,'replicate','conv');
```

3.3.4 MF-FDOG thresholding

The MF-FDOG thresholding scheme as described in section 2.2.3.2 is implemented in MATLAB using equations (2.4) – (2.7). The inputs to the threshold scheme are MF response image, FDOG response image and the parameters discussed in section 3.2.3.1. A binary edge map image consisting of symmetric edges is obtained as step edges are suppressed by the thresholding scheme. Algorithm 3.4 illustrates the program code for obtaining the binary edge map image.

Algorithm 3.4: MF-FDOG Thresholding Scheme MATLAB Code

```
% reference threshold based on the image response to FDOG
c = 3;
uG = mean(mean(G));
TG = c*uG;

% adjust threshold based on the response to matched filter
r= 10;
R=(1/r).*ones(r);
M_bar=imfilter(M,R);

IM = M_bar;
IM = IM - min(IM(:));
IM = IM / max(IM(:));
Mn_bar = IM;
T = (1 + Mn_bar).*TG;

%apply threshold on the response to FDOG response
Q=G>T;
```

3.3.5 Morphological filtering

The non-power line edges are filtered out by taking full advantage of these facts. First, the over-short edges whose connected components with a pixel count less than a threshold are filtered out. A small threshold may cause high computational complexity, inversely a large one may suppress the power line edges and cause high false negatives. A trade-off is made and the threshold is set to 30 as in literature (Candamo *et al.*, 2009). This is implemented on the binary edge map image in MATLAB using the 'bwareaopen' command.

Then, smoothness of the detected edges are measured (as discussed in section 2.2.6.4) before filtering based on the threshold obtained in section 3.2.3.2.A line segment pool is obtained from this process.Algorithm 3.5 illustrates the program code for obtaining the line segment pool.

Algorithm 3.5: Morphological filtering MATLAB Code

```
% remove edge pixels with less than 30 connectedness
Q=bwareaopen(Q, 30);

% filtering by thresholding the smoothness measurement.
CC = bwconncomp(Q);
for idx=1:CC.NumObjects

    ind = CC.PixelIdxList{idx};

    [x,y]=ind2sub(CC.ImageSize, ind);
    p = polyfit(x,y,2);
    x0 = min(x):max(x);
    y0 = polyval(p,x0, 'r');

    curvexy = [x0', y0'];
    mapxy = [x,y];
    [xy,distance,t] = distance2curve(curvexy,mapxy,'linear');

    S=var(distance)*100;

    if (S > 110)
        Q(CC.PixelIdxList{idx})=0;
    end
```

3.3.6 Line segment clustering

The graph theory approach is used to group line segments from line segment pool into whole power lines, as discussed in section 2.2.5. Algorithm 3.6 illustrates the MATLAB program code for implementing the graph theory approach.

Algorithm 3.6: Line Segment Clustering MATLAB Code

```
% represents weights of the edges of the grapgh as adjacency matrix
A=A+diag(ones(1, length(A)));
S=sparse(tril(A));
[I, J, s] = find(S);
line = struct([]);

% group lines using  Normalised-cut
for i=1:length(s)
    pair = sort([I(i) J(i)]);
    n = 0;
    for j=1:numel(line)
        currentPair = line(j).segment;
        if any(ismember(pair, currentPair))
            n=j;
        end
    end
    if n == 0
       line(numel(line)+1).segment = pair;
    else
       line(n).segment = unique([line(n).segment pair]);
    end
end
```

3.4 Development of the FF-FDOG based PLD algorithm

The processes involved in the development of the FF-FDOG based PLD algorithm(as shown in Figure 3.3), involves pre-processing the image (as implemented in section 3.3.1) prior to obtaining a FF response image (as implemented in the following sub-sections) and FDOG filter response image(as implemented in section 3.3.2) which are used in obtaining the binary edge map image (as implemented in the following sub-sections). A line segment pool is obtained by the application of morphological operations to the binary edge map image (as implemented in section 3.3.6). Finally whole power lines are detected by grouping line segment from the pool by graph theory approach (as implemented in section 3.3.7):

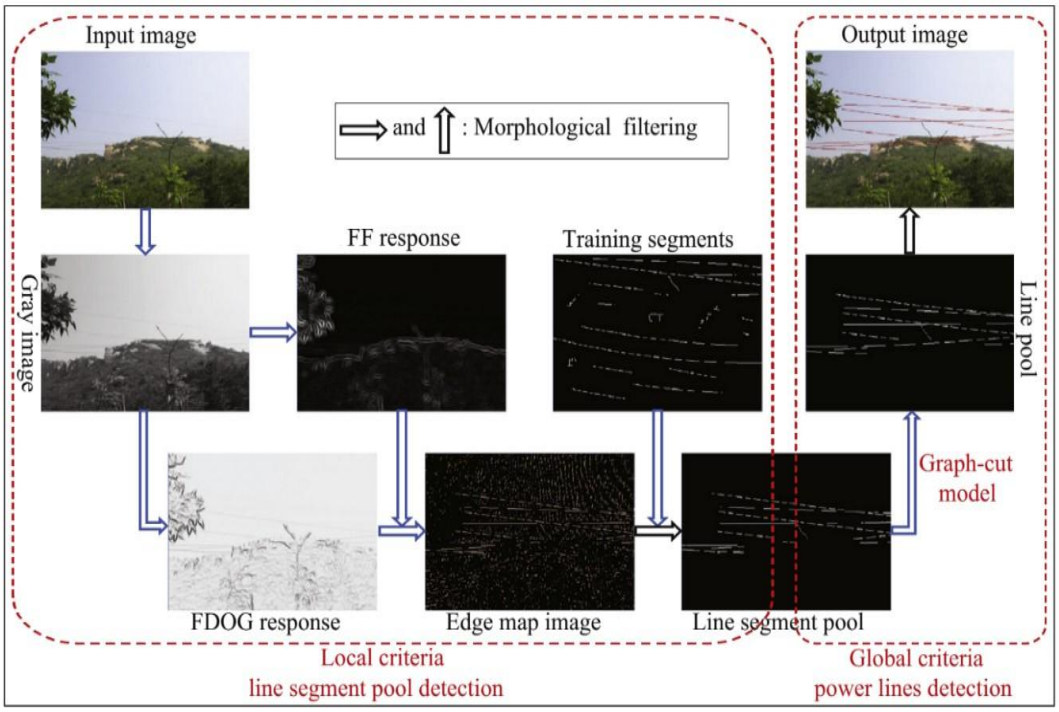


Figure 3.3: Overview of FF-FDOG Power Line Detection Algorithm

3.4.1 Frangi filtering

A FF response image is obtained based on the eigen value analysis of the second order local structure of the image (Hessian), as discussed in section 2.2.4.Algorithm 3.7 illustrates the program code for obtaining the FF response image.

Algorithm 3.7: Frangi Filtering MATLAB Code

```
% Frangi filter for all sigmas
for i = 1:length(sigmas),
    % Make 2D hessian
    [Dxx,Dxy,Dyy] = Hessian2D(I,sigmas(i));

    % Correct for scale
    Dxx = (sigmas(i)^2)*Dxx;
    Dxy = (sigmas(i)^2)*Dxy;
    Dyy = (sigmas(i)^2)*Dyy;

    % Calculate (abs sorted) eigenvalues and vectors
    [Lambda2,Lambda1,Ix,Iy]=eig2image(Dxx,Dxy,Dyy);

    % Compute the direction of the minor eigenvector
    angles = atan2(Ix,Iy);

    % Compute some similarity measures
    Lambda1(Lambda1==0) = eps;
    Rb = (Lambda2./Lambda1).^2;
    S2 = Lambda1.^2 + Lambda2.^2;
```

3.4.2 FF-FDOG thresholding

The FF-FDOG thresholding scheme is similar to the MF-FDOG thresholding scheme described in section 2.2.3.2. Firstly, a reference threshold T_G (defined in equation 2.4) is set based on the response to FDOG, then adjusted by the response to FF. They are denoted by *G* and *F* respectively. Next the reference threshold is adjusted by the following equation

$$T = (1 + F_n) \cdot T_G \tag{3.2}$$

Where F_n is calculated by normalising *F*. By thresholding *T* on the response image *G*, the final edge map image *D* is obtained as shown in equation (2.7). It can be seen from (2.7) that if there is a symmetrical edge in the image, the magnitude in F_n will be high at the corresponding area and the threshold T will become larger by equation (3.2), so the edge can be easily detected by equation (2.7); if there is a non-symmetrical edge, the corresponding magnitude in F_n will be weak and the threshold T will become smaller, so it can be suppressed adaptively.

Algorithm 3.8 illustrates the MATLAB program code for obtaining the binary edge map image from the FF-FDOG thresholding scheme.

Algorithm 3.8: FF-FDOG Thresholding Scheme MATLAB Code

```
% reference threshold base on the image response to FDOG
c = 3;
uG = mean(mean(G));
TG = c*uG;

% adjust threshold based on the response to frangi filter
IM = Ivessel;
IM = IM - min(IM(:));
IM = IM / max(IM(:));
Ivn = IM;
T = (1 + Ivn).*TG;

%apply threshold on the response to FDOG response
Q=G>T;
```

3.5 Performance measures

For each power line, the ground truth is obtained by first manually labeling several points, and then approximating with a single straight line or a quadratic polynomial. In order to evaluate the validity of the proposed algorithm under a reasonable performance metric, a straight power line is considered to be correctly detected if the angle and the y-intercept between it and the ground truth within 10degree and 20 pixels respectively (Cuijpers *et al.*, 2002). A curved power line is considered to be correctly detected if the average distance of the corresponding points is within 20 pixels. The performance is measured by true positives (TP) and false positives(FP) at line-level. The contingency table for power line detection is given in Table 3.1(Song & Li, 2014).

Table 3.1: The Contingency Table for PLD

Type	**Power line present**	**Power line absent**
Power line Detected	True positive	False positive
Power line not detected	False negative	True negative

Focusing on the two-by-two matrix in the bottom right corner of the Table 3.1, the elements on the major diagonal indicate the correct decisions made, and the other diagonal indicate the errors. Based on the confusion matrix, true positive rate (TPR) and false positive rate (FPR) are computed dividing the number of TP and FP by the number of truth power lines which are the sum of TP and false negatives (FN).Algorithm 3.9 illustrates the MATLAB program code for estimating the performance measure of the PLD algorithm.

Algorithm 3.9: PLD Algorithm Performance Measure MATLAB Code

```
% estimate TPR & FPR
TP = 0;
FP = 0;
dl = [];
CC = bwconncomp(W);
result = [];
for i=1:CC.NumObjects
    [x, y] = ind2sub(CC.ImageSize, CC.PixelIdxList{i});
    p = polyfit(x,y,1);
    result = [result; [atand(p(1)) p(2) (0-p(2))/p(1)]];
    d=find((line_angle < atand(p(1)) + 10) & (line_angle > atand(p(1)) - 10)...
        & (line_intercept2 < (0-p(2))/p(1) + 20) & (line_intercept2 > (0-p(2))/p(1) - 20));
    if isempty(d)
        FP = FP + 1;
    else
        if (~ismember(d, dl))
            dl = [dl d];
            TP = TP + 1;
        end
    end
end
TPR = (TP/NumOfLines) * 100;
FPR = (FP/CC.NumObjects) * 100;
```

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

In this section, the clutter measure and the performance of the matched filter and first order derivative of Gaussian (MF-FDOG) based power line detection (PLD)algorithm as well as that of the developed frangi filter and first order derivative of Gaussian (FF-FDOG) based PLD algorithm on the downloaded and captured power line imagesare discussed and relevant results reported.The two algorithms are implemented in MATLAB by a PC with a Core i7 CPU 2.00-GHz, 2.60-GHz with 4-GB memory.

4.2 Clutter Measure

From the clutter measure the power line image dataset (Appendix D2), the minimum and maximum clutter values are5.62 and 52.75 respectively which gives a range of the clutter to be 47.12, with a mean and standard deviation of 23.84 and11.36 respectively, the statistical histogram is as shown in Figure. 4.1.

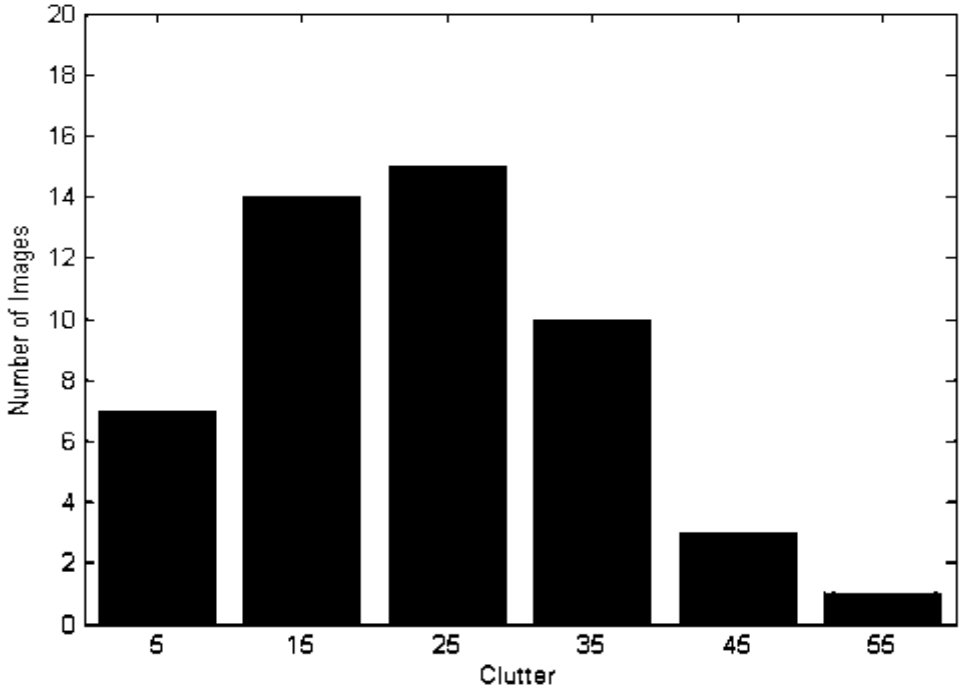


Figure 4.1: Clutter Statistics of Power Line Image Dataset

4.3 Experimental Results

To evaluate the efficiency of the developed FF-FDOG based PLD algorithm, the detection results are visualized and quantified by comparing with MF-FDOG based PLD algorithm in the following sub sections.

4.3.1 Visualized results

The visualized Image response of the various filters, thresholding schemesas well as the detection results of the MF-FDOG and FF-FDOG based PLD algorithms on the sample images are displayed in the following sub-sections.

4.3.1.1 Preprocessed images

The obtained power line images were converted to grayscale images. This process reduces dimension of the image from 3 dimension RGB image to a 2 dimension grayscacle image hence reducing the processing time for the algorithms. Plate 4.1 displays the corresponding grayscale images for the sample images of Plate 3.1.



a) Image 1



b) Image 2



c) Image 3



d) Image 4



e) Image 5



f) Image 6

Plate 4.1. Grayscale of Sample Images

4.3.1.2 Matched filter response images

The Matched filter response of the corresponding grayscale images in Plate 4.1 are displayed in Plate 4.2



a) Image 1 b) Image 2



c) Image 3 d) Image 4



e) Image 5 f) Image 6

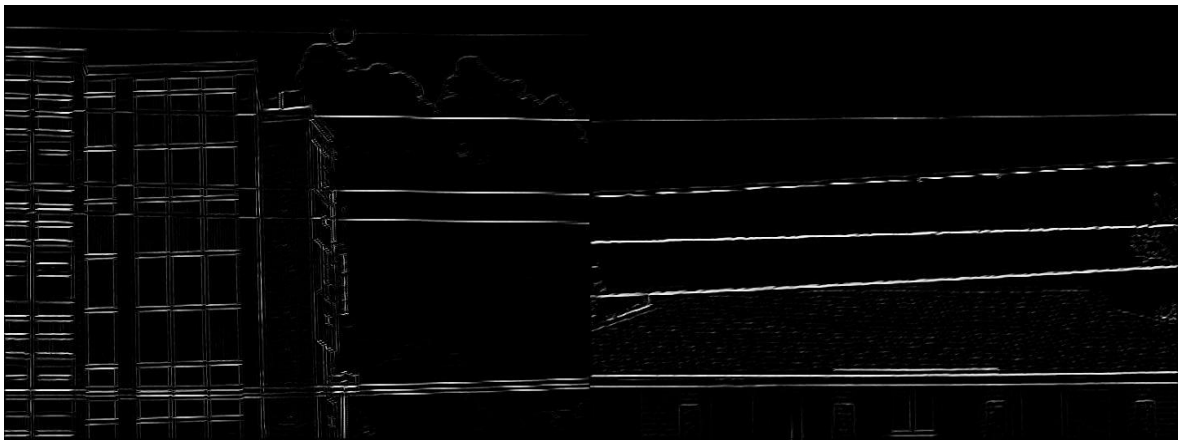
Plate 4.2: Matched Filter Response of Sample Images

4.3.1.3 Frangi filter response images

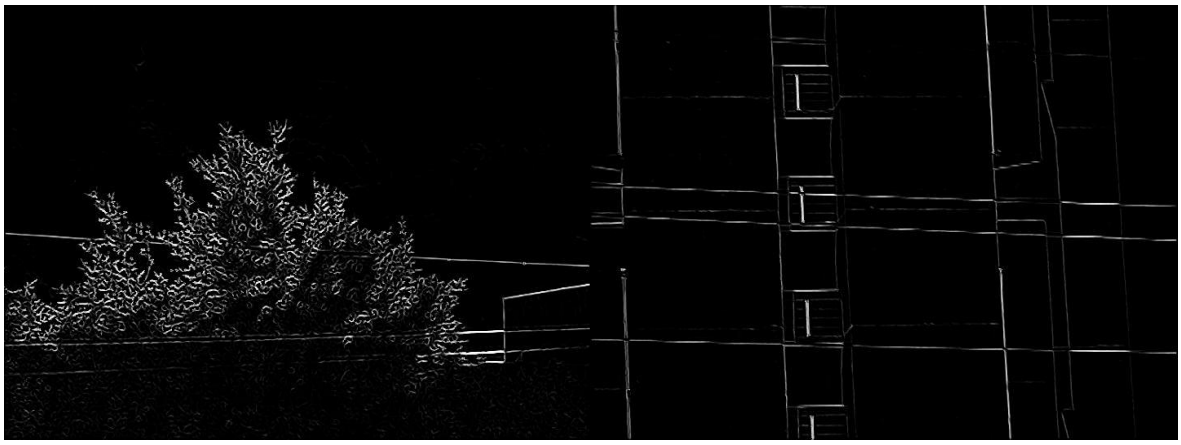
The Frangi filter response of the corresponding grayscale images in Plate 4.1 are displayed in Plate 4.3



a) Image 1 b) Image 2



c) Image 3 d) Image 4

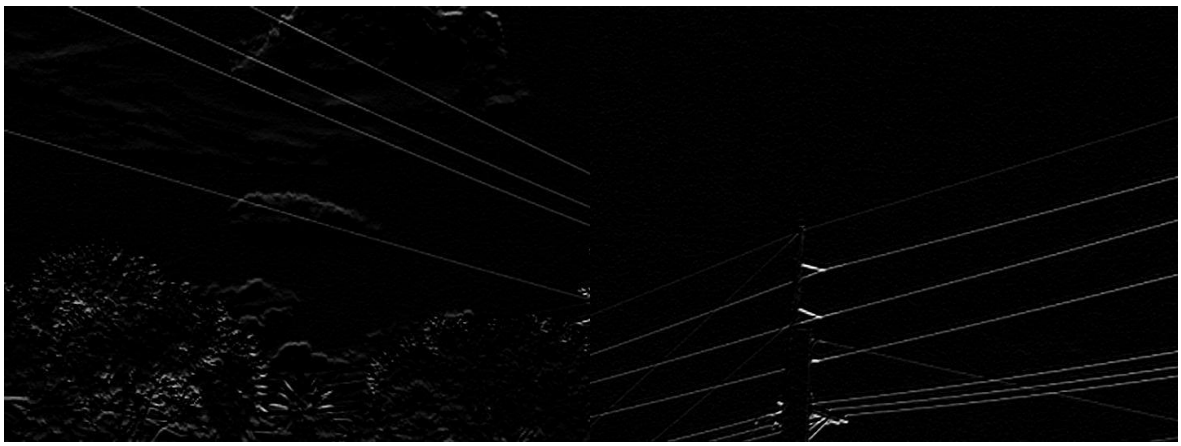


e) Image 5 f) Image 6

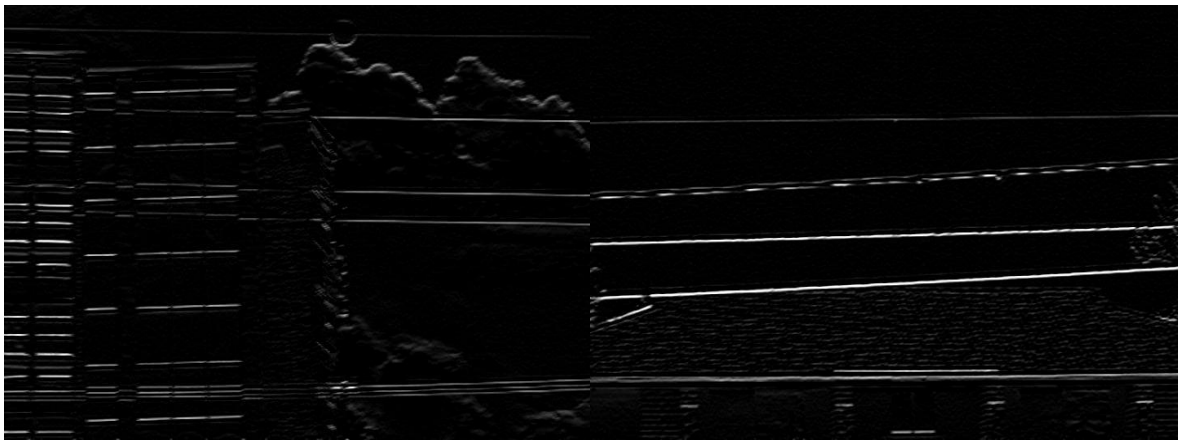
Plate 4.3: Frangi Filter Response of Sample Images

4.3.1.4 First order derivative of Gaussian filter response images

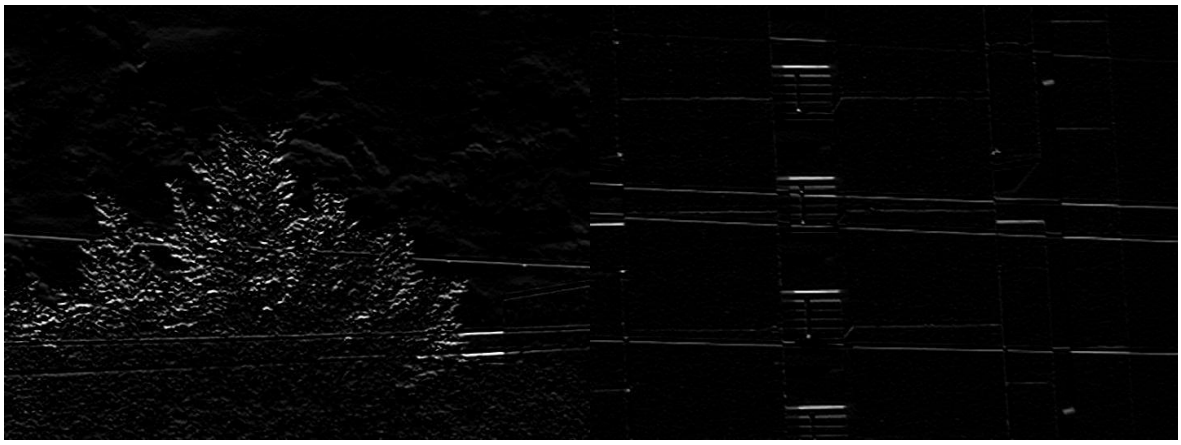
The First order derivative of Gaussian filter response of the corresponding grayscale images in Plate 4.1 are displayed in Plate 4.4



a) Image 1 b) Image 2



c) Image 3 d) Image 4

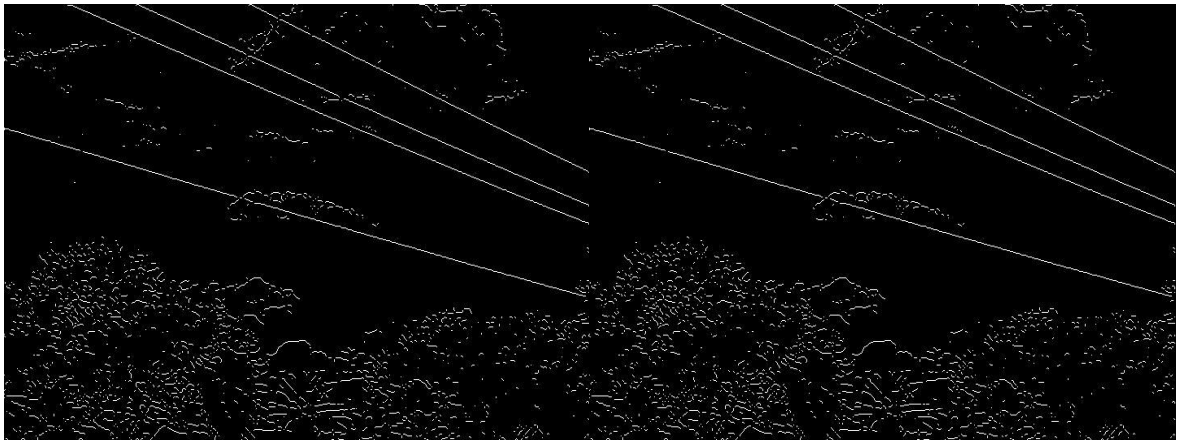


e) Image 5 f) Image 6

Plate 4.4: First Order Derivative of Gaussian Filter Response of Sample Images

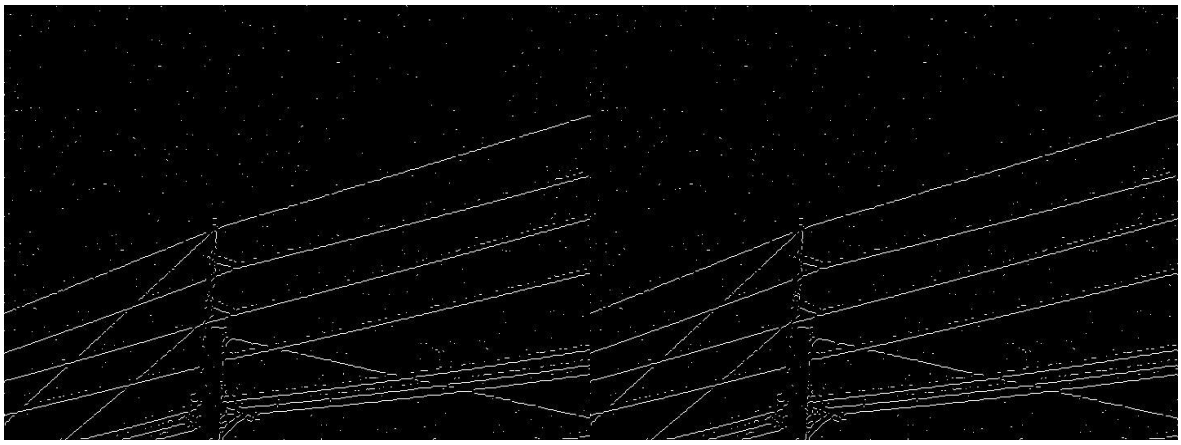
4.3.1.5 Edgemap images

The edgemap images obtained from the application of the MF-FDOG and FF-FDOG thresholding schemes on the various sample images are displayed in Plates 4.5 – 4.10.



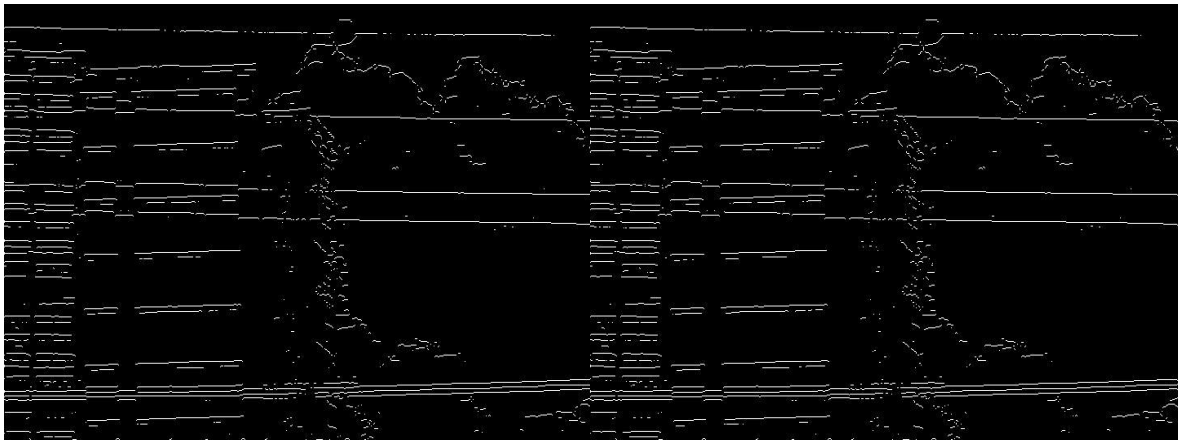
a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.5: Edgemap of Sample Image 1



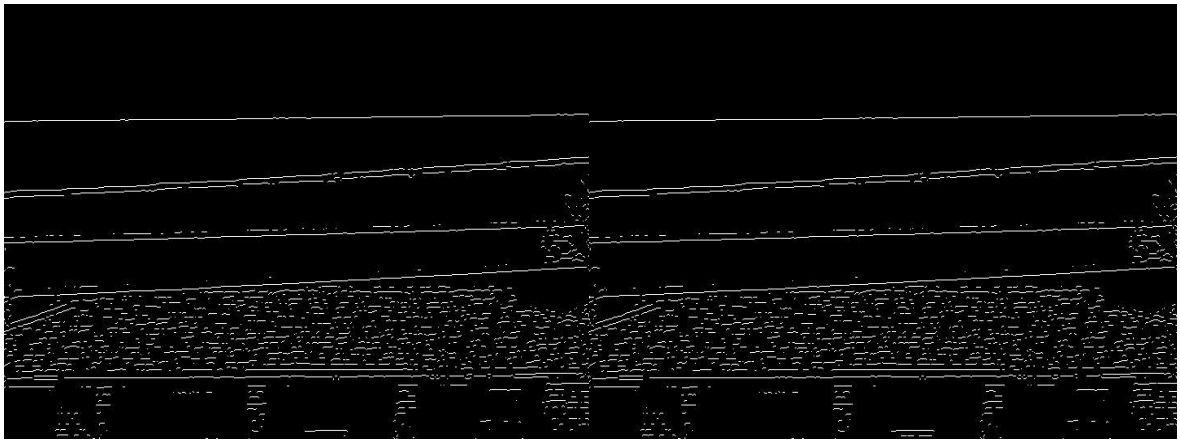
a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.6:Edgemap of Sample Image 2



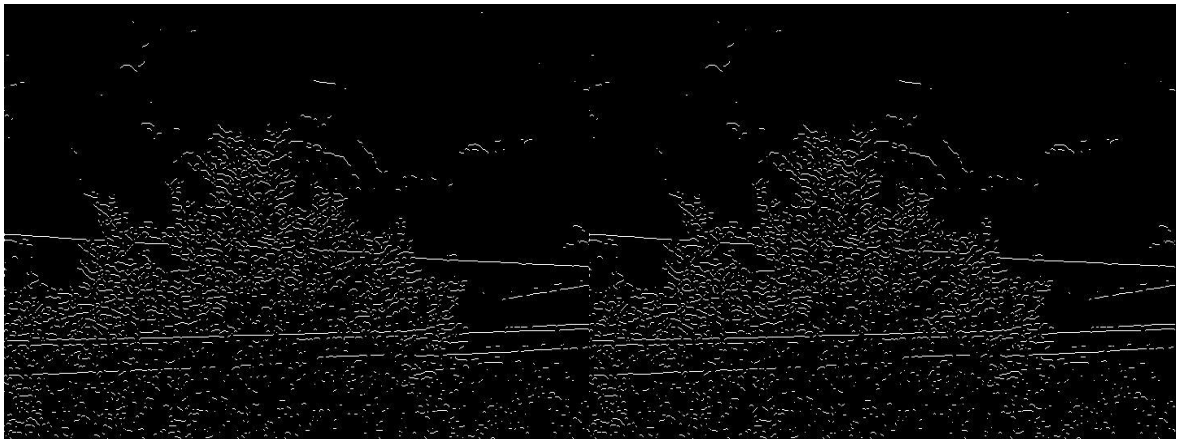
a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.7: Edgemap of Sample Image 3



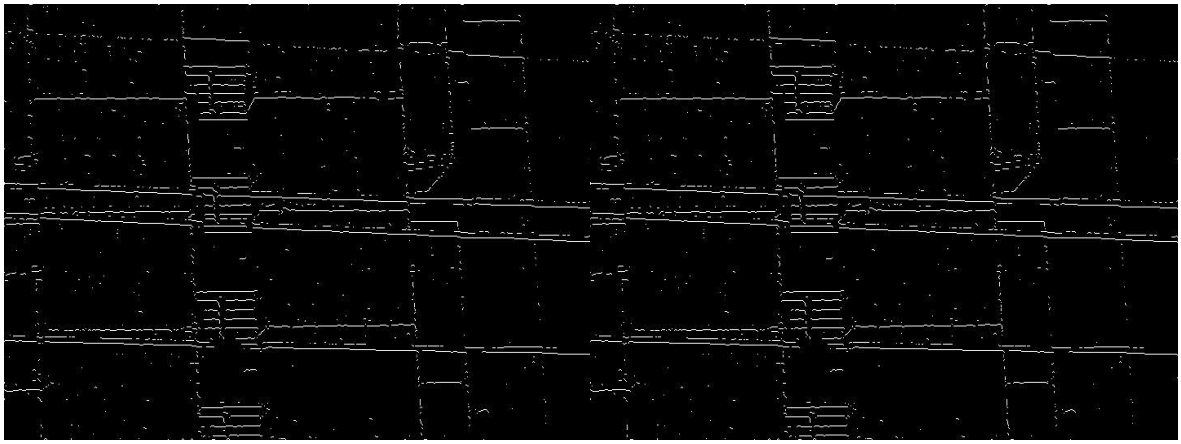
a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.8: Edgemap of Sample Image 4



a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.9: Edgemap of Sample Image 5

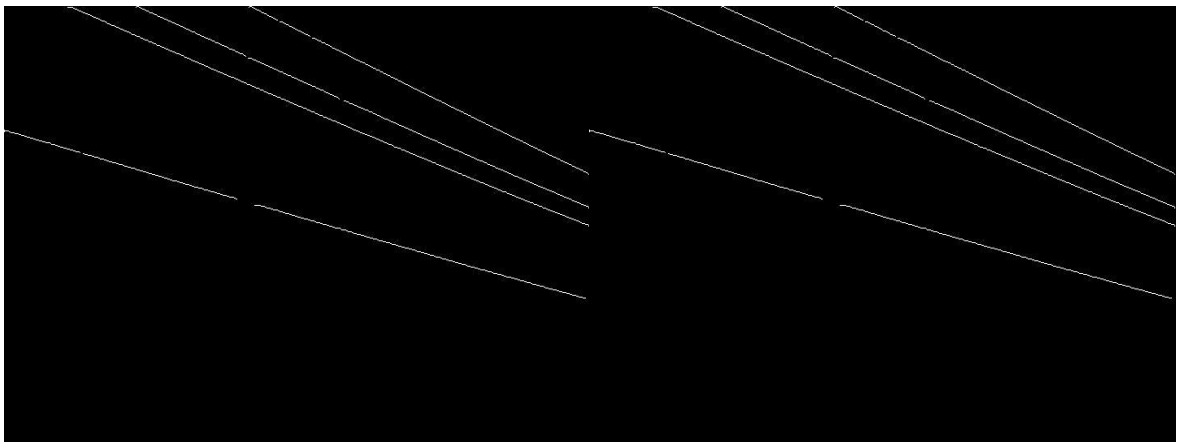


a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.10: Edgemap of Sample Image 6

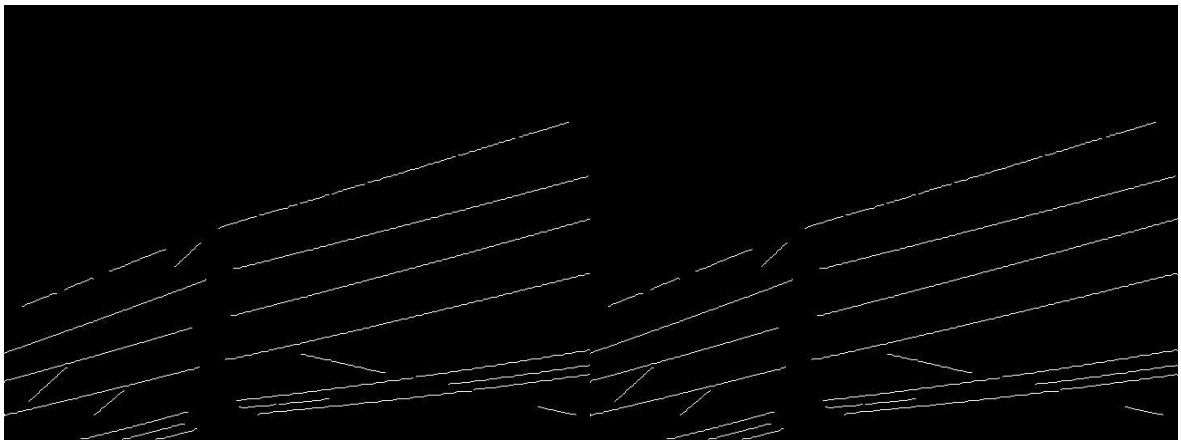
4.3.1.6 Line segment pool

The line segment pool obtained by morphologically filtering corresponding edgemap images in Plates 4.5 – 4.10 are displayed in Plates 4.11 – 4.16



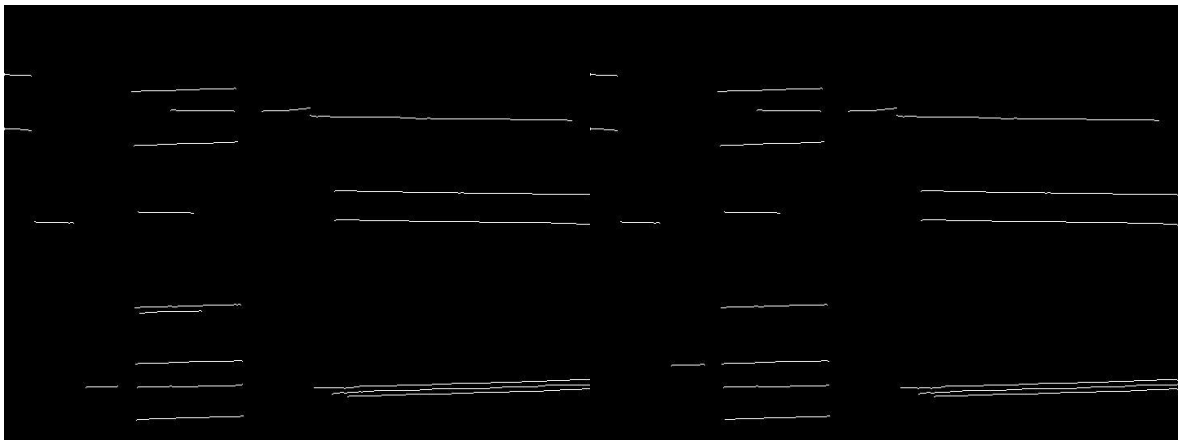
a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.11Line Segment Pool of Sample Image 1



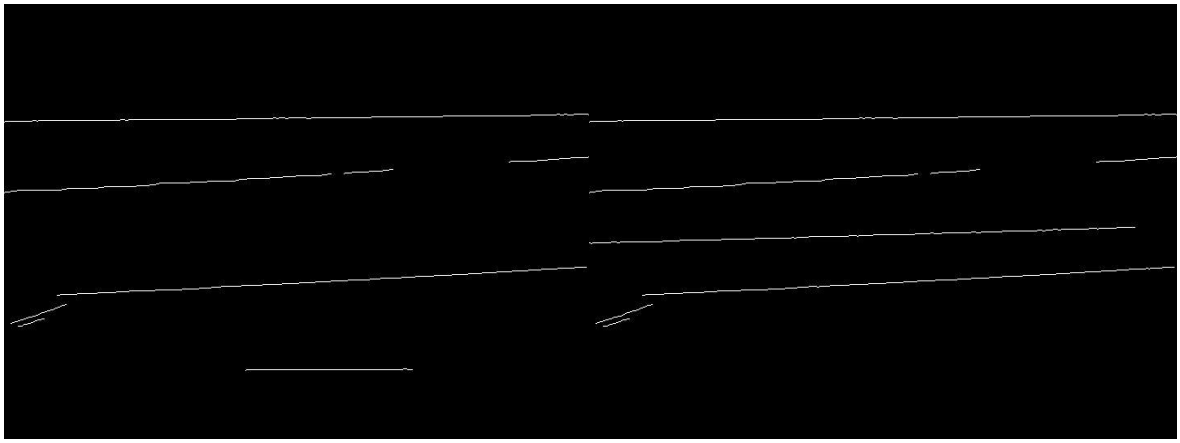
a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.12 Line Segment Pool of Sample Image 2



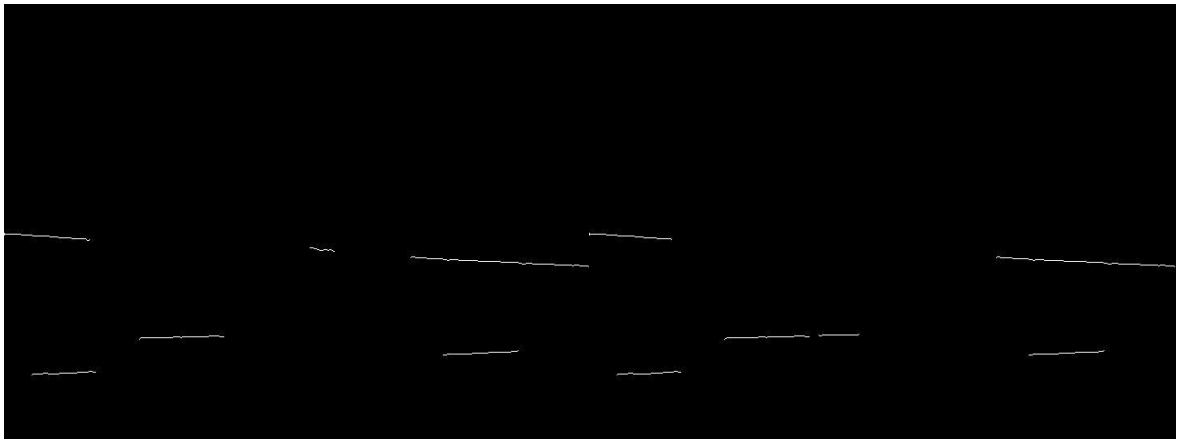
a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.13 Line Segment Pool of Sample Image 3



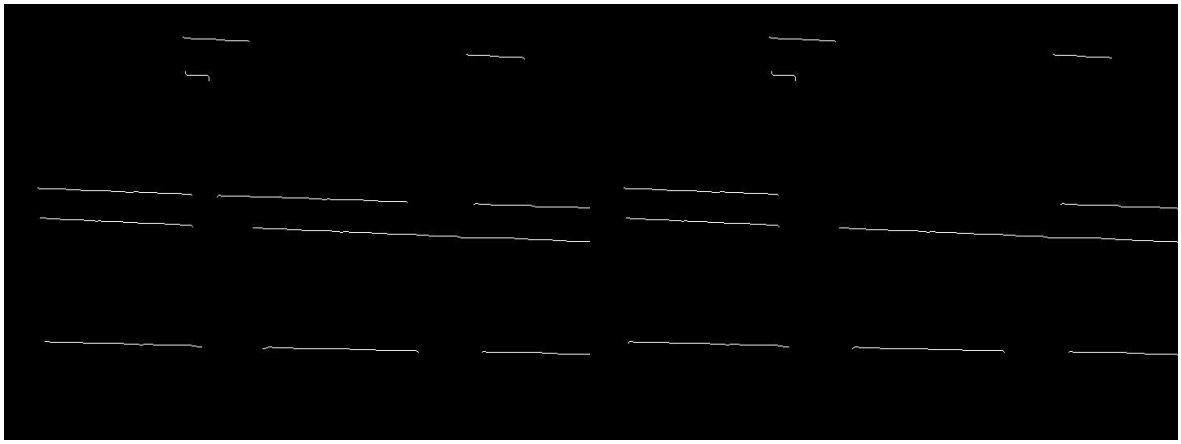
a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.14 Line Segment Pool of Sample Image 4



a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.15 Line Segment Pool of Sample Image 5



a) MF-FDOG Thresholding Scheme b) FF-FDOG Thresholding Scheme

Plate 4.16 Line Segment Pool of Sample Image 6

4.3.1.7 Power Line Detection results

The power lines detected by grouping line segments in the pool obtained by MF-FDOG and FF-FDOG based PLD algorithms (displayed in Plates 4.11 – 4.16)using graph theory approach are displayed in Plates 4.17 – 4.22.

From the detection results drawn by red lines, it can be concluded that FF-FDOG is as good as MF-FDOG based PLD algorithm when the detection problem is easy, (such as Plate 4.17)or with high contrastwith simple backgrounds (Plate 4.18).



a) MF-FDOG based PLD Algorithm b) FF-FDOG based PLD Algorithm

Plate 4.17: PLD Result of Sample Image 1



a) MF-FDOG based PLD Algorithm b) FF-FDOG based PLD Algorithm

Plate 4.18: PLD Result of Sample Image 2

The detection problem in the rest of the four images is tough for the dense power lines and seriously cluttered backgrounds (Plates4.19 - 4.21),



a) MF-FDOG based PLD Algorithm b) FF-FDOG based PLD Algorithm

Plate 4.19: PLD Result of Sample Image 3



a) MF-FDOG based PLD Algorithm b) FF-FDOG based PLD Algorithm

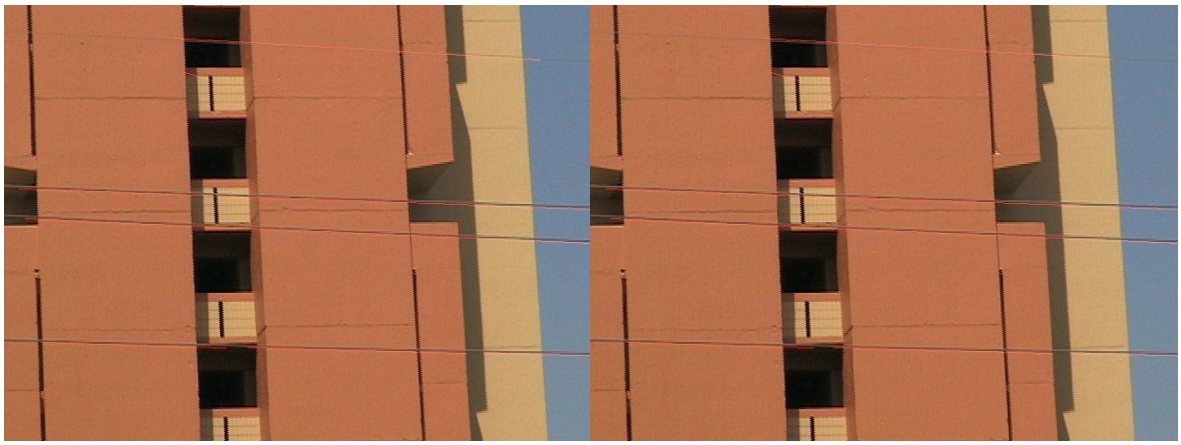
Plate 4.20: PLD Result of Sample Image 4



a) MF-FDOG PLD based Algorithm b) FF-FDOG based PLD Algorithm

Plate 4.21: PLD Result of Sample Image 5

or poor visibility conditionsespecially for background-like power lines as shownPlate 4.22.



a) MF-FDOG based PLD Algorithm b) FF-FDOG based PLD Algorithm

Plate 4.22: PLD Result of Sample Image 6

In these cases, the FF-FDOG method is better than MF-FDOG method.

4.3.2 Quantified results

The quantified comparisons focus on three aspects: true positive rate, false positive rate and time consumption.The quantified comparison between MF-FDOG method and the developed FF-FDOG method is given in Table 4.1.

Table 4.1: Power Line Detection Result of Dataset

Methods	**TPR (%)**	**FPR (%)**	**Computing time (s)**
MF-FDOG	84.16	17.91	0.62
FF-FDOG	86.39	11.45	0.61

It can be seen(from Table 4.1) that the time consumptions of these two algorithms are almost the same, but the developed FF-FDOGmethod is more efficient than the MF-FDOG method for higher true positive rate and lower false positive rate.

The true positive rate of the developed FF-FDOG PLD algorithm is 86.39%, which is 2.64% higher than MF-FDOG's 84.16%. The false positive rate of the developed FF-FDOG PLD algorithm is 11.45%, which is 36.06% lower than MF-FDOG's 17.91%. However, it still needs about 0.6s for each image which is insufficient for real-time processing. One way to reduce this insufficiency is using more powerful computer.

4.4 Discussion

Certainly, the developed FF-FDOG method has its weak points. From the results shown in Table 4.1, it can be found that the true positive rate and false positive rateare 86.39% and 11.45%respectively, which is far from perfect. In our analysis, there may be two causes:

1. Algorithm: In the local criterion of the developed algorithm, over detection is adopted to ensure low false negatives on account that the damage caused by it may be intolerable.
2. Data: Various line and vessel-like structures in images are strongly confusing, which makes the detection much more challenging. Especially, the images coming from urban settings, where the ripples on the roofs as well as edges of buildings are easily mistaken for power lines by the algorithm.

4.5 Application of FF-FDOG on captured images

The developed FF-FDOG based PLD algorithms is applied on the captured images of power linesfrom Samaru, Sabon-Gari Local Government Area of Kaduna State.The detection results are displayed in Plates 4.23 – 4.32.

From the detection results drawn by red lines, it can be concluded that FF-FDOG performed great when the detection problem is easy,with high contrast andsimple backgrounds(such as Plates 4.23 - 4.25).



a) Original Image b) Detection Result

Plate 4.23: Captured Image 1



a) Original Image b) Detection Result

Plate 4.24: Captured Image 2



a) Original Image b) Detection Result

Plate 4.25: Captured Image 3

The detection result was also great in images with the dense power lines (Plates 4.26 - 4.28),



a) Original Image b) Detection Result

Plate 4.26: Captured Image 4



a) Original Image b) Detection Result

Plate 4.27: Captured Image 5



a) Original Image b) Detection Result

Plate 4.28: Captured Image 6

Or seriously cluttered background with curved power lines as displayed in Plates 4.29 – 4.32



a) Original Image b) Detection Result

Plate 4.29: Captured Image 7



a) Original Image b) Detection Result

Plate 4.30: Captured Image 8



a) Original Image b) Detection Result

Plate 4.31: Captured Image 9



a) Original Image b) Detection Result

Plate 4.32: Captured Image 10

CHAPTER FIVE

SUMMARY,CONCLUSION AND RECOMMENDATIONS

5.1 Summary

This research is aimed at the development of a Frangi filter and first order derivative of Gaussian (FF-FDOG) based power line detection (PLD) algorithm to detect power lines from noisy and cluttered images. Results obtained when applied to the University of South Florida computer vision and pattern recognition group wire database showed an improvement in true positive rate (TPR) and false positive rate (FPR) when compared with the matched filter and first order derivative of Gaussian based (MF-FDOG) based PLD algorithm.

5.2 Conclusion

In the local criterion of the FF-FDOG based PLD algorithm, a line segment pool was formed by morphologically filtering an edgemap image obtained based on the application of Frangi filter (FF) and first order derivative of Gaussian (FDOG) filter on the grayscale image which resulted in over detection in order to reduce the intolerable false negative. Whole power lines were then detected in the global criterion by grouping line segments from the pool using graph cut model based on graph theory.

The developed FF-FDOG based PLD algorithm was implemented on MATLAB 2013b and used to detect power lines in images with different noise level, ambiguity, weather conditions and cluttered backgrounds, downloaded from the University of South Florida, computer vision and pattern recognition group wire database. The performance of the developed algorithm, measured in terms of TPR, FPR and time consumption, was compared with the standard MF-FDOG based PLD algorithm. It was found that the time consumptions of these two methods are almost the same (about 0.61s), but the developed FF-FDOG method was more efficient with higher TPR(86.39%),and lower FPR (11.45%) compared to MF-FDOG's 84.16% TPR and 17.91% FPR.

The developed FF-FDOG based algorithm also detected power lines effectively when applied on images of power lines captured in Samaru, Zaria.

5.3 Significant Contributions

The significant contributions of this research work are as follows:

a) Development of a FF-FDOG based thresholding scheme for detecting symmetrical edges and suppressing step edges in images of power lines.
b) The FF-FDOG based PLD outperformed the standard MF-FDOG based power line detection algorithm in by 2.64% and 36.06% improvements in terms of TPR and FPR respectively

5.4 Limitations

The limitation of this research work is highlighted as follows:

1. The captured images did not incorporate sensor jitters as images werenot taken by aerial vehicles as should have been the case.

5.5 Recommendations for Further Work

The following possible areas of further work are recommended for consideration for future research:

1. The detection at the local criterion can be further improved by introducing an approach that effectively discriminate between symmetrical and step edges.
2. The computational time can also be reduce by introducing a less complex edge detection method in the local criterion.
3. The developed method can be applied in retinal blood vessel detection.

REFERENCES

Alpatov, B., Babayan, P., & Shubin, N. (2016). *Power line detection using Integrated Vector Radon Transform.* Paper presented at the Embedded Computing (MECO), 2016 5th Mediterranean Conference on.

Avizonis, P., & Barron, B. (1999). *Low cost wire detection system.* Paper presented at the Digital Avionics Systems Conference, 1999. Proceedings. 18th.

Bhujade, R. M., Adithya, V., Hrishikesh, S., & Balamurali, P. (2013). DETECTION OF POWER-LINES IN COMPLEX NATURAL SURROUNDINGS. *Computer Science*.

Byrne, J., Cosgrove, M., & Mehra, R. (2006). *Stereo based obstacle detection for an unmanned air vehicle.* Paper presented at the Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.

Candamo, J., Kasturi, R., Goldgof, D., & Sarkar, S. (2009). Detection of thin lines using low-quality video from low-altitude aircraft in urban settings. *IEEE Transactions on Aerospace and Electronic Systems, 45*(3), 937-949.

Candamo, J., Katsuri, R., Goldgof, D., & Sarkar, S. (2010). Detection of thin lines using low-quality video from low-altitude aircraft in urban settings (Images). Available from University of South Florida, computer vision and pattern recognition group wiredata Retrieved October 24, 2016 http://marathon.cse.usf.edu/wiresdata/

Cao, W., Yang, X., Zhu, L., Han, J., & Wang, T. (2013). *Power line detection based on symmetric partial derivative distribution prior.* Paper presented at the Information and Automation (ICIA), 2013 IEEE International Conference on.

Cerón, A., & Prieto, F. (2014). *Power line detection using a circle based search with UAV images.* Paper presented at the Unmanned Aircraft Systems (ICUAS), 2014 International Conference on.

Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., & Goldbaum, M. (1989). Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on medical imaging, 8*(3), 263-269.

Chen, Y., Li, Y., Zhang, H., Tong, L., Cao, Y., & Xue, Z. (2016). Automatic power line extraction from high resolution remote sensing imagery based on an improved Radon transform. *Pattern Recognition, 49*, 174-186.

Cuijpers, R., Kappers, A., & Koenderink, J. (2002). Visual perception of collinearity. *Perception & Psychophysics, 64*(3), 392-404.

Essen, H., Boehmsdorff, S., Biegel, G., & Wahlen, A. (2002). On the scattering mechanism of power lines at millimeter-waves. *IEEE Transactions on Geoscience and Remote Sensing, 40*(9), 1895-1903.

Florack, L. M., ter Haar Romeny, B. M., Koenderink, J. J., & Viergever, M. A. (1992). Scale and the differential structure of images. *Image and Vision Computing, 10*(6), 376-388.

Frangi, A. F., Niessen, W. J., Vincken, K. L., & Viergever, M. A. (1998). *Multiscale vessel enhancement filtering.* Paper presented at the International Conference on Medical Image Computing and Computer-Assisted Intervention.

Gandhi, T., Yang, M.-T., Kasturi, R., Camps, O., Coraor, L., & McCandless, J. (2003). Detection of obstacles in the flight path of an aircraft. *IEEE Transactions on Aerospace and Electronic Systems, 39*(1), 176-191.

Gao, H., Xue, P., Lin, W., & Hou, C. (2003). *Disjoint set data structure for morphological area operators.* Paper presented at the Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on.

Garcia-Pardo, P. J., Sukhatme, G. S., & Montgomery, J. F. (2002). Towards vision-based safe landing for an autonomous helicopter. *Robotics and Autonomous Systems, 38*(1), 19-29.

Gerke, M., & Seibold, P. (2014). *Visual inspection of power lines by UAS.* Paper presented at the Electrical and Power Engineering (EPE), 2014 International Conference and Exposition on.

Greene, R. A., Levine, P., Simons, L., & Teter, R. D. (2013). Obstacle detection and notification system: Google Patents.

Jähne, B. (2005). *Digitale Bildverarbeitung*: Springer-Verlag.

Jähne, B., Haussecker, H., & Geissler, P. (1999). *Handbook of computer vision and applications* (Vol. 2): Citeseer.

Kasturi, R., & Camps, O. I. (2002). Wire detection algorithms for navigation.

Koenderink, J. J. (1984). The structure of images. *Biological cybernetics, 50*(5), 363-370.

Li, Z., Liu, Y., Walker, R., Hayward, R., & Zhang, J. (2010). Towards automatic power line detection for a UAV surveillance system using pulse coupled neural filter and an improved Hough transform. *Machine Vision and Applications, 21*(5), 677-686.

Lindeberg, T. (1996). *Edge detection and ridge detection with automatic scale selection.* Paper presented at the Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on.

Liu, Y., Mejias, L., & Li, Z. (2012). Fast power line detection and localization using steerable filter for active UAV guidance.

Luo, X., Zhang, J., Cao, X., Yan, P., & Li, X. (2014). Object-aware power line detection using color and near-infrared images. *IEEE Transactions on Aerospace and Electronic Systems, 50*(2), 1374-1389.

Lütkepohl, H. (1996). Handbook of matrices. *John Wiley&Sons*.

Martinez, C., Sampedro, C., Chauhan, A., & Campoy, P. (2014). *Towards autonomous detection and tracking of electric towers for aerial power line inspection.* Paper presented at the Unmanned Aircraft Systems (ICUAS), 2014 International Conference on.

Matikainen, L., Lehtomäki, M., Ahokas, E., Hyyppä, J., Karjalainen, M., Jaakkola, A., . . . Heinonen, T. (2016). Remote sensing methods for power line corridor surveys. *ISPRS Journal of Photogrammetry and Remote Sensing, 119*, 10-31.

McGee, T. G., Sengupta, R., & Hedrick, K. (2005). *Obstacle detection for small autonomous aircraft using sky segmentation.* Paper presented at the Proceedings of the 2005 IEEE International Conference on Robotics and Automation.

Mirallès, F., Pouliot, N., & Montambault, S. (2014). *State-of-the-art review of computer vision for the management of power transmission lines.* Paper presented at the Applied Robotics for the Power Industry (CARPI), 2014 3rd International Conference on.

Mukundan, R., & Ramakrishnan, K. (1998). *Moment functions in image analysis: theory and applications* (Vol. 100): World Scientific.

Russ, J. C. (2016). *The image processing handbook*: CRC press.

Safe Flight. (2009). Enstrom to use Safe Flight power line unit. Retrieved 14th October, 2016, from http://www.spacedaily.com/reports/Enstrom_to_use_Safe_Flight_power_line_unit_999.html

Scherer, S., Singh, S., Chamberlain, L., & Elgersma, M. (2008). Flying fast and low among obstacles: Methodology and experiments. *The International Journal of Robotics Research, 27*(5), 549-574.

Schmieder, D. E., & Weathersby, M. R. (1983). Detection performance in clutter with variable resolution. *IEEE Transactions on Aerospace and Electronic Systems*(4), 622-630.

Seibold, P., Gerke, M., Masár, I., Jelenčiak, F., Bahnik, P., & Bahr, L. (2013). Detection of power lines by image processing. *Computational Vision and Medical Image Processing IV: VIPIMAGE 2013*, 203.

Shan, H., Zhang, J., & Cao, X. (2015). *Power line detection using spatial contexts for low altitude environmental awareness.* Paper presented at the 2015 Integrated Communication, Navigation and Surveillance Conference (ICNS).

Sharma, H., Bhujade, R., Adithya, V., & Balamuralidhar, P. (2014). *Vision-based detection of power distribution lines in complex remote surroundings.* Paper presented at the Communications (NCC), 2014 Twentieth National Conference on.

Sharma, H., Dutta, T., Adithya, V., & Balamuralidhar, P. (2015). *A Real-Time Framework for Detection of Long Linear Infrastructural Objects in Aerial Imagery.* Paper presented at the International Conference Image Analysis and Recognition.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 22*(8), 888-905.

Song, B., & Li, X. (2014). Power line detection from optical images. *Neurocomputing, 129*, 350-361.

Steger, C. (1998). An unbiased detector of curvilinear structures. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 20*(2), 113-125.

Yamamoto, K., & Yamada, K. (1997). *Analysis of the infrared images to detect power lines.* Paper presented at the TENCON'97. IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications., Proceedings of IEEE.

Yang, T. W., Yin, H., Ruan, Q. Q., Da Han, J., Qi, J. T., Yong, Q., . . . Sun, Z. Q. (2012). *Overhead power line detection from UAV video images.* Paper presented at the Mechatronics and Machine Vision in Practice (M2VIP), 2012 19th International Conference.

Zhang, B., Zhang, L., Zhang, L., & Karray, F. (2010). Retinal vessel extraction by matched filter with first-order derivative of Gaussian. *Computers in biology and medicine, 40*(4), 438-445.

Zhang, J., Liu, L., Wang, B., Chen, X., Wang, Q., & Zheng, T. (2012). *High speed automatic power line detection and tracking for a UAV-based inspection.* Paper presented at the Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on.

Zhou, G., Yuan, J., Yen, I.-L., & Bastani, F. (2016). *Robust real-time UAV based power line detection and tracking.* Paper presented at the Image Processing (ICIP), 2016 IEEE International Conference on.

Zhu, L., Cao, W., Han, J., & Du, Y. (2013). *A double-side filter based power line recognition method for UAV vision system.* Paper presented at the Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on.

APPENDIX A1

MATLAB code for MF-FDOG PLD algorithm

```
% matched filter and first order derivative of gaussian power line
detection
% algorithm

I1=imread(image);
I2=rgb2gray(I1);

fim=mat2gray(I1);

%get the matched filter image response
[imx,imy]=matchedfilter(fim,1);
M=rgb2gray(imadd(imx, imy));

% use first order derivative of gaussian filter on the image
[imx,imy]=gaussgradient(fim,1);
G=rgb2gray(imy);

% reference threshold based on the image response to FDOG
c = 3;
uG = mean(mean(G));
TG = c*uG;

% adjust threshold based on the response to matched filter
r= 10;
R=(1/r).*ones(r);
M_bar=imfilter(M,R);

IM = M_bar;
IM = IM - min(IM(:));
IM = IM / max(IM(:));
Mn_bar = IM;
T = (1 + Mn_bar).*TG;

%apply threshold on the response to FDOG response
Q=G>T;

%apply morphological filtering to binary image
Q=bwmorph(Q, 'skel', inf);
bp=find(bwmorph(Q, 'branchpoints'));

for i=1:numel(bp)
    [i,j] = ind2sub(size(I2), bp(i));
if any(~[i-1:i+1,j-1:j+1])==0
Q(i-1:i+1,j-1:j+1)=0;
end

end

% remove edge pixels with less than 30 connectedness
Q=bwareaopen(Q, 30);

% filtering by thresholding the smoothness measurement.
CC = bwconncomp(Q);
for idx=1:CC.NumObjects
```

```
ind = CC.PixelIdxList{idx};

    [x,y]=ind2sub(CC.ImageSize, ind);
    p = polyfit(x,y,1);
x0 = min(x):max(x);
    y0 = polyval(p,x0, 'r');

curvexy = [x0', y0'];
mapxy = [x,y];
    [xy,distance,t] = distance2curve(curvexy,mapxy,'linear');

    S=var(distance)*100;

if (S > 110)
Q(CC.PixelIdxList{idx})=0;
end

end

%  Grouping detected line segments into power lines using graph theory
CC = bwconncomp(Q);
A = zeros(CC.NumObjects);
a=1; b=1;

for i=1:CC.NumObjects

for j=1:CC.NumObjects
        [xi, yi] = ind2sub(CC.ImageSize, CC.PixelIdxList{i});
        [xj, yj] = ind2sub(CC.ImageSize, CC.PixelIdxList{j});

if isempty(intersect(xi, xj)) && isempty(intersect(yi, yj))
if xi(1)<xj(1)
                x=[xi; xj];
                y=[yi; yj];
else
                x=[xj; xi];
                y=[yj; yi];

end
pi = polyfit(xi,yi,1);
pj = polyfit(xj,yj,1);
            p = polyfit(x,y,1);
            f = polyval(p,x, 'r');
err= mean(abs(y-f));

            [uCentre, vCentre, Ru, Rv, thetarad] = fitellipse1(x,y);
ecc = sqrt(1-(min(Ru, Rv)^2/max(Ru, Rv)^2));

            w = abs(a * exp(-b * err/ecc));

if w>0.00001
A(i,j)=1;
end
end
end

end
```

```
A=A+diag(ones(1, length(A)));
S=sparse(tril(A));
[I, J, s] = find(S);
line = struct([]);

for i=1:length(s)
pair = sort([I(i) J(i)]);
    n = 0;
for j=1:numel(line)
currentPair = line(j).segment;
if any(ismember(pair, currentPair))
            n=j;
end
end
if n == 0
line(numel(line)+1).segment = pair;
else
line(n).segment = unique([line(n).segment pair]);
end
end

% morphological properties
W = zeros(size(Q));
CC = bwconncomp(Q);

for i=1:numel(line)

C=line(i).segment ;

LInd=[];
for j=C
    LInd=vertcat(LInd, CC.PixelIdxList{j});
end

V(LInd)=0;

[x, y] = ind2sub(CC.ImageSize, sort(LInd));

p = polyfit(x,y,1);
x0= linspace(min(x),max(x), 1000)';
f = round(polyval(p,x0, 'r'));

f(find(f<1))=1;
f(find(isnan(f)))=1;
f(find(f>CC.ImageSize(2)))=CC.ImageSize(2);

x0=round(x0);
[x0 f];

W(sub2ind(CC.ImageSize,x0,f))=1;

IA=setdiff(1:length(A),C);
A = A(IA,IA);

end

W=bwmorph(W, 'skel', inf);
```

```
bp=find(bwmorph(W, 'branchpoints'));

for i=1:numel(bp)
    [i,j] = ind2sub(size(I2), bp(i));
if any(~[i-1:i+1,j-1:j+1])==0
W(i-1:i+1,j-1:j+1)=0;
end
end

% display detected result

W=W(:,1:720);

dI = I1;
rInd = find(W==1);
rChan = dI(:,:,1);
rChan(rInd) = 256;
dI(:,:,1) = rChan;

figure; imshow(dI)
title('detected result')
imwrite(dI, strcat(folder, 'detectedresult.jpg'));
```

APPENDIX A2

MATLAB code for FF-FDOG PLD algorithm

```
% frangi filter and first order derivative of gaussian power line detection
% algorithm

% read image

I1=imread(image);
I2=rgb2gray(I1);

%use frangi vesselness filter on the image
Ivessel=FrangiFilter2D(double(I2));

% use first order derivative of gaussian filter on the image
fim=mat2gray(I1);
[imx,imy]=gaussgradient(fim,1);
G=rgb2gray(imy);

% reference threshold base on the image response to FDOG
c = 3;
uG = mean(mean(G));
TG = c*uG;

% adjust threshold based on the response to frangi filter
IM = Ivessel;
IM = IM - min(IM(:));
IM = IM / max(IM(:));
Ivn = IM;
T = (1 + Ivn).*TG;

%apply threshold on the response to FDOG response
Q=G>T;

%apply morphological filtering to binary image
Q=bwmorph(Q, 'skel', inf);
bp=find(bwmorph(Q, 'branchpoints'));

for i=1:numel(bp)
    [i,j] = ind2sub(size(I2), bp(i));
if any(~[i-1:i+1,j-1:j+1])==0
Q(i-1:i+1,j-1:j+1)=0;
end

end


% remove edge pixels with less than 30 connectedness
Q=bwareaopen(Q, 30);

% filtering by thresholding the smoothness measurement.
CC = bwconncomp(Q);
for idx=1:CC.NumObjects

ind = CC.PixelIdxList{idx};
```

```
    [x,y]=ind2sub(CC.ImageSize, ind);
    p = polyfit(x,y,1);
x0 = min(x):max(x);
    y0 = polyval(p,x0, 'r');

curvexy = [x0', y0'];
mapxy = [x,y];
    [xy,distance,t] = distance2curve(curvexy,mapxy,'linear');

    S=var(distance)*100;

if (S > 110)
Q(CC.PixelIdxList{idx})=0;
end

end


%  Grouping detected line segments into power lines using graph theory
CC = bwconncomp(Q);
A = zeros(CC.NumObjects);
a=1; b=1;

for i=1:CC.NumObjects
% obtain weight for each line segments
for j=1:CC.NumObjects
        [xi, yi] = ind2sub(CC.ImageSize, CC.PixelIdxList{i});
        [xj, yj] = ind2sub(CC.ImageSize, CC.PixelIdxList{j});

if isempty(intersect(xi, xj)) && isempty(intersect(yi, yj))
if xi(1)<xj(1)
                x=[xi; xj];
                y=[yi; yj];
else
                x=[xj; xi];
                y=[yj; yi];
end

pi = polyfit(xi,yi,1);
pj = polyfit(xj,yj,1);
            p = polyfit(x,y,1);
            f = polyval(p,x, 'r');
err= mean(abs(y-f));

            [uCentre, vCentre, Ru, Rv, thetarad] = fitellipse1(x,y);
ecc = sqrt(1-(min(Ru, Rv)^2/max(Ru, Rv)^2));
            w = abs(a * exp(-b * err/ecc));

if w>0.00001
A(i,j)=1;
end
end
end

end

% represents weignts of the edges of the grapgh as adjacency matrix
A=A+diag(ones(1, length(A)));
S=sparse(tril(A));
```

```
[I, J, s] = find(S);
line = struct([]);

% group lines using  Normalised-cut
for i=1:length(s)
pair = sort([I(i) J(i)]);
    n = 0;
for j=1:numel(line)
currentPair = line(j).segment;
if any(ismember(pair, currentPair))
            n=j;
end
end
if n == 0
line(numel(line)+1).segment = pair;
else
line(n).segment = unique([line(n).segment pair]);
end
end

% morphological properties
W = zeros(size(Q));
CC = bwconncomp(Q);


% approximate lines as a quadratic polynomial
for i=1:numel(line)

C=line(i).segment ;

LInd=[];
for j=C
    LInd=vertcat(LInd, CC.PixelIdxList{j});
end

V(LInd)=0;

[x, y] = ind2sub(CC.ImageSize, sort(LInd));

p = polyfit(x,y,1);
x0= linspace(min(x),max(x), 1000)';
f = round(polyval(p,x0, 'r'));

f(find(f<1))=1;
f(find(isnan(f)))=1;
f(find(f>CC.ImageSize(2)))=CC.ImageSize(2);

x0=round(x0);
[x0 f];

W(sub2ind(CC.ImageSize,x0,f))=1;

IA=setdiff(1:length(A),C);
A = A(IA,IA);

end

W=bwmorph(W, 'skel', inf);
```

```
bp=find(bwmorph(W, 'branchpoints'));

for i=1:numel(bp)
    [i,j] = ind2sub(size(I2), bp(i));
if any(~[i-1:i+1,j-1:j+1])==0
W(i-1:i+1,j-1:j+1)=0;
end
end

% display detected result

W=W(:,1:720);

dI = I1;
rInd = find(W==1);
rChan = dI(:,:,1);
rChan(rInd) = 256;
dI(:,:,1) = rChan;

figure; imshow(dI)
title('detected result')
imwrite(dI, strcat(folder, 'detectedresult.jpg'));
```

APPENDIX B1

MATLAB code for matched filtering

```
% % Matched Filter
function [gx,gy]=matchedfilter(IM,sigma)

epsilon=1e-2;
halfsize =  ceil(sigma*sqrt(-2*log(sqrt(2*pi)*sigma*epsilon)));
size= 2*halfsize+1;
t=3;
p = normcdf([-3*sigma 3*sigma]);
s = (diff(p)/(2*t*sigma));
%generate a 2-D Gaussian kernel along x direction
for i=1:size
for j=1:size
        u=[i-halfsize-1 j-halfsize-1];
hx(i,j)=gauss(u(1),sigma)- s;
end
end
hx=hx/sqrt(sum(sum(abs(hx).*abs(hx))));
%generate a 2-D Gaussian kernel along y direction
hy=hx';
%2-D filtering
gx=imfilter(IM,hx,'replicate','conv');
gy=imfilter(IM,hy,'replicate','conv');

function y = gauss(x,sigma)
%Gaussian
y = exp(-x^2/(2*sigma^2)) / (sigma*sqrt(2*pi));

function y = dgauss(x,sigma)
%first order derivative of Gaussian
y = -x * gauss(x,sigma) / sigma^2;
```

APPENDIX B2

MATLAB code for frangi filtering

```
% Frangi Filter
function [outIm,whatScale,Direction] = FrangiFilter2D(I, options)

defaultoptions = struct('FrangiScaleRange', [1 1], 'FrangiScaleRatio', 2,
'FrangiBetaOne', 0.5, 'FrangiBetaTwo', 15,
'verbose',true,'BlackWhite',true);

% Process inputs
if(~exist('options','var')),
options=defaultoptions;
else
tags = fieldnames(defaultoptions);
for i=1:length(tags)
if(~isfield(options,tags{i})),  options.(tags{i})=defaultoptions.(tags{i});
end
end
if(length(tags)~=length(fieldnames(options))),
warning('FrangiFilter2D:unknownoption','unknown options found');
end
end

sigmas=options.FrangiScaleRange(1):options.FrangiScaleRatio:options.FrangiS
caleRange(2);
sigmas = sort(sigmas, 'ascend');

beta  = 2*options.FrangiBetaOne^2;
c     = 2*options.FrangiBetaTwo^2;

% Make matrices to store all filterd images
ALLfiltered=zeros([size(I) length(sigmas)]);
ALLangles=zeros([size(I) length(sigmas)]);

% Frangi filter for all sigmas
for i = 1:length(sigmas),
% Make 2D hessian
    [Dxx,Dxy,Dyy] = Hessian2D(I,sigmas(i));

% Correct for scale
    Dxx = (sigmas(i)^2)*Dxx;
    Dxy = (sigmas(i)^2)*Dxy;
    Dyy = (sigmas(i)^2)*Dyy;

% Calculate (abs sorted) eigenvalues and vectors
    [Lambda2,Lambda1,Ix,Iy]=eig2image(Dxx,Dxy,Dyy);

% Compute the direction of the minor eigenvector
angles = atan2(Ix,Iy);

% Compute some similarity measures
Lambda1(Lambda1==0) = eps;
    Rb = (Lambda2./Lambda1).^2;
    S2 = Lambda1.^2 + Lambda2.^2;

% Compute the output image
```

```
    Ifiltered = exp(-Rb/beta) .*(ones(size(I))-exp(-S2/c));

% see pp. 45
if(options.BlackWhite)
Ifiltered(Lambda1<0)=0;
else
Ifiltered(Lambda1>0)=0;
end
% store the results in 3D matrices
ALLfiltered(:,:,i) = Ifiltered;
ALLangles(:,:,i) = angles;
end

% Return for every pixel the value of the scale(sigma) with the maximum
% output pixel value
if length(sigmas) > 1,
    [outIm,whatScale] = max(ALLfiltered,[],3);
outIm = reshape(outIm,size(I));
if(nargout>1)
whatScale = reshape(whatScale,size(I));
end
if(nargout>2)
        Direction = reshape(ALLangles((1:numel(I))'+(whatScale(:)-
1)*numel(I)),size(I));
end
else
outIm = reshape(ALLfiltered,size(I));
if(nargout>1)
whatScale = ones(size(I));
end
if(nargout>2)
        Direction = reshape(ALLangles,size(I));
end
end

function [Dxx,Dxy,Dyy] = Hessian2D(I,Sigma)

if nargin < 2, Sigma = 1; end

% Make kernel coordinates
[X,Y]   = ndgrid(-round(3*Sigma):round(3*Sigma));

% Build the gaussian 2nd derivatives filters
DGaussxx = 1/(2*pi*Sigma^4) * (X.^2/Sigma^2 - 1) .* exp(-(X.^2 +
Y.^2)/(2*Sigma^2));
DGaussxy = 1/(2*pi*Sigma^6) * (X .* Y)           .* exp(-(X.^2 +
Y.^2)/(2*Sigma^2));
DGaussyy = DGaussxx';

Dxx = imfilter(I,DGaussxx,'conv');
Dxy = imfilter(I,DGaussxy,'conv');
Dyy = imfilter(I,DGaussyy,'conv');


function [Lambda1,Lambda2,Ix,Iy]=eig2image(Dxx,Dxy,Dyy)

% Compute the eigenvectors of J, v1 and v2
tmp = sqrt((Dxx - Dyy).^2 + 4*Dxy.^2);
v2x = 2*Dxy; v2y = Dyy - Dxx + tmp;
```

```
% Normalize
mag = sqrt(v2x.^2 + v2y.^2); i = (mag ~= 0);
v2x(i) = v2x(i)./mag(i);
v2y(i) = v2y(i)./mag(i);

% The eigenvectors are orthogonal
v1x = -v2y;
v1y = v2x;

% Compute the eigenvalues
mu1 = 0.5*(Dxx + Dyy + tmp);
mu2 = 0.5*(Dxx + Dyy - tmp);

% Sort eigen values by absolute value abs(Lambda1)<abs(Lambda2)
check=abs(mu1)>abs(mu2);

Lambda1=mu1; Lambda1(check)=mu2(check);
Lambda2=mu2; Lambda2(check)=mu1(check);

Ix=v1x; Ix(check)=v2x(check);
Iy=v1y; Iy(check)=v2y(check);
```

APPENDIX B3

MATLAB code for first order derivative of Gaussian filtering

```
% % First Order Derivative of Gaussian Filter
function [gx, gy]=gaussgradient(IM,sigma)

epsilon=1e-2;
halfsize = ceil(sigma*sqrt(-2*log(sqrt(2*pi)*sigma*epsilon)));
size= 2*halfsize+1;
%generate a 2-D Gaussian kernel along x direction
for i=1:size
for j=1:size
        u=[i-halfsize-1 j-halfsize-1];
        hx(i,j)=gauss(u(1),sigma)*dgauss(u(2),sigma);
end
end
hx=hx/sqrt(sum(sum(abs(hx).*abs(hx))));
%generate a 2-D Gaussian kernel along y direction
hy=hx';

%2-D filtering
gx=imfilter(IM,hx,'replicate','conv');
gy=imfilter(IM,hy,'replicate','conv');


function y = gauss(x,sigma)
%Gaussian
y = exp(-x^2/(2*sigma^2)) / (sigma*sqrt(2*pi));

function y = dgauss(x,sigma)
%first order derivative of Gaussian
y = -x * gauss(x,sigma) / sigma^2;
```

APPENDIX C1

MATLAB code for performance measure

```
% performance measure

%end timer
runtime = toc;

%obtain Ground truth data

GT = fopen(groundTruth);
GTData = fscanf(GT, '%d');
fclose(GT);

NumOfLines=GTData(end);
GTPoints = reshape(GTData(1:end-5), 4, NumOfLines)';

% approximate points with a single straight line and
% find the angle and the y-intercept
line_angle =[];
line_intercept1 = [];
line_intercept2 = [];

for i=1:NumOfLines

xGT=GTPoints(i,[2,4])';
yGT=GTPoints(i,[1,3])';

pGT = polyfit(xGT,yGT,1);
    line_angle = [line_angle atand(pGT(1))];
    line_intercept1 = [line_intercept1 pGT(2)];
    line_intercept2 = [line_intercept2 (0-pGT(2))/pGT(1)];
end



% check if the angle and the y-intercept between the detected lines
% and the ground truth within 10 and 20 pixels respectively
TP = 0;
FP = 0;
dl = [];
CC = bwconncomp(W);
result = [];
for i=1:CC.NumObjects
    [x, y] = ind2sub(CC.ImageSize, CC.PixelIdxList{i}s);
    p = polyfit(x,y,1);
result = [result; [atand(p(1)) p(2) (0-p(2))/p(1)]];
    d=find((line_angle < atand(p(1)) + 10) & (line_angle > atand(p(1)) -
10)...
& (line_intercept2 < (0-p(2))/p(1) + 20) & (line_intercept2 > (0-p(2))/p(1)
- 20));
if isempty(d)
        FP = FP + 1;
else
if (~ismember(d, dl))
dl = [dl d];
            TP = TP + 1;
end
```

```
end
end

% estimate TPR & FPR
TPR = (TP/NumOfLines) * 100;
FPR = (FP/CC.NumObjects) * 100;
```

APPENDIX C2

Data for MF-FDOG performance measure

Runtime	TPR	FPR
0.4282	60.00	0.00
1.2329	60.00	20.00
1.4791	100.00	25.00
0.6689	100.00	16.67
0.7580	100.00	50.00
0.4327	100.00	0.00
0.5353	100.00	0.00
0.2589	100.00	0.00
0.1551	0.00	0.00
1.0541	100.00	0.00
0.3211	100.00	0.00
0.5033	100.00	0.00
0.2376	33.33	0.00
0.4843	100.00	30.00
2.7548	71.43	12.50
0.2316	50.00	0.00
0.1526	0.00	0.00
0.2483	100.00	0.00
0.2749	100.00	0.00
0.3522	50.00	80.00
0.4723	100.00	0.00
0.7337	100.00	0.00
0.5379	100.00	11.11
0.2706	33.33	0.00
0.4378	66.67	45.45
0.4051	100.00	66.67
0.3477	100.00	0.00

0.3319	100.00	0.00
0.5099	100.00	20.00
0.2818	100.00	33.33
0.2308	100.00	0.00
1.5263	100.00	61.90
0.7485	100.00	80.00
0.5298	42.86	10.00
2.6678	71.43	55.17
0.6483	100.00	50.00
1.0420	100.00	40.00
0.4851	100.00	27.27
0.5496	83.33	0.00
0.3460	100.00	0.00
0.3430	75.00	0.00
0.4061	66.67	16.67
0.2721	75.00	0.00
0.6694	100.00	16.67
0.8908	100.00	31.25
0.5042	57.14	44.44
0.3585	100.00	0.00
1.0171	100.00	0.00
0.2552	100.00	0.00
0.2794	83.33	0.00

APPENDIX C3

Data for FF-FDOG performance measure

Runtime	TPR	FPR
0.5131	60.00	0.00
0.6617	55.56	0.00
0.5671	100.00	0.00
0.7932	100.00	0.00
0.4281	100.00	0.00
0.4978	100.00	0.00
0.6512	100.00	0.00
0.3431	100.00	0.00
0.8162	100.00	0.00
0.5980	100.00	0.00
0.4107	100.00	0.00
0.5953	100.00	0.00
0.3197	33.33	0.00
0.4260	75.00	0.00
2.5510	71.43	12.50
0.3655	66.67	0.00
0.3547	100.00	0.00
0.3400	100.00	0.00
0.3272	100.00	0.00
0.3888	50.00	80.00
1.1026	100.00	6.25
0.7691	100.00	0.00
0.6892	85.71	9.09
0.3954	66.67	25.00
0.5410	40.00	0.00
0.3480	100.00	33.33
0.4019	100.00	0.00

0.3699	100.00	0.00
0.4432	75.00	0.00
0.3749	100.00	0.00
0.3602	100.00	0.00
0.7589	100.00	33.33
0.3977	100.00	75.00
0.6538	57.14	10.00
2.0731	71.43	52.17
0.7596	100.00	50.00
0.8531	75.00	25.00
0.5855	100.00	30.00
0.6779	83.33	0.00
0.4256	100.00	0.00
0.5455	100.00	23.08
0.5769	66.67	0.00
0.3407	75.00	0.00
0.7106	100.00	20.00
1.1140	100.00	27.78
0.4494	28.57	0.00
0.4838	100.00	0.00
1.2218	100.00	0.00
0.3951	100.00	16.67
0.3726	83.33	0.00

APPENDIX D1

MATLAB code for clutter measure

```
function clutterI = clutterMeasure(image)
% read and convert image to grayscale
I = rgb2gray(imread(image));
% obtain the variance for each sub-image window
iVar = [];
for i=1:120:480
for j=1:160:640
       w = I(i:i+119, j:j+159);
       x = reshape(w, numel(w), 1);
iVar = [iVar var(double(x))];
size(x);
end
end
% calculate the image clutter
k =16;
clutterI = sqrt(sum(iVar)/k);
```

APPENDIX D2

Data for clutter measure

Clutter measure
5.6269
5.9322
11.8735
10.0389
7.0076
28.4395
28.3943
17.5583
17.8650
8.0471
13.0384
17.3890
7.9971
18.9302
9.9145
8.3997
11.6572
17.9443
18.8453
37.8687
20.7470
38.5275
37.1691
52.7521
23.9317
27.6282
41.4949

27.5141

35.9325

31.3997

48.3777

32.5707

29.3125

21.2537

21.2235

28.4716

40.4435

28.3558

19.7771

33.8753

20.7234

18.8217

29.3841

34.2047

15.4956

22.2851

35.7969

30.2020

24.7826

16.8873

41.2631

35.0039