

A FAST SHAPE DETECTION APPROACH BY DIRECTIONAL INTEGRATIONS

**A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY**

BY

OSMAN ERMAN OKMAN

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING**

SEPTEMBER 2013

Approval of the thesis:

A FAST SHAPE DETECTION APPROACH BY DIRECTIONAL INTEGRATIONS

submitted by **OSMAN ERMAN OKMAN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Gönül Turhan Sayan
Head of Department, **Electrical and Electronics Eng.**

Prof. Dr. Gözde Bozdağı Akar
Supervisor, **Electrical and Electronics Eng. Dept., METU**

Examining Committee Members:

Prof. Dr. A. Aydın Alatan
Electrical and Electronics Eng. Dept., METU

Prof. Dr. Gözde Bozdağı Akar
Electrical and Electronics Eng. Dept., METU

Prof. Dr. Şebnem Düzgün
Mining Eng. Dept., METU

Assoc. Prof. Dr. İlkay Ulusoy
Electrical and Electronics Eng. Dept., METU

Assoc. Prof. Dr. Selim Aksoy
Computer Eng. Dept., Bilkent University

Date: 06.09.2013

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Osman Erman Okman

Signature :

ABSTRACT

A FAST SHAPE DETECTION APPROACH BY DIRECTIONAL INTEGRATIONS

Okman, Osman Erman

**Ph.D., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Gözde Bozdağı Akar**

September 2013, 164 pages

Detection and identification of objects from aerial images are important problems for various types of application areas. For many of the man-made structures shape is a fundamental feature by which these objects are separated from the background and other structures. In this thesis, a novel geometric shape detection algorithm based on the spatial properties of structures is proposed. Since the objects are transformed into 1-D vectors by evaluating directional integrals and detections occur by the analysis of the features obtained in those 1-D spaces, the proposed approach requires less memory and computation than most of the approaches reported in the literature. Detailed derivation of the method is given and the experimental results are presented in order to show the detection performance of the method under different amount of noise and geometric deformations. Experimental results on real images also show that the proposed approach can significantly speed up the computation without degrading the performance. Moreover, a Petroleum Oil Lubricants (POL) depots identification procedure in high-resolution satellite images is developed where detection of the circular structures is one of the crucial steps, which is achieved by the proposed shape detection approach. Performed experiments over a large data set imply promising identification performance and the usability of the shape detection approach in real world applications.

Keywords: Shape Detection, Hough Transform, POL Identification, Land Cover/Use Classification, High Resolution Satellite Images

ÖZ

YÖNLÜ İNTEGRALLER İLE HIZLI BİR GEOMETRİK ŞEKİL TESPİT YÖNTEMİ

Okman, Osman Erman

Doktora, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Gözde Bozdağı Akar

Eylül 2013, 164 sayfa

Havadan çekilen görüntülerin kullanımı ile nesnelerin tespit edilmesi bir çok uygulama alanında karşılaşılan önemli bir problemdir. İnsan yapımı birçok nesne için bu nesneleri diğer yapılardan ve arka plandan ayıran en önemli özelliklerinden birisi ise bu nesnelerin şekilleridir. Bu tez kapsamında, nesnelerin uzamsal özelliklerini temel alan yeni bir geometrik şekil tespit algoritması önerilmektedir. Bu yöntemde bulunan kenarlar yönlü entegrallerin hesaplanması ile 1-B vektörlere dönüştürülmekte ve tespit işlemi bu 1-B dönüşüm uzayında gerçekleştirildiği için önerilen yöntem literatürde sıkça kullanılan yöntemlere göre daha az belleğe ve işlemci gücüne ihtiyaç duymaktadır. Bu tezde, önerilen metod için detaylı matematiksel türetimler verilmekte, yapılan deneyler ile üretilen çıktılar sunulmakta ve değişik gürültü seviyeleri ve geometrik bozunum durumlarında bu yöntemin tespit performansı verilmektedir. Çeşitli kullanım alanları için bulunan gerçek görüntüler üzerinde de deneyler yapılmış ve yapılan deneyler ile bu yöntemin tespit performansını bozmaksızın diğer yöntemlere sağladığı hız avantajı gösterilmiştir. Ayrıca bu tez kapsamında, yüksek çözünürlüklü uydu görüntüleri kullanılarak Petrol Gaz Yağları (Petroleum Oil Lubricants, POL) depolarının bulunmasına yönelik bir yöntem de sunulmaktadır. Bu yöntemdeki esas aşamalardan bir tanesi görüntüdeki dairesel şekillerin bulunması olup, bu işlem tez kapsamında önerilen yöntem ile başarılmıştır. Geniş bir veri kümesi üzerinde yapılan denemeler yöntemin başarımının yüksek olduğunu ve önerilen şekil bulma yaklaşımının birçok uygulamada kullanılabilir olduğunu göstermiştir.

Anahtar Kelimeler: Şekil Tespiti, Hough Dönüşümü, POL Tespiti, Dünya Yüzeyi Örtü/Kullanım Sınıflandırması, Yüksek Çözünürlüklü Uydu Görüntüleri

To my family

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor Prof. Dr. Gözde Bozdağı Akar for her guidance, criticism, support, encouragement, insight and friendship throughout the research.

I would like to thank to Multimedia Research Group (MMRG) members for their technical support. Besides, I spent great time in all our activities in these years.

I also would like to thank all my colleagues at TUBITAK ILTAREN and SDT Inc. for their support and since they share their knowledge and experience with me.

My deepest thanks and love go to my parents and my sister for their endless love, support and encouragement throughout my whole life.

Finally I would like to thank to my wife and my love Nutiye. She was always with me and gives me a lot of courage and strength throughout the most stressful periods of this thesis, which is impossible to be finished without her endless support and patience.

TABLE OF CONTENTS

ABSTRACT.....	v
ÖZ.....	vi
ACKNOWLEDGEMENTS.....	viii
TABLE OF CONTENTS.....	ix
LIST OF TABLES.....	xi
LIST OF FIGURES.....	xii
LIST OF ABBREVIATIONS.....	xvii
CHAPTERS	
1. INTRODUCTION.....	1
1.1 Scope of the Thesis.....	3
1.2 Outline of the Thesis.....	4
2. CIRCLE DETECTION.....	5
2.1 Modified Hough Transform (MHT).....	7
2.2 Randomized Circle Detector (RCD).....	8
2.3 Circle Detection by Directional Integrations (CDDI).....	9
2.3.1 Analysis at Discrete Domain.....	11
2.3.2 Proposed Circle Detection Algorithm.....	16
2.4 Effects of Edge Detection Approaches on CDDI.....	20
2.5 Experimental Results.....	27
2.6 Parameter Selection for CDDI.....	43
2.7 Discussion.....	44
3. RECTANGLE DETECTION.....	45
3.1 Windowed Hough Transform (WHT).....	46
3.2 Rectangle Detection by Directional Integrations (RDDI).....	48
3.2.1 Proposed Rectangle Detection Algorithm.....	52
3.3 Experimental Results.....	57
3.4 Parameter Selection for RDDI.....	65
3.5 Discussion.....	65
4. ELLIPSE DETECTION.....	67
4.1 Ellipse Detection by Directional Integrations (EDDI).....	68

4.1.1	Analysis at Discrete Domain	70
4.1.2	Proposed Ellipse Detection Algorithm	77
4.2	Experimental Results.....	80
4.3	Parameter Selection for EDDI.....	87
4.4	Discussion	87
5.	AN APPLICATION: RECOGNITION OF GEOMETRICAL OBJECTS IN THE INDUSTRIAL REGIONS	89
5.1	Detection and Recognition of POLs.....	89
5.1.1	Land Cover/Use Classification.....	90
5.1.2	Mask Extraction for Industrial Regions.....	102
5.1.3	Prefiltering.....	103
5.1.4	Circle Detection.....	103
5.1.5	Post Processing and POL Recognition	104
5.1.6	Experimental Results.....	105
5.1.7	Comparison of Circle Detection Approaches in Terms of POL Detection	112
5.1.8	Discussion.....	122
5.2	Detection of Rectangular Buildings in the Industrial Regions	123
5.2.1	Discussion.....	126
6.	CONCLUSION	127
	REFERENCES	131
	APPENDICES	
A	EVALUATION OF PROFILE FUNCTIONS FOR ROTATED ELLIPSES	139
B	FUZZY C-MEANS CLUSTERING	143
C	DATA SET USED IN POL DETECTION	145
	VITA	165

LIST OF TABLES

TABLES

Table 1 MHT Algorithm.....	8
Table 2 CDDI circle detection procedure.	19
Table 3 Execution durations of different methods in seconds for different images.....	33
Table 4 Execution durations and performance measures for POL images data set of different approaches.....	43
Table 5 RDDI rectangle detection procedure.....	56
Table 6 Execution durations of WHT in connected components and RDDI in seconds for different images.	60
Table 7 EDDI ellipse detection procedure.....	80
Table 8 Execution durations of EDDI in seconds for different images.	84
Table 9 Number of pixels used in training of SVMs and their coverage areas according to the cluster types.....	98
Table 10 Feature sets used for classification experiments.	99
Table 11 Confusion matrix when only color features are used (Feature Set 1).	99
Table 12 Confusion matrices when only texture features are used for (a) Feature Set 2, (b) Feature Set 3, (c) Feature Set 4, (d) Feature Set 5.	100
Table 13 Confusion matrix when color features and texture features with different parameters are used (Feature Set 6).	100
Table 14 POL recognition performance of the proposed schema evaluated using test set.	106
Table 15 POL recognition performances for the images given from Figure 65 to Figure 69.	106
Table 16 POL recognition performance of the proposed schema when MHT is used for circle detection.	112
Table 17 POL recognition performance of the proposed schema when MHT CC is used for circle detection.	112
Table 18 POL recognition performance of the proposed schema when RCD CC is used for circle detection.	113
Table 19 Average computation duration of circle detection methods for one image in seconds.....	113

LIST OF FIGURES

FIGURES

Figure 1 General Object Recognition Scheme for Remote Sensing Images	2
Figure 2 Necessary complexity for data, model and strategy.....	3
Figure 3 Differential elements on the circle.	10
Figure 4 (a) $g_1(x)$ and (b) $g_2(y)$ for a circle of radius 40 centered at (50, 60). Both x and y axes are sampled by 0.1 units.	11
Figure 5 (a) A synthetic circle image of radius 40 centered at (50, 60) and evaluated directional summations (b) $g_1[m]$ and (c) $g_2[n]$ from this image.	14
Figure 6 (a) A sample image containing coins, (b) its edge map using Canny detector, (c) manually extracted circular structure from the edge map (d) estimated circle from (e) vertical profile, $g_1[m]$, and (f) horizontal profile $g_2[n]$	15
Figure 7 (a) A sample POL image, (b) its edge map using Canny detector, (c) manually extracted circular structure from the edge map (d) estimated circle from (e) vertical profile, $g_1[m]$, and (f) horizontal profile $g_2[n]$	16
Figure 8 (a) A circle with radius 50 and its noise versions when (b) $L_{\text{Noise}}=0.1$, (c) $L_{\text{Noise}}=0.5$, (d) $L_{\text{Noise}}=1$	23
Figure 9 Edges detected by (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny methods when $L_{\text{Noise}} = 0$	24
Figure 10 Edges detected by (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny methods when $L_{\text{Noise}} = 0.1$	24
Figure 11 Edges detected by (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny methods when $L_{\text{Noise}} = 0.5$	24
Figure 12 Edges detected by (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny methods when $L_{\text{Noise}} = 1$	24
Figure 13 Detected circles by CDDI using (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny edge detectors when $L_{\text{Noise}} = 0$	25
Figure 14 Detected circles by CDDI using (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny edge detectors when $L_{\text{Noise}} = 0.1$	25
Figure 15 Detected circles by CDDI using (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny edge detectors when $L_{\text{Noise}} = 0.5$	25
Figure 16 Detected circles by CDDI using (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny edge detectors when $L_{\text{Noise}} = 1$	25
Figure 17 (a) A satellite image of a POL with its edge maps and detected circles by CDDI using (b), (g) Robert's; (c), (h) Prewitt; (d), (i) Sobel; (e), (j) LoG; (f), (k) Canny method.	26
Figure 18 (a) A satellite image of a POL with its edge maps and detected circles by CDDI using (b), (g) Robert's; (c), (h) Prewitt; (d), (i) Sobel; (e), (j) LoG; (f), (k) Canny method.	26
Figure 19 (a) An example for test images with Salt & Pepper noise added where $L_{\text{Noise}} = 1$, (b) edge map of the image with noise, (c) detected circles by CDDI.....	27
Figure 20 (a) Detection probability and (b) false alarm probability of the proposed method with respect to the added noise level.	28

Figure 21 ROC curves of each method when L_{Noise} is equal to (a) 0.1, (b) 0.5, (c) 1, (d) 1.5.	29
Figure 22 (a) Circle edges with missing parts and detection results of CDDI when (b) α, β, γ are 0.5, λ is 0.75, μ_{\min} is 0.25 and μ_{\max} is 1.75, (c) α, β, γ are 0.25, λ is 0.75, μ_{\min} is 0.25 and μ_{\max} is 1.75.....	30
Figure 23 (a) Circles connected with other shapes and (b) detected circles using CDDI when $\alpha, \beta, \gamma, \lambda$ are 0.75, μ_{\min} is 0.5, μ_{\max} is 1.5, n_n is 2 and n_p is 5.	31
Figure 24 Computation durations of MHT, RCD and CDDI for different number of circles in the image.....	31
Figure 25 Computation durations of MHT and RCD when evaluated for each connected component and CDDI for different number of circles in the image when noise level is (a) 0, (b) 0.1, (c) 0.3.	32
Figure 26 (a) Coin image, (b) its edge map and circle detections using (c) MHT, (d) MHT in connected components, (e) RCD, (f) RCD in connected components and (g) CDDI....	34
Figure 27 (a) Eye image, (b) its edge map and circle detections using (c) MHT, (d) MHT in connected components, (e) RCD, (f) RCD in connected components and (g) CDDI.....	35
Figure 28 (a) Football image, (b) its edge map and circle detections using (c) MHT, (d) MHT in connected components, (e) RCD, (f) RCD in connected components and (g) CDDI.....	36
Figure 29 (a) A satellite image of an oil refinery taken from Google Earth, (b) its edge map and circle detections using (c) MHT, (d) MHT in connected components, (e) RCD in connected components and (f) CDDI.....	37
Figure 30 (a) An example image or POL test data set and (b) its ground truth image.	38
Figure 31 (a) An example image or POL test data set and (b) its ground truth image.	39
Figure 32 Detected circles on the image given in Figure 30 by (a) MHT, (b) RCD in connected components when $n_n=2$, (c) CDDI when $n_n=1$ (d) CDDI when $n_n=2$	41
Figure 33 Detected circles on the image given in Figure 31 by (a) MHT, (b) RCD in connected components when $n_n=2$, (b) CDDI when $n_n=1$ (c) CDDI when $n_n=2$	42
Figure 34 Representation of a line in polar coordinates.	46
Figure 35(a) A rectangle centered at the origin and (b) its linear HT[52].	47
Figure 36 A rectangle and its parameters x_c, y_c, a, b and θ	48
Figure 37 (a) $g_1[m]$ and (b) $g_2[n]$ for a rectangle when $\theta=0$	49
Figure 38 Rectangle with $\theta>0$	49
Figure 39 (a) $g_1[m]$ and (b) $g_2[n]$ for a rectangle when $\theta\neq 0$	50
Figure 40 Possible values of integration of a rectangle in horizontal or vertical directions with respect to its orientation	51
Figure 41 Detection probability of RDDI with respect to the added noise level.	57
Figure 42 (a) An example for test images with Salt & Pepper noise added where $L_{\text{Noise}} = 1$, (b) edge map of the image with noise, (c) detected rectangles by RDDI.	58
Figure 43 Computation durations of WHT and RDDI for different number of rectangles in the image.....	59
Figure 44 Computation durations of WHT when evaluated for each connected component and RDDI for different number of rectangles in the image when noise level is (a) 0.01, (b) 0.1 and (c) 0.5.	60
Figure 45 (a) A car image, (b) its edge map and rectangle detections using (c) WHT in CC and (d) RDDI.	61

Figure 46 (a) A building image, (b) its edge map and rectangle detections using (c) WHT in CC and (d) RDDI.	62
Figure 47 (a) A satellite image of an industrial zone taken from Google Earth, (b) its edge map and rectangle detections using (c) WHT in CC and (d) RDDI.	63
Figure 48 An ellipse and its parameters x_c, y_c, a, b and θ	69
Figure 49 (a) $g_1(x)$ and (b) $g_2(y)$ for an ellipse with $a=50, b=20$ and centered at (70, 100). Both x and y axes are sampled by 0.1 units.	70
Figure 50 (a) A synthetic ellipse image with parameters $a=50, b=20$, centered at (70, 100) and evaluated directional summations (b) $g_1[m]$ and (c) $g_2[n]$ from this image.	74
Figure 51 (a) A speaker image, (b) its edge map using Canny detector, (c) manually extracted elliptical structure from the edge map (d) estimated ellipse from (e) vertical profile, $g_1[m]$, and (f) horizontal profile $g_2[n]$	75
Figure 52 (a) A sample image containing a dish antenna, (b) its edge map using Canny detector, (c) manually extracted and rotated elliptical structure from the edge map (d) estimated ellipse from (e) vertical profile, $g_1[m]$, and (f) horizontal profile $g_2[n]$	76
Figure 53 Detection probability of EDDI with respect to the added noise level.	81
Figure 54 (a) An example for test images with Salt & Pepper noise added where $L_{\text{Noise}} = 1$, (b) edge map of the image with noise, (c) detected ellipses by EDDI.	82
Figure 55 Computation duration of EDDI for different number of ellipses in the image. ...	83
Figure 56 Computation duration of EDDI for different number of ellipses in the image when noise level is (a) 0.01, (a) 0.1 and (c) 0.5.	83
Figure 57 (a) The image in Figure 51(a), (b) its edge map and (c) automatically detected ellipse by EDDI.	84
Figure 58 (a) The image in Figure 52 (a), (b) its edge map and (c) automatically detected ellipses by EDDI.	85
Figure 59 (a) A satellite image of a garden taken from Google Earth, (b) its edge map and ellipse detections using (c) EDDI.	86
Figure 60 Block diagram for POL recognition process.	90
Figure 61 Procedure for GLCM computation when $d=1$ and $\theta=0$	95
Figure 62 An example for (a) a satellite image and (b) its classification result.	101
Figure 63 An example for (a) a satellite image and (b) its classification result.	102
Figure 64 Two possible connected circle and defined region where mean intensities evaluated.	105
Figure 65 (a) An example image from test set, (b) obtained industrial region map, (c) edge map and (d) recognized POLs on this image.	107
Figure 66 (a) An example image from test set, (b) obtained industrial region map, (c) edge map and (d) recognized POLs on this image.	108
Figure 67 (a) An example image from test set, (b) obtained industrial region map, (c) edge map and (d) recognized POLs on this image.	109
Figure 68 (a) An example image from test set, (b) obtained industrial region map, (c) edge map and (d) recognized POLs on this image.	110
Figure 69 (a) An example image from test set, (b) obtained industrial region map, (c) edge map and (d) recognized POLs on this image.	111
Figure 70 Recognized POLs on the image in Figure 65 using (a) MHT, (b) MHT CC and (c) RCD CC.	115

Figure 71 Recognized POLs on the image in Figure 66 using (a) MHT, (b) MHT CC and (c) RCD CC.....	116
Figure 72 Recognized POLs on the image in Figure 67 using (a) MHT, (b) MHT CC and (c) RCD CC.....	117
Figure 73 Recognized POLs on the image in Figure 68 using (a) MHT, (b) MHT CC and (c) RCD CC.....	118
Figure 74 Recognized POLs on the image in Figure 68 using (a) MHT, (b) MHT CC and (c) RCD CC.....	119
Figure 75 POL-wise precision values for each of the image in the data set when CDDI, MHT, MHT CC and RCD CC are used for detection of the circles.	120
Figure 76 POL-wise recall values for each of the image in the data set when CDDI, MHT, MHT CC and RCD CC are used for detection of the circles.	121
Figure 77 (a) An image containing dark colored POLs and (b) the detected POLs by the proposed algorithm using CDDI.	122
Figure 78 (a) An example image from the data set, (b) mask used for rectangle detection, (c) used edge map and (d) detected rectangles by RDDI.....	124
Figure 79 (a) An example image from the data set, (b) mask used for rectangle detection, (c) used edge map and (d) detected rectangles by RDDI.....	125
Figure C.1	145
Figure C.2	145
Figure C.3	145
Figure C.4	146
Figure C.5	146
Figure C.6	146
Figure C.7	146
Figure C.8	147
Figure C.9	147
Figure C.10	147
Figure C.11	147
Figure C.12	148
Figure C.13	148
Figure C.14	148
Figure C.15	148
Figure C.16	149
Figure C.17	149
Figure C.18	149
Figure C.19	149
Figure C.20	150
Figure C.21	150
Figure C.22	150
Figure C.23	150
Figure C.24	151
Figure C.25	151
Figure C.26	151
Figure C.27	151
Figure C.28	152

Figure C.29.....	152
Figure C.30.....	152
Figure C.31.....	152
Figure C.32.....	153
Figure C.33.....	153
Figure C.34.....	153
Figure C.35.....	153
Figure C.36.....	154
Figure C.37.....	154
Figure C.38.....	154
Figure C.39.....	154
Figure C.40.....	155
Figure C.41.....	155
Figure C.42.....	155
Figure C.43.....	155
Figure C.44.....	156
Figure C.45.....	156
Figure C.46.....	156
Figure C.47.....	156
Figure C.48.....	157
Figure C.49.....	157
Figure C.50.....	157
Figure C.51.....	157
Figure C.52.....	158
Figure C.53.....	158
Figure C.54.....	158
Figure C.55.....	158
Figure C.56.....	159
Figure C.57.....	159
Figure C.58.....	159
Figure C.59.....	159
Figure C.60.....	160
Figure C.61.....	160
Figure C.62.....	160
Figure C.63.....	160
Figure C.64.....	161
Figure C.65.....	161
Figure C.66.....	161
Figure C.67.....	161
Figure C.68.....	162
Figure C.69.....	162
Figure C.70.....	162

LIST OF ABBREVIATIONS

POL	Petroleum Oil Lubricants
ATR	Automatic Target Recognition
HT	Hough Transform
MHT	Modified Hough Transform
WHT	Windowed Hough Transform
RHT	Randomized Hough Transform
RCD	Randomized Circle Detector
CDDI	Circle Detection by Directional Integrations
RDDI	Rectangle Detection by Directional Integrations
GLCM	Gray Level Co-occurrence Matrix
SVM	Support Vector Machines
ROI	Region of Interest

CHAPTER 1

INTRODUCTION

The process of measuring the object properties on the earth's surface using the data acquiring from air vehicles or satellite is called remote sensing, [1]. Some important applications of remote sensing can be listed as:

- Environmental assessment and monitoring
- Renewable/nonrenewable resource exploitation
- Agricultural condition assessment and yield prediction
- Weather prediction
- Mapping
- Global change detection
- Military surveillance and reconnaissance
- News media

For each of these applications, one of the main assets is detecting objects; correlate them with the already known ones and classifying them according to their spatial, spectral or temporal properties, which constitutes the main effort in this thesis in order to automatically interpret the images.

The automatic interpretation of the satellite or airborne images is an old and very difficult problem. This problem has been very popular since the very first civilian satellite image was acquired in the early 1970s. The popularity is caused of the possibility of gathering information about the interested area in a shorter time and with relatively less cost by using such a method. However, according to the needs of the applications from very different fields, it is not possible to find a general solution. Hence each study in the literature focuses on their goals and characteristics of the data they have.

Man-made structures are definitely one of the most widely studied targets to be automatically detected because of their importance in any type of application. The building detection and recognition problem is examined in [2] and it is stated that the building detection and recognition is such a complex problem so it cannot be obtained by a single step procedure. It is possible to generalize this conclusion to any application for automatic interpretation of the aerial images and the steps for such an approach can be summarized as given in Figure 1.

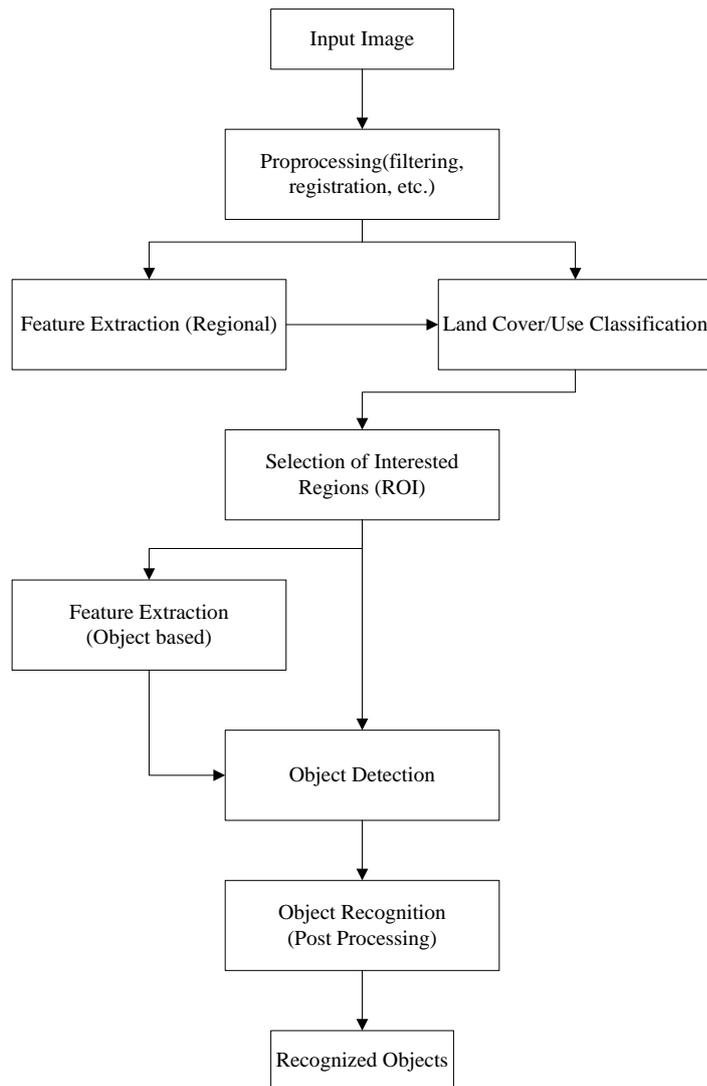


Figure 1 General Object Recognition Scheme for Remote Sensing Images

Nowadays, remote sensing images are acquired using very different types of sensors from conventional visible band Electro-optical cameras to hyperspectral imaging modalities covering different optical bands with hundreds of spectral samples and from radar imagers to laser imaging devices. Despite, these defined steps of processing are common in a sense; they may differ in some manners because of the nature of the obtained images or resolution they have. Obviously, the data obtained is increasing in amount day by day and for each of application field and any imaging modality it is required to utilize automatic systems that detects the objects of interest fast and precisely. In [3], it is stated that the complexity of images, the resolution and the preferred approach have to be as complex as possible but they should be carefully tuned not to introduce further difficulties or requirement for processing as seen in Figure 2.

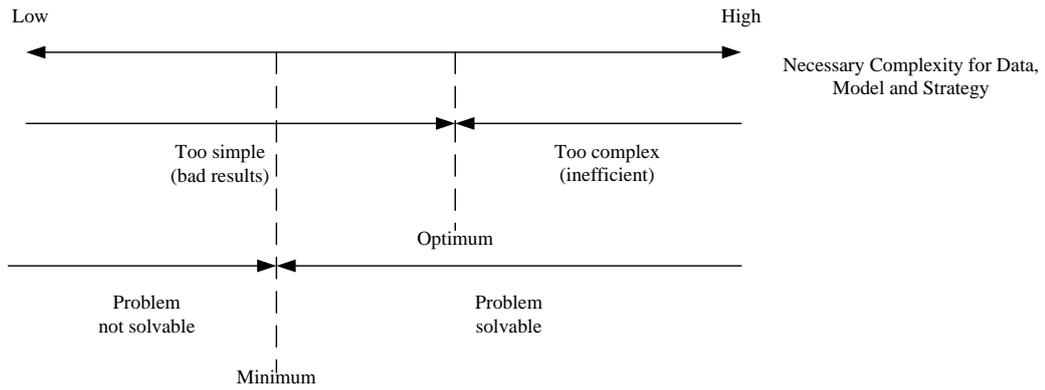


Figure 2 Necessary complexity for data, model and strategy.

Detection of geometrical shapes is one of the main problems of object detection since many of the man-made objects are in these shapes. In computer vision, identification of two dimensional (2-D) shapes is a fundamental problem, where shapes are defined as 2-D features of an object whose variation to scene factors can be modeled easily [4]. Imitating the human visual system is one of the alternatives to solve this problem where it is thought for this system to make use of many different kinds of information in the scene to detect these shapes. However, the researches show that edge and region information are utilized by human vision to extract a perceptual unit in the scene at the lowest level of recognition [5]. Hence, directional derivatives of image intensity are investigated to get information about object boundaries and those candidate boundaries are used either for detection of an object or identification of the object's shape if there is a priory clue about its existence.

For detection of the shapes many different approaches have been proposed in the literature. Upon these methods, one of the very first and very widely used one is Hough Transform (HT). HT is proposed by Paul Hough in 1962 for detecting the alignments of television lines [6]. It is than adopted for detection of circles and ellipses [7] and finally Ballard showed that HT can be used for any parametric shape detection by which he proposed the Generalized Hough Transform (GHT) [8]. The simplicity of the idea and robustness against noise make this approach popular where the computation load makes it infeasible for some of the applications. There are some other parametric shape detection approaches in the literature and they are examined in the following sections where another fast method is proposed. After presenting the detection performance of the proposed algorithm, it is compared with the performances of commonly used circle detection algorithms in the literature. Besides, the successes of these methods are investigated using different real world scenes in order to understand their behavior under different circumstances and the performance of the proposed approach is examined in terms of the detection of the some critical structures in the industrial zones by use of the aerial optical images.

1.1 Scope of the Thesis

In this thesis, the main goal is to obtain a fast geometrical shape detection approach especially for detection of the manmade objects from big data like remote sensing images. The approach is implemented for detection of circles, rectangles and ellipses and performances of these methods are examined by comparing the frequently used approaches

in the literature (i.e. Hough Transform). A possible application, where the proposed shape detection approach can be used, is the detection of the common structures in the industrial areas from the visible band satellite images. Large rectangle buildings and the circular oil depots (Petroleum, Oil, & Lubricants, POL) are selected as the main structures to be detected in these areas since their shapes are the main feature to identify these structures. Then, a procedure for detection of these structures is proposed in this thesis. In this context, identification of the industrial areas is another concern of this thesis and a classification based approach is utilized to achieve this goal. Further processing steps are also proposed after detection of the interested objects in order to reduce the false detections and a complete approach for detection of the common structures in the industrial areas is established. This resulting algorithm is the second contribution of this thesis since, to the best of our knowledge, there is no other similar work in the literature. Besides, the performance of this algorithm is presented using a large data set taken from Google Earth™ to validate the applicability of the shape detection approach for real world applications.

1.2 Outline of the Thesis

The outline of the thesis is as follows. In the following chapters the proposed geometrical shape detection approach is explained with the existing approaches in the literature. In CHAPTER 2 the proposed circle detection method is explained, the performance of the method is analyzed and compared with the commonly used methods in the literature. Similarly, 0 focuses on rectangle detection and the details of the proposed algorithm are given with some analysis and comparisons. In 0 the proposed ellipse detection approach is given and its performance is analyzed. 0 includes the whole scheme for detection of the POLs in the industrial areas including the land cover/use classification approach used in the thesis, the pre-processing and post-processing steps to reduce the false alarms and performance of the approach is presented. Detection of rectangular structures in these industrial regions is also analyzed in this chapter. Finally 0 concludes the thesis and suggests items for possible future work.

CHAPTER 2

CIRCLE DETECTION

As stated previously shape identification is one of the essential problems of pattern analysis and automatic target recognition (ATR) where detection of circles is one of the important steps of this process. Due to this, circle detection problem has been extensively studied in the literature and many different approaches have been developed, [9].

The approaches used for circle detection can be classified as accumulator based and non-accumulator based. Hough Transform (HT) is the most popular accumulator based method and it is still widely used because of its robustness against noise, clutter, object defects, shape distortions, etc. [6, 7, 10]. On the other hand, this method requires massive memory and excessive computation power.

For efficient detection of circles, many researches proposed modifications to the HT, [9, 11-24]. In [18], accumulating only the points in the gradient direction of the edges is proposed. This revised method is standardized as Modified Hough Transform (MHT) in [1,19] where the information about the local edge direction is extracted using Sobel operator. There are many other studies trying to reduce the computational load or memory consumption of HT using different voting strategies, transform spaces or validation scheme [11-16, 21]. In [25], some of these proposed modifications are combined and shown to be equal to a size invariant filtering operation. However, the improvements in computation times of these methods are far from being significant.

Xu et. al. introduced Randomized Hough Transform (RHT) [23, 24], by which the computational performance of HT is improved drastically. They proposed to select n pixels randomly and map them to a point in the parameter space, instead of transforming each pixel into n -dimensional hyperspace. RHT is also extended for detection of ellipses in [19, 20] and some improved sampling strategies and solutions to reduce the localization error are proposed in [17, 22].

In addition to the methods mentioned above, literature includes a vast amount of accumulator based approaches for circle detection that are not HT based. In [26], a size invariant accumulator based method is proposed that mainly depends on detecting the edge pixel pairs at which the gradients are just in the opposite direction of each other and grouping the mid points of these pixel pairs. Li and Wong proposed a method for detecting the object center where the operations are kept in spatial domain by use of the object area and first moment [27]. Similarly, Ho and Chen proposed an approach which utilizes the geometrical symmetry of the circles and ellipses, [28], where the pixels are classified and the circle/ellipse parameters are estimated from each group of pixels. However, these methods are examined only on synthetic data or on a limited type of data. Besides, the performances of the methods are not discussed based on the number of circular structures in the scene, i.e. the number of edge pixels which is an important factor in computational complexity.

Randomized circle detection (RCD) algorithm proposed in [29] is a non accumulator-based approach and reduces the search range after detecting each circle. In this method, four edge pixels are selected randomly and checked whether these pixels can form a circle or not. Experimental results show that up to a scene complexity, RCD is more efficient than RHT. In [30], a look-up table based method is proposed to increase the computational performance of RCD and in [31] an approach to reduce the localization error is presented. Recently, Chung *et. al.* proposed two strategies for RCD to effectively determine the candidate points [32]. Moreover, a refinement strategy is proposed to make the method more accurate for center and radius estimations. However, this method still requires high computation, especially for complicated scenes.

Optimization techniques are also used for detection of the circles. One of the first attempts are presented by Kawaguchi and Nagata for detection of ellipses using Genetic Algorithm [33] to find highest fits of the different combinations of line support regions to the image. Ayala-Ramirez *et. al.* also utilized this optimization technique to detect circles [34]. In this technique, the encoding of the three edge points are selected as the chromosome of a candidate circle and a fitness function is evaluated to detect the existence of the circle. The encoding in this study is used for reducing the search space to obtain a fast approach. In [35-37] evolutionary optimization techniques, Artificial Bee Colony (ABC) algorithm and adaptive bacterial foraging optimization (ABFO) are used to detect circles. Despite the results of the approaches are accurate, the methods are lack of speed because of their requirement of having multiple runs to detect multiple circles. Also the analyses are very limited about the speed and performance under noisy cases.

Some real time algorithms are also introduced in the literature [38, 39] to detect circles. Frosio and Borghese proposed likelihood based fast approach that also detects occluded circles [39]. Nevertheless, its application area is too limited since radius of the circles should be predefined. Recently, Akinlar and Topal introduced a real time circle detection algorithm [38]. This approach is based on joining the detected arc segments on the image by an edge segment detector Edge Drawing Parameter Free (EDPF) [40]. The success of the approach is presented using comparisons with HT and RCD but the performance of the approach is limited with the success of EDPF and the performance degrades for noisy cases.

In [41], an approach which uses Radon Transform for detection of circles is presented. It is possible to reconstruct an object back from a set of its Radon Transforms if the angle resolution is set properly. Peng and Rao used this property in order to detect the circles after filtering Radon Transform of image for a set of different angles [41]. Despite higher SNR is achieved with respect to Hough Transform, the method still requires a search in radius and suffers from the forward and back projections of a various number of functions.

In this study, a new, non accumulator based circle detection algorithm is proposed, which is based on the detection of the spatial properties of a circle. In order to reduce the required computation, image parts are transformed into two 1-D vectors by evaluating directional integrals and circle detection is achieved by using the similarity between the extracted features in the 1-D transformed vectors. This method is compared by two common approaches in the literature, MHT and RCD hence the details of these methods are explained and then the proposed method is presented in the following chapters. Finally, the performances of all these three approaches are given in Chapter 2.4.

2.1 Modified Hough Transform (MHT)

In the original circular HT, initially, intensity gradient of the image is computed and thresholded to determine the positions of significant edges, [9]. Then the positions of all possible center locations – namely, all points a distance r (radius of the anticipated circle) away from every edge pixel – are accumulated. The accumulation results are stored according to the center locations to form a two dimensional histogram. Finally, the formed data is searched for local peaks that correspond to the centers of circular objects. Since edges have nonzero width and noise will always interfere with the process peak location, accurate center location requires the use of suitable averaging procedures.

In MHT,[18],in order to decrease the computational load, it is proposed to accumulate only the points in the gradient direction of the edges. Using the Sobel operation mask, two intensity gradient values g_x and g_y are obtained by which the magnitude and orientation of the edge gradient is obtained as follows:

$$g = \sqrt{g_x^2 + g_y^2} \quad (1)$$

$$\theta = \tan^{-1}\left(\frac{g_y}{g_x}\right) \quad (2)$$

The positive gradient means that the intensity is decreasing in the direction of the gradient vector but it is not known whether the circle is dark and the background is light or vice versa. Hence, the accumulation is performed for the estimated center location both on the same direction and the opposite direction of the gradient vector. In other words, the candidate center locations are evaluated by:

$$x_c = x \pm r \cos \theta \quad (3)$$

$$y_c = y \pm r \sin \theta \quad (4)$$

where (x, y) is the location of the edge and (x_c, y_c) is the location of the probable circle with radius r . To sum up, in the MHT, only two center locations are accumulated for each edge point. Finally, local peaks of the accumulator data are detected as the center locations. Since the radius r is usually unknown, the whole procedure is repeated for a set of radius values $(r_{\min}, r_{\min}+r_{\text{step}}, \dots, r_{\max}-r_{\text{step}}, r_{\max})$. Then, the MHT procedure can be summarized as in Table 1.

Table 1 MHT Algorithm

1.	Extract edge map $E[m,n]$, where $E[m,n] \in \{0,1\}$, and gradient vectors, $g_x[m,n]$ and $g_y[m,n]$, of the image $I[m,n]$.
2.	Set $r = r_{min}$
2.1.	Evaluate the possible center locations for each edge pixel on the image using Equations (3) and (4).
2.2.	Increase related values of the accumulator correspond to those center locations are increased by one.
3.	Detect the local peaks of the accumulator.
4.	Find the peaks whose values are greater than a threshold τ and find related circles' parameters from peaks' locations.

2.2 Randomized Circle Detector (RCD)

RCD algorithm proposed in [29] is more efficient than HT in terms of memory and processing power requirements. The main reason is that RCD is not an accumulator-based approach and reduces the search range after detecting each circle. This method selects four edge pixels in the image randomly and checks whether these pixels can constitute a circle. The possible circles are determined by solving the circle equation given in Equation (5) using the three of the four randomly selected edge pixels (x_i, y_i) where $i=1..4$,

$$(x_i - x_c)^2 + (y_i - y_c)^2 = r^2 \quad (5)$$

where (x_c, y_c) is the center and r is the radius of the circle. The solution of the center coordinates and radius of the circle is found as given in Equations (6)-(8).

$$x_c = \frac{\begin{vmatrix} (x_2^2 + y_2^2) - (x_1^2 + y_1^2) & 2(y_2 - y_1) \\ (x_3^2 + y_3^2) - (x_1^2 + y_1^2) & 2(y_3 - y_1) \end{vmatrix}}{4((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1))} \quad (6)$$

$$y_c = \frac{\begin{vmatrix} 2(x_2 - x_1) & (x_2^2 + y_2^2) - (x_1^2 + y_1^2) \\ 2(x_3 - x_1) & (x_3^2 + y_3^2) - (x_1^2 + y_1^2) \end{vmatrix}}{4((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1))} \quad (7)$$

$$r = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}, \quad i \in \{1, 2, 3\} \quad (8)$$

Note that, in [29], it is stated that the distance between these three points should be less than a threshold, T_a , since otherwise evaluated center points may have very large values.

After extracting the circle parameters, the distance of the fourth point to this circle, d_4 , is evaluated using the following equation and the existence of a probable circle with these parameters is decided if this distance is smaller than the predetermined threshold, T_d .

$$d_j = \left| r - \sqrt{(x_j - x_c)^2 + (y_j - y_c)^2} \right| \quad (9)$$

Finally, the edge image is examined to determine whether the probable circle really exists by counting the number of edge pixels whose distance to the circle, evaluated by Equation(9), is smaller than T_d . In case that the obtained number exceeds another threshold, $2\pi r T_r$, this circle is considered as a valid circle. Once a circle is detected, the corresponding edge pixels are removed from the search image and the operations are repeated until no circle is found. Note that, two parameters, T_{min} and T_f are utilized in this method in order to decide the termination of operation if the number of edges is less than T_{min} or number of iterations exceeds T_f without detection of a circle.

2.3 Circle Detection by Directional Integrations (CDDI)

In this study, we propose a new circle detection algorithm based on angular profiles, i.e. directional integrals. The main idea behind this method is reducing the search space for circle parameters in order to reduce the complexity of the algorithm by evaluating limited number of directional integrals. It has already been shown that the angular profiles of circular objects have their own characteristics [41], so they are used for detection of circles by evaluating Radon Transform through many directions. In this thesis, different than these approaches, detecting a circle by use of a set of features evaluated in each angular profile and similarities of observed features is introduced.

Radon Transform can be expressed as:

$$R_\theta(\rho) = \iint f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy \quad (10)$$

where R_θ is the Radon Transform of function $f(x, y)$ at angle θ and δ is the Dirac function. When $\theta = 0$ and $\theta = \pi/2$, Equation (10) reduces to:

$$R_0(\rho) = \iint f(x, y) \delta(x - \rho) dx dy = \int f(\rho, y) dy \quad (11)$$

$$R_{\pi/2}(\rho) = \iint f(x, y) \delta(y - \rho) dx dy = \int f(x, \rho) dx \quad (12)$$

Assuming there is only one circle f in the image, it can be expressed as:

$$f(x, y) = \begin{cases} \delta(x, y), & \text{if } (x - x_c)^2 + (y - y_c)^2 = r^2 \\ 0, & \text{else} \end{cases} \quad (13)$$

where (x_c, y_c) is the center and r is the radius of that circle.

For any curve defined in 2-D plane, the arc length, l_{arc} , can be evaluated by integrating infinitely small parts, i.e. ds (see Figure 3), as in Equation (14).

$$l_{arc} = \int_{\{x, y | f(x, y)=1\}} ds \quad \text{where } ds = \sqrt{(dx)^2 + (dy)^2} \quad (14)$$

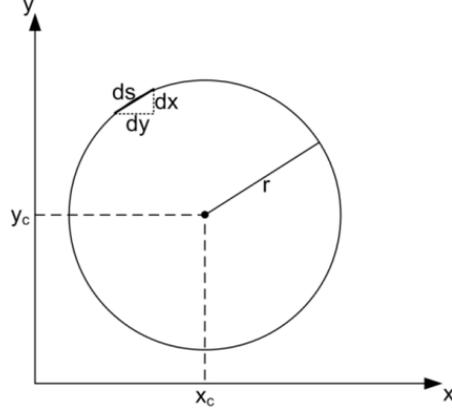


Figure 3 Differential elements on the circle.

Since I_{arc} is equal to the circumference of the circle, it can also be expressed as:

$$l_{arc} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy \quad (15)$$

Let $g_I(x)$ be the integration of the circle along y-axis which is expressed as

$$g_1(x) = \int_{-\infty}^{\infty} f(x, y) dy \quad (16)$$

Substituting $g_I(x)$ into Equation(15), we have

$$l_{arc} = \int_{-\infty}^{\infty} g_1(x) dx \quad (17)$$

Taking the derivative of s with respect to x , $g_I(x)$ can be found as:

$$\begin{aligned} g_1(x) &= \frac{ds}{dx} \\ &= \sqrt{1 + \left(\frac{dy}{dx}\right)^2} \end{aligned} \quad (18)$$

Using Equation (5), $g_I(x)$ becomes

$$\begin{aligned} g_1(x) &= 2 \sqrt{1 + \left(\frac{-2(x-x_c)}{2(r^2 - (x-x_c)^2)^{\frac{1}{2}}} \right)^2} \text{rect}\left(\frac{x-x_c}{2r}\right) \\ &= 2 \frac{1}{\sqrt{1 - \frac{(x-x_c)^2}{r^2}}} \text{rect}\left(\frac{-x+x_c}{2r}\right) \end{aligned} \quad (19)$$

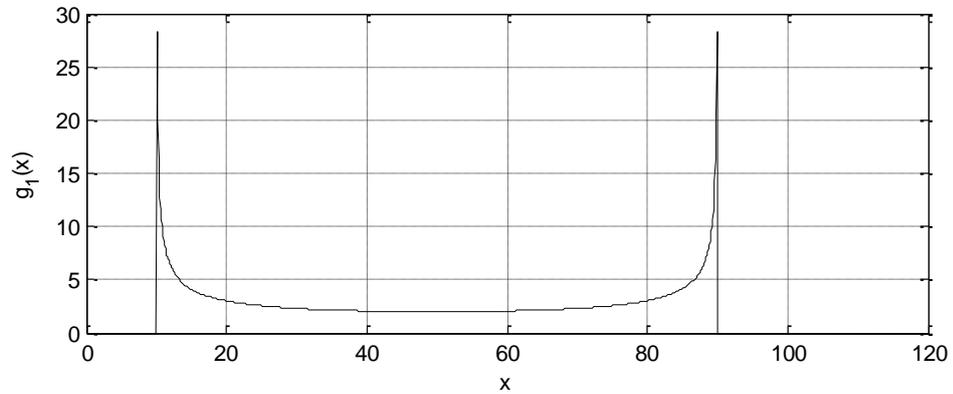
where $rect(t)$ is rectangle function defined as below:

$$rect(t) = \begin{cases} 1, & \text{if } |t| \leq \frac{1}{2} \\ 0, & \text{else} \end{cases} \quad (20)$$

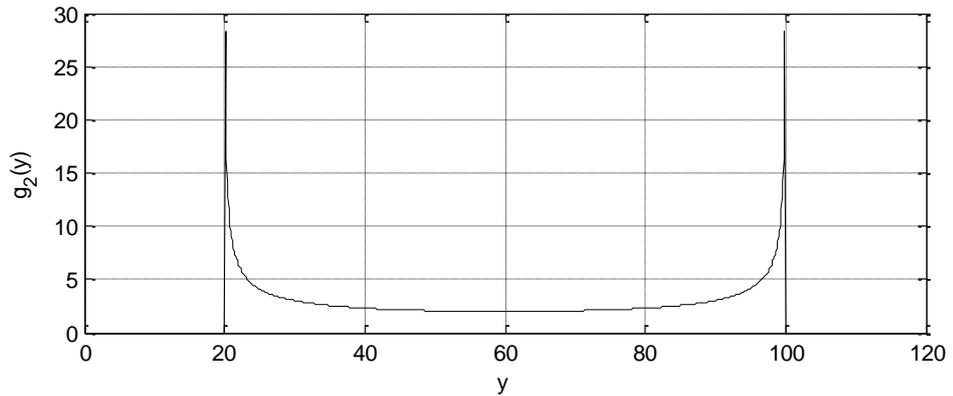
Using the same approach, the integration of the circle along x-axis, $g_2(y)$, can be found as:

$$g_2(y) = 2 \frac{1}{\sqrt{1 - \frac{(y - y_c)^2}{r^2}}} rect\left(\frac{y - y_c}{2r}\right) \quad (21)$$

In Figure 4, $g_1(x)$ and $g_2(y)$ are plotted for a circle of radius 40, where its center is at (50, 60). From this figure and Equations (19) and (21), it can be easily seen that $g_1(x)$ is minimum at $x=x_c$ and has asymptotes at $x=x_c-r$ and $x=x_c+r$; $g_2(y)$ is minimum at $y=y_c$ and has asymptotes at $y=y_c-r$ and $y=y_c+r$.



(a)



(b)

Figure 4 (a) $g_1(x)$ and (b) $g_2(y)$ for a circle of radius 40 centered at (50, 60). Both x and y axes are sampled by 0.1 units.

2.3.1 Analysis at Discrete Domain

For a discrete circle $f[m,n]$, $g_1[m]$ and $g_2[n]$ can be defined as follows:

$$\begin{aligned}
g_1[m] &= \sum_n f[m,n] \\
g_2[n] &= \sum_m f[m,n]
\end{aligned} \tag{22}$$

Then,

$$\begin{aligned}
g_1[m] &= \int_{m-0.5}^{m+0.5} 2 \frac{\text{rect}\left(\frac{x-x_c}{2r}\right)}{\sqrt{1-\left(\frac{x-x_c}{r}\right)^2}} dx & m \in \mathbf{Z} \\
g_2[n] &= \int_{n-0.5}^{n+0.5} 2 \frac{\text{rect}\left(\frac{y-y_c}{2r}\right)}{\sqrt{1-\left(\frac{y-y_c}{r}\right)^2}} dy & n \in \mathbf{Z}
\end{aligned} \tag{23}$$

Assuming x_c and r be integers, the minimum point of $g_1[m]$ is at $m=x_c$, where maximum points of $g_1[m]$ are at $m=x_c-r$ and $m=x_c+r$. When $m=x_c$:

$$\begin{aligned}
g_1[x_c] &= 2r \int_{-\frac{0.5}{r}}^{\frac{0.5}{r}} \frac{1}{\sqrt{1-a^2}} da \\
&= 2r \sin^{-1}(a) \Big|_{-0.5/r}^{0.5/r} \\
&= 4r\alpha
\end{aligned} \tag{24}$$

where

$$\begin{aligned}
\alpha &= \sin^{-1}\left(\frac{1}{2r}\right) \\
&\approx \frac{1}{2r} \quad \text{as } r \rightarrow \infty
\end{aligned} \tag{25}$$

Hence,

$$g_1[x_c] \approx 2 \quad \text{as } r \rightarrow \infty \tag{26}$$

When $m=x_c+r$:

$$\begin{aligned}
g_1[x_c+r] &= \int_{x_c+r-0.5}^{x_c+r+0.5} 2 \frac{\text{rect}\left(\frac{x-x_c}{2r}\right)}{\sqrt{1-\left(\frac{x-x_c}{r}\right)^2}} dx \\
&= \int_{x_c+r-0.5}^{x_c+r} 2 \frac{1}{\sqrt{1-\left(\frac{x-x_c}{r}\right)^2}} dx
\end{aligned} \tag{27}$$

By change of variable $a = (-x + x_c)/r$

$$\begin{aligned}
g_1[x_c + r] &= 2r \int_{\frac{r-0.5}{r}}^1 \frac{1}{\sqrt{1-a^2}} da \\
&= 2r \sin^{-1}(a) \Big|_{(r-0.5)/r}^1 \\
&= 2r \left[\frac{\pi}{2} - \sin^{-1}\left(\frac{r-0.5}{r}\right) \right] \\
&= 2r \cos^{-1}\left(\frac{r-0.5}{r}\right)
\end{aligned} \tag{28}$$

Small angle approximation indicates that:

$$\cos(\theta) \approx 1 - \frac{\theta^2}{2} \text{ as } \theta \rightarrow 0 \tag{29}$$

Then,

$$\begin{aligned}
1 - \frac{\alpha^2}{2} &\approx \frac{r-0.5}{r} \\
\Rightarrow \alpha &\approx \frac{1}{\sqrt{r}} \text{ as } r \rightarrow \infty
\end{aligned} \tag{30}$$

Hence,

$$g_1[x_c + r] \approx 2\sqrt{r} \text{ as } r \rightarrow \infty \tag{31}$$

Similarly, when $m = x_c - r$

$$g_1[x_c - r] \approx 2\sqrt{r} \text{ as } r \rightarrow \infty \tag{32}$$

Similarly, for $g_2[n]$, the extremes can be approximated as:

$$\begin{aligned}
g_2[y_c] &\approx 2 \\
g_2[y_c + r] &\approx 2\sqrt{r} \\
g_2[y_c - r] &\approx 2\sqrt{r}
\end{aligned} \tag{33}$$

A synthetically generated circle can be seen in Figure 5 (a), where the image is 120x120, the circle is centered at (50, 60) with a radius of 40. When the vertical and horizontal profiles are evaluated on this image resultant $g_1[m]$ and $g_2[n]$ can be seen in Figure 5 (b) and (c), respectively. As seen from these graphs, these two graphs are identical but shifted from the origin according to the x and y axis components of the circle's center. For both $g_1[m]$ and $g_2[n]$, the distances between maxima are 80 (diameter of the circle) units and their values are equal to 13 ($2\sqrt{r} = 12.6491$). These functions have values of 2 at $m=50$ and $n=60$, respectively but there are other locations having a value of 2 due to the discretization. These results prove the validity of the analytical solutions given in Equations(26), (31)-(33).

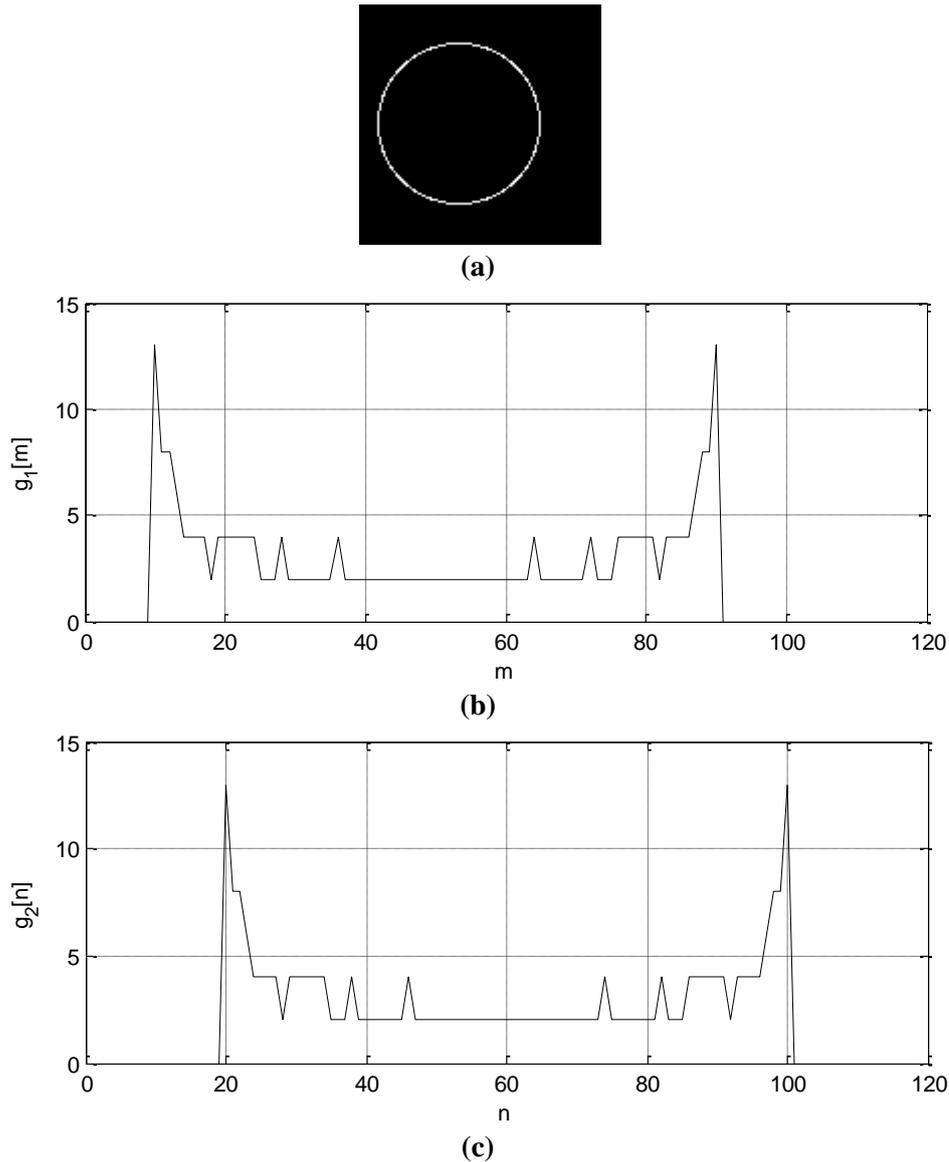


Figure 5 (a) A synthetic circle image of radius 40 centered at (50, 60) and evaluated directional summations (b) $g_1[m]$ and (c) $g_2[n]$ from this image.

The profiles are also calculated for some circular structures on the real images and the outcomes are presented in Figure 6 and Figure 7. In Figure 6 (a), an image contains of some coins are presented and its edge map is given in Figure 6 (b). The circle given in Figure 6 (c) is extracted manually from this edge map to evaluate the horizontal and vertical profiles, $g_1[m]$ and $g_2[n]$, as given in Figure 6 (e) and (f), respectively. When $g_1[m]$ is examined, its local peaks are found at $m=27$ and $m=85$. This result yields the x component of its center, x_c , to be at $m=56$ and radius to be 29. These peak values are 14 and 13, which are also comparable with the expectation $2\sqrt{r} = 10.7703$. When $g_2[n]$ is examined a similar case is observed. The peaks are found at $n=43$ and $n=99$, which yields the y component of its center, y_c , to be at 71 and the radius to be 28. The peak values are 14 and 12 and they are comparable with the theoretical expectation $2\sqrt{r} = 10.5830$. This specific result shows that one can estimate the center point and the radius of the circle from the locations of the

peaks in $g_1[m]$ and $g_2[n]$, once the peak values are similar and they are also similar with the theoretical expectation given in Equations (31)-(33). When x_c is taken from $g_1[m]$ observations, y_c is estimated from $g_2[n]$ observations and r is estimated by taking the average of both observations a circle with radius of 28.5 centered at (56, 71) is found, which is drawn on the original image in Figure 6 (d).

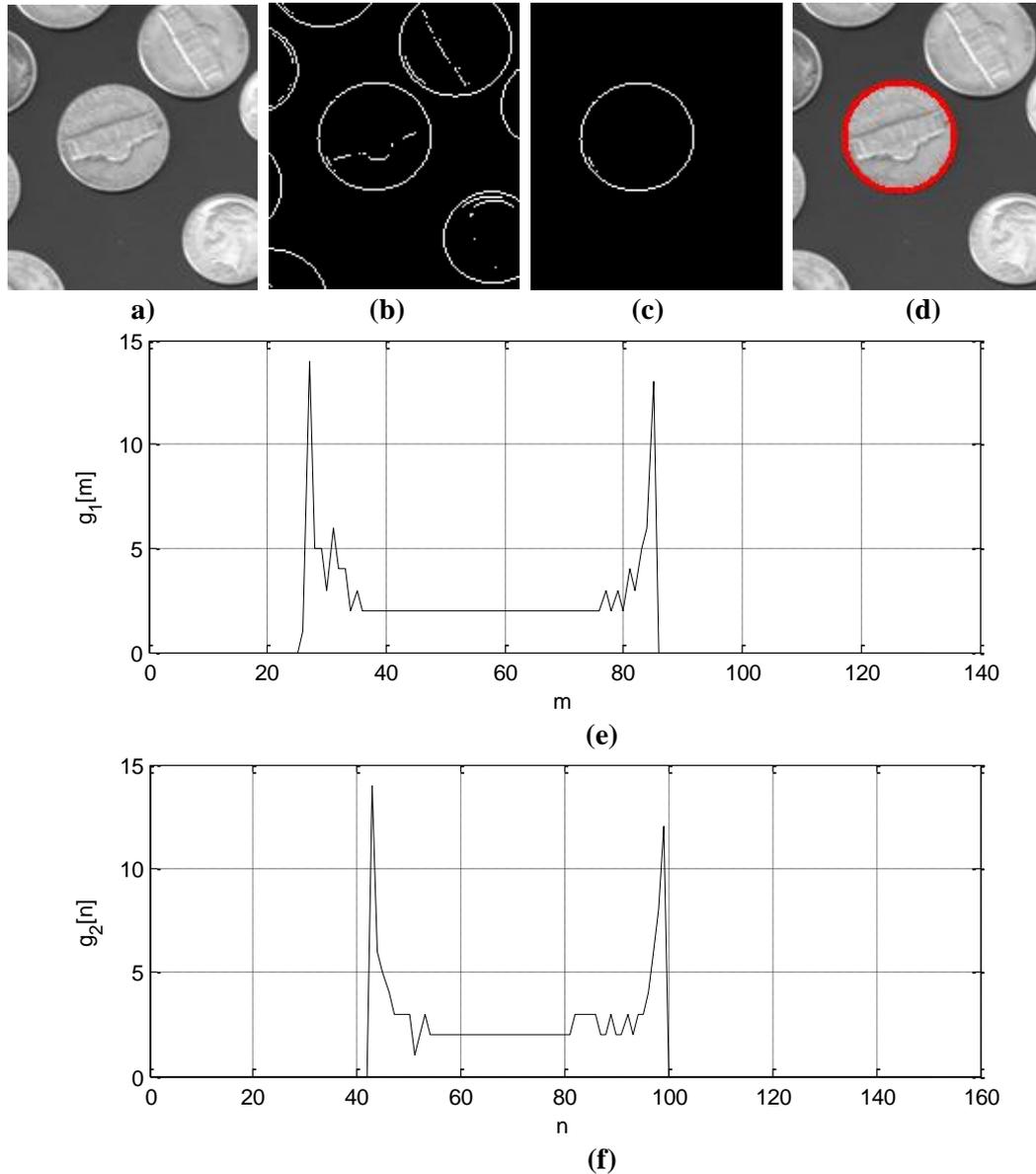


Figure 6 (a) A sample image containing coins, (b) its edge map using Canny detector, (c) manually extracted circular structure from the edge map (d) estimated circle from (e) vertical profile, $g_1[m]$, and (f) horizontal profile $g_2[n]$.

An image of a POL, given in Figure 7 (a), is also examined for checking the consistency of the findings. The edge map and the manually extracted circular edge are given in Figure 7 (b) and (c), respectively. When the profiles given in Figure 7 (e) and (f) are analyzed as

explained in the previous paragraph, a circle with a radius of 22.5 centered at (52.5, 52.5) can be estimated. The estimated circle is drawn on the original image in Figure 7 (d).

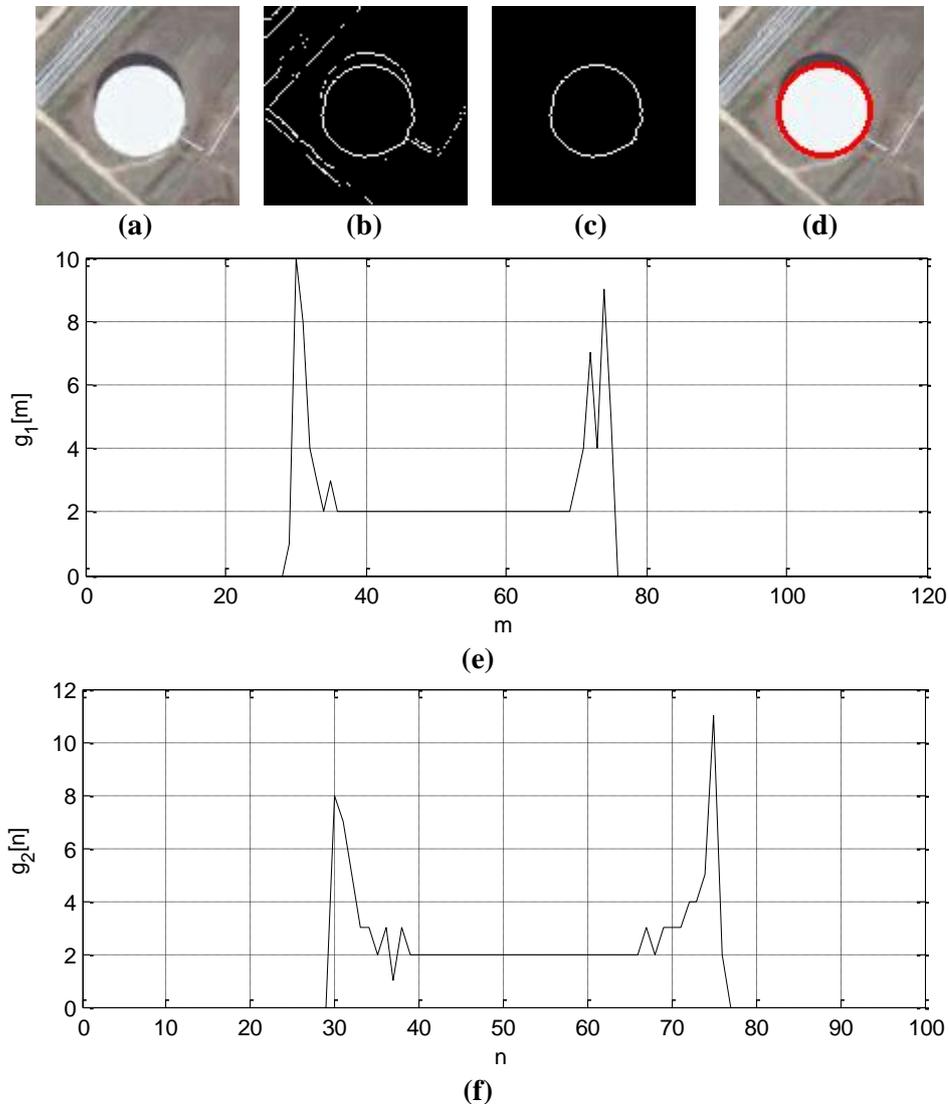


Figure 7 (a) A sample POL image, (b) its edge map using Canny detector, (c) manually extracted circular structure from the edge map (d) estimated circle from (e) vertical profile, $g_1[m]$, and (f) horizontal profile $g_2[n]$.

2.3.2 Proposed Circle Detection Algorithm

In the previous section, it is shown that information about the parameters of a circle can be extracted from its horizontal and vertical profiles. According to the obtained results, there has to be two peaks in both of the profiles, and these peak pairs in each profile have to possess similar characteristics. Therefore, in this study, the problem of detecting circles is reduced to identifying matching peak pairs in the horizontal and vertical profiles of the circle.

Equations (31)-(33) imply existence of only two peaks in each profile of the circle, which are also identical to each other. However, discretization, distorted shapes or noise on the image may cause other peaks to appear and/or desired peaks to be different from each other. So, a two step process is proposed to detect the matching peaks in the profiles. The first step is finding the global maxima of $g_1[m]$ and $g_2[n]$, $g_{1,max}$ and $g_{2,max}$, respectively, by:

$$\begin{aligned} m_{\max} &= \arg \max_m \{g_1[m]\} \\ g_{1,\max} &= g_1[m_{\max}] \end{aligned} \quad (34)$$

$$\begin{aligned} n_{\max} &= \arg \max_{n_{\max}} \{g_2[n]\} \\ g_{2,\max} &= g_2[n_{\max}] \end{aligned} \quad (35)$$

In the second step, features related to each local peak, m_i and n_j , in $g_1[m]$ and $g_2[n]$, given in Equations (36)-(41) are examined in order to detect the matching peak.

$$p_{1,i} = g_1[m_i] \quad (36)$$

$$d_{1,i} = |m_i - m_{\max}| \quad (37)$$

$$\mu_{1,i} = \frac{1}{d_{1,i}} \sum_{m=\min\{m_i, m_{\max}\}}^{\max\{m_i, m_{\max}\}} g_1[m] \quad (38)$$

$$p_{2,j} = g_2[n_j] \quad (39)$$

$$d_{2,j} = |n_j - n_{\max}| \quad (40)$$

$$\mu_{2,j} = \frac{1}{d_{2,j}} \sum_{n=\min\{n_j, n_{\max}\}}^{\max\{n_j, n_{\max}\}} g_2[n] \quad (41)$$

where $p_{1,i}$'s and $p_{2,j}$'s denote the values of local peaks of $g_1[m]$ and $g_2[n]$; $d_{1,i}$'s and $d_{2,j}$'s denote the distances between the local peaks and global maxima of $g_1[m]$ and $g_2[n]$; $\mu_{1,i}$'s and $\mu_{2,j}$'s denote the average values of $g_1[m]$ and $g_2[n]$ in between their global maxima and peaks.

Based on Equations (31)-(33) if there is a circle in the image, for at least one of the local peak in each profile, the following conditions should be satisfied:

- Peak Similarity: The value of the local peaks in $g_1[m]$ and $g_2[n]$ are close to the global maxima of that profile. Then,

$$\begin{aligned} \frac{p_{1,i}}{p_{1,\max}} &\geq \alpha \\ \frac{p_{2,j}}{p_{2,\max}} &\geq \alpha \quad \text{where } \alpha \in (0,1] \end{aligned} \quad (42)$$

- Peak-To-Radius Consistency: The peak values of the profile are close to $2\sqrt{r}$ and the distances between the desired peaks in each profile are close to $2r$. Since r is not known, the ratio of these two constraints should satisfy

$$\beta \leq \frac{g_{1,\max}}{\sqrt{2d_{1,i}}} \leq \frac{1}{\beta}$$

$$\beta \leq \frac{g_{2,\max}}{\sqrt{2d_{2,j}}} \leq \frac{1}{\beta}, \quad \text{where } \beta \in (0,1]$$
(43)

- Axis Similarity: Due to the symmetry of the circle, the peak values of both profiles are close to each other. Then,

$$\gamma \leq \frac{g_{1,\max}}{g_{2,\max}} \leq \frac{1}{\gamma}, \quad \text{where } \gamma \in (0,1]$$
(44)

- Radius Uniqueness: The estimated radii from $g_1[m]$ and $g_2[n]$ are similar. Then,

$$\lambda \leq \frac{d_{1,i}}{d_{2,j}} \leq \frac{1}{\lambda}, \quad \text{where } \lambda \in (0,1]$$
(45)

- Mean Validity: Since the average values of $g_1[m]$ and $g_2[n]$ between the matching peak pairs should be close to π , $\mu_{1,i}$ and $\mu_{2,j}$ should satisfy:

$$\mu_{\min} \leq \frac{1}{\pi} \mu_{1,i} \leq \mu_{\max}$$

$$\mu_{\min} \leq \frac{1}{\pi} \mu_{2,j} \leq \mu_{\max}, \quad \text{where } \mu_{\min} < 1 \text{ and } \mu_{\max} > 1$$
(46)

Once the local peaks satisfying all the conditions above, m_i and n_j , are detected, the parameters of the circle can be estimated by:

$$x_c = \frac{m_{\max} + m_i}{2}$$

$$y_c = \frac{n_{\max} + n_j}{2}$$

$$r = \frac{d_{1,i} + d_{2,j}}{4}$$
(47)

On the other hand, due to the missing edges, it is possible to detect only the matching peak pair in one of the profiles. For this case, we propose to use the radius estimate in that profile and derive the coordinate of the center in the other profile, accordingly. In other words, if the condition for Axis Similarity is satisfied and the local peak of $g_1[m], m_i$, satisfying the Peak Similarity, Peak-to-Radius Consistency and Mean Validity conditions is detected but such a local peak cannot be found in $g_2[n]$ the parameters of the circle is estimated by Equation (52). Similarly, when n_j is detected but m_i , is not, and the Axis Similarity condition is satisfied, the parameters of the circle are estimated by Equation (53).

Table 2 CDDI circle detection procedure.

1. Extract edge map $E[m,n]$ of the image $I[m,n]$, where $E[m,n] \in \{0,1\}$.
2. Set $E[m, n]$ to 0 for the locations having more than three edge pixels in its 8 neighborhoods as:

$$E[m,n] = 0, \text{ if } C_E[m,n] > 3$$

$$\text{where } C_E[m,n] = \sum_{k \in \{-1,1\}} \sum_{l \in \{-1,1\}} E[m+k, n+l] \quad (48)$$

3. Extract connected components, c_k 's, in $E[m,n]$.
4. Obtain the objects (or blobs), b_k 's, satisfying the following conditions:
 - It should be composed of at most n_n connected components
 - If there are more than one connected component in the object, for each of c_u there should be another connected component c_v having a distance at most n_p pixels, where the distances of the connected components are defined as below:

$$\|c_u - c_v\| = \min \left(\sqrt{|m_u - m_v|^2 + |n_u - n_v|^2} \right) \quad (49)$$

$$\text{where } (m_u, n_u) \in c_u \quad (m_v, n_v) \in c_v$$

5. For each b_k
 - 5.1. Compute $g_1[m]$ and $g_2[n]$ as in Equation (22).
 - 5.2. Detect the global maximum points in $g_1[m]$ (m_{max}) and $g_2[n]$ (n_{max}).
 - 5.3. Go back to Step 5 if Axis Similarity condition is not satisfied.
 - 5.4. Detect the local maximum points in $g_1[m]$ (m_i 's) and $g_2[n]$ (n_j 's).
 - 5.5. Solve Equations (50) and (51) to find the matching peaks on the profiles that satisfy the Peak Similarity, Peak-To-Radius Consistency, Mean Validity conditions to detect the largest possible circle.

$$i_r = \arg \max_i \{|m_{max} - m_i|\} \quad (50)$$

$$j_r = \arg \max_j \{|n_{max} - n_j|\} \quad (51)$$

- 5.6. A circle is detected if
 - i_r and j_r are found which satisfy the Radius Uniqueness condition; and the parameters of the circle are estimated using Equation (47)
 - only i_r is found; and the parameters of the circle are estimated by Equation (52)
 - only j_r is found; and the parameters of the circle are estimated by Equation (53)

$$\begin{aligned}
x_c &= \frac{m_{\max} + m_{i_r}}{2} \\
r &= \frac{d_{1,i_r}}{2}
\end{aligned} \tag{52}$$

$$\begin{aligned}
y_c &= n_{\max} + \begin{cases} r, & \text{if } g_2[n_{\max} + 1] > g_2[n_{\max} - 1] \\ -r, & \text{otherwise} \end{cases} \\
y_c &= \frac{n_{\max} + n_{j_r}}{2} \\
r &= \frac{d_{2,j_r}}{2} \\
x_c &= m_{\max} + \begin{cases} r, & \text{if } g_1[m_{\max} + 1] > g_1[m_{\max} - 1] \\ -r, & \text{otherwise} \end{cases}
\end{aligned} \tag{53}$$

Note that, since many peak pairs may appear that satisfy these conditions, we propose to get the most distant pairs in each profile.

When this problem is tried to be solved for whole scene, it becomes very costly due to the response of possible different shapes, noise and ambiguities caused by the existence of many circles. Thus, we propose to detect the circles after determination of the connected components in the image. Besides, two thresholds, n_n and n_p , are defined to process at most n_n neighbor connected components, which are at most n_p pixels away from one of the components, together. Then, the overall procedure for the proposed method is given in Table 2.

2.4 Effects of Edge Detection Approaches on CDDI

In this study, Canny edge detector is utilized for determination of the object boundaries to be used in CDDI approach. On the other hand there are many other commonly used approaches in the literature. Some of these approaches use some filters for discrete approximations of first order differentiation, i.e. Robert, Prewitt, Sobel, and the local maxima on the filter outputs are detected as edges. Some other methods use second order differentiation and searched for the zero crossings at the output, i.e., Laplacian of Gaussian (LoG). Despite producing very close results, especially in the absence of noise, the environmental disturbances may cause big differences at their output that change the performance of CDDI drastically. Before analyzing the effects of these methods on CDDI, brief descriptions of these edge detection approaches are given.

Robert's Edge Detector: Robert's Edge Detection is the simplest edge detection approach and performs a simple and quick 2-D spatial gradient measurement on an image. A pair of operator, given in Equations (54) and (55), is convolved with the gray scale image, I , for estimation of the horizontal and vertical gradients, respectively.

$$h_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{54}$$

$$h_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (55)$$

Then, the magnitudes of the gradients are evaluated by use of the horizontal and vertical gradients at each spatial coordinate as given below:

$$G_{abs}[m,n] = \sqrt{G_x^2[m,n] + G_y^2[m,n]} \quad (56)$$

where

$$\begin{aligned} G_x &= I * h_x \\ G_y &= I * h_y \end{aligned} \quad (57)$$

Finally, the local maxima on G_{abs} are detected as the edges on the image if its value is greater than a predetermined threshold.

Prewitt's Edge Detector: This edge detection method is based on the same idea with Robert's method but uses a larger kernel to estimate the gradients better as given below:

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (58)$$

$$h_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (59)$$

Sobel Edge Detector: Sobel operator also uses a pair of kernel to approximate the horizontal and vertical gradients which are given in Equations (60) and (61) and the local maxima on the magnitude of the gradients are detected as the edges if its value is greater than a predetermined threshold.

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (60)$$

$$h_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (61)$$

Laplacian of Gaussian (LoG): Laplacian is an isotropic measure for the 2nd order derivative of an image as given below:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (62)$$

Since the second derivation is very sensitive to noise, an image is generally low-pass filtered before the Laplacian operation. LoG operator is the special case of Laplacian where the image is filtered by a Gaussian filter first. These operators are convolved analytically in order to reduce the computation and the LoG operator is obtained as given in Equation (63).

$$\begin{aligned} LoG(x, y) &= L(x, y) * G(x, y) \\ &= \frac{-1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2} \right) e^{-\frac{x^2 + y^2}{2\sigma^2}} \end{aligned} \quad (63)$$

where σ is the standard deviation of the Gaussian filter. As σ increases, the smoothing increases and as the size of the used kernel increases a better approximation is obtained. Since the output of the filter is the second order derivative of the image, the zero crossings on the obtained data are detected as the edges.

Canny Edge Detector: Canny edge detector is a complex algorithm, composed of four main steps to improve the classical edge detection approaches by reducing the error rate, increasing the localization accuracy and having only one response to a single edge [42]. These steps are:

1. Noise reduction: Canny proposed to use a Gaussian smoothing filter before detection of the edges. The parameters of the filter, σ and the kernel size, are the parameters of the approach and increasing these values make the method less vulnerable to noise but less accurate.
2. Gradient estimation: The magnitude and direction of the gradients are evaluated at each pixel location by use of derivatives in horizontal and vertical directions. These gradients can be evaluated by Sobel operator.
3. Non-maxima suppression: The edges are localized by detection of the local maxima of the gradient image. The directions of the edges are also utilized for determination of these local maxima by determining the region where the controlled pixel's gradient should be greater than.
4. Thresholding: Finally each local maximum of the gradient image is thresholded using a hysteresis approach which uses two threshold values to determine whether it is an edge pixel or not. If the magnitude of the gradient at any local maxima is below the lower threshold then this pixel is not an edge pixel where it is definitely an edge pixel if it is greater than the higher threshold. If the gradient magnitude is in between these thresholds, it is set to zero unless it is on a path from one edge pixel to another where all the gradients are above the lower threshold.

As explained in these steps, Canny edge detection algorithm is a complex approach and many different constraints are considered to decide an edge pixel. Therefore, it produces smooth and thinned edge map with low localization error.

In order to analyze the effect of different edge detectors on CDDI approach, a binary circle image is created as seen in Figure 8 (a). Different amounts of Salt & Pepper noise are added to the image and the images given in Figure 8 (b)-(d) are obtained. Finally for each of these images, CDDI approach is operated to detect the circle. From Figure 9 to Figure 12 detected edges are presented using previously explained five methods and from Figure 13

to Figure 16, detected circles by CDDI approach are presented by use of each of edge detectors.

For Robert's, Prewitt and Sobel operators the threshold is determined by Equation (64).

$$\tau_{lm} = 2 \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N G_{abs}^2 [i, j]} \quad (64)$$

For LoG method the zero crossing are detected as edges if there is a large jump between the either side of 0, so another threshold is induced, which is evaluates as in Equation (65), in order to decide the edges.

$$\tau_{zc} = 0.75 \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N LoG [i, j] \quad (65)$$

Finally, for Canny edge detector the low and high threshold values are evaluated as given in below:

$$\begin{aligned} \tau_{high} &= G_{sorted} [i], & i &= 0.7 \times M \times N \\ \tau_{low} &= 0.4 * \tau_{high} \end{aligned} \quad (66)$$

where G_{sorted} is the vector composed of the ascending ordered sorted values of absolute values of the gradients, G_{abs} .

Note that, for CDDI approach the parameters are set as $\alpha=0.5$, $\beta=0.5$, $\gamma=0.5$, $\lambda=0.75$, $\mu_{min}=0.5$ and $\mu_{max}=1.5$.

As seen from these figures Robert's operator is the most sensitive approach to the noise and when $L_{Noise}=1$, it fails in detecting edges. By using Prewitt and Sobel operators, extracted edges are good enough to detect the circles up to noise level of 0.1. Since in LoG and Canny detectors, smoothing is applied first, they are more robust against to noise but Canny approach preserves the shape of the edges better by which the circle is still detectable when $L_{Noise}=1$.

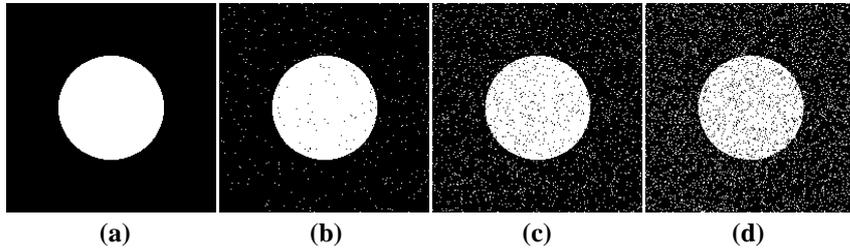


Figure 8 (a) A circle with radius 50 and its noise versions when (b) $L_{Noise}=0.1$, (c) $L_{Noise}=0.5$, (d) $L_{Noise}=1$.

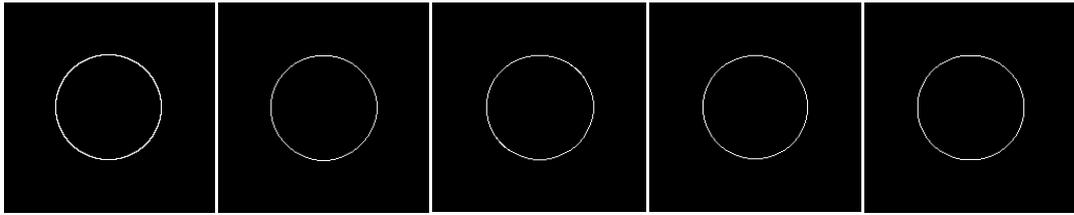


Figure 9 Edges detected by (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny methods when $L_{\text{Noise}} = 0$.

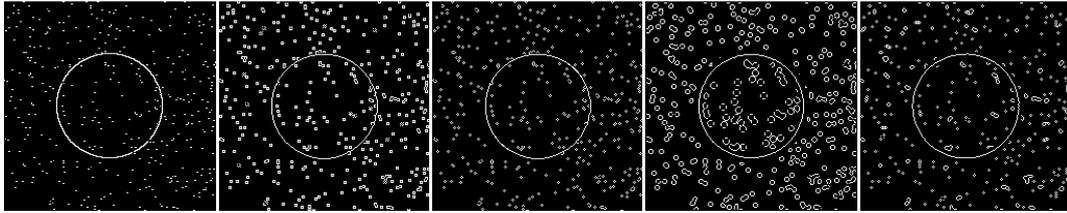


Figure 10 Edges detected by (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny methods when $L_{\text{Noise}} = 0.1$.

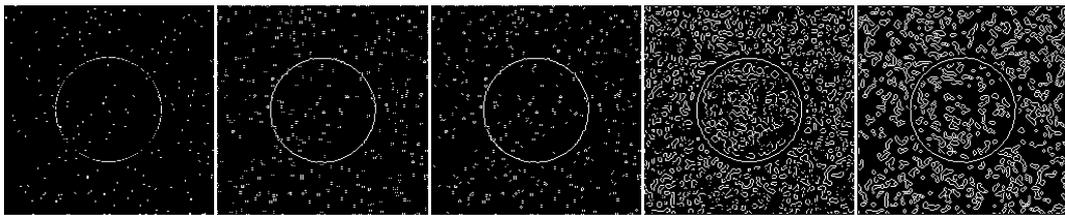


Figure 11 Edges detected by (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny methods when $L_{\text{Noise}} = 0.5$.

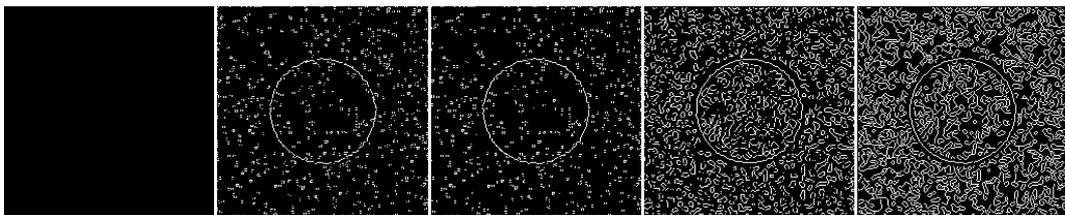


Figure 12 Edges detected by (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny methods when $L_{\text{Noise}} = 1$.

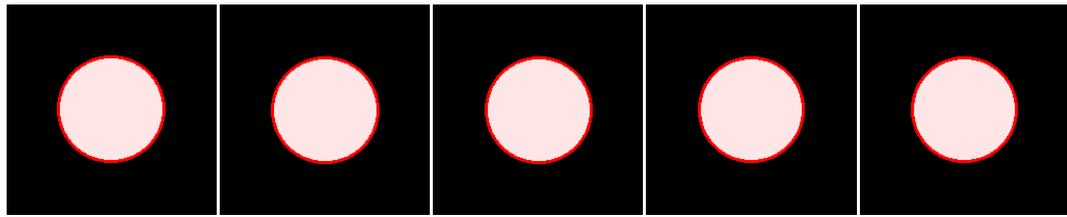


Figure 13 Detected circles by CDDI using (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny edge detectors when $L_{\text{Noise}} = 0$.

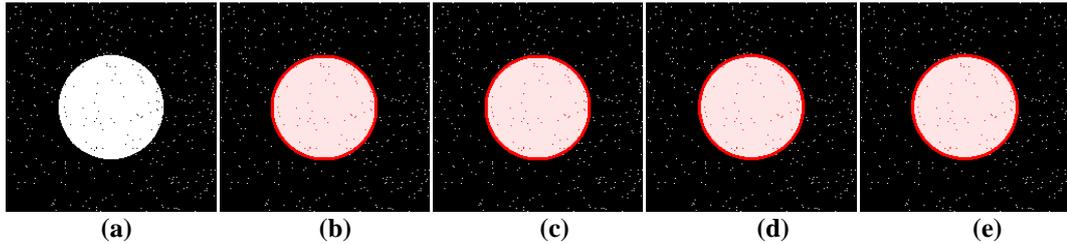


Figure 14 Detected circles by CDDI using (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny edge detectors when $L_{\text{Noise}} = 0.1$.

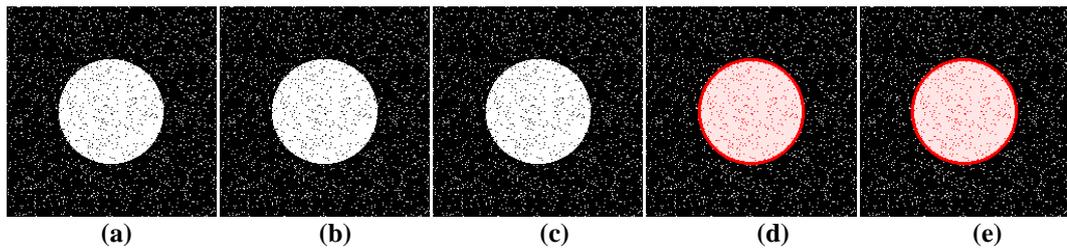


Figure 15 Detected circles by CDDI using (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny edge detectors when $L_{\text{Noise}} = 0.5$.

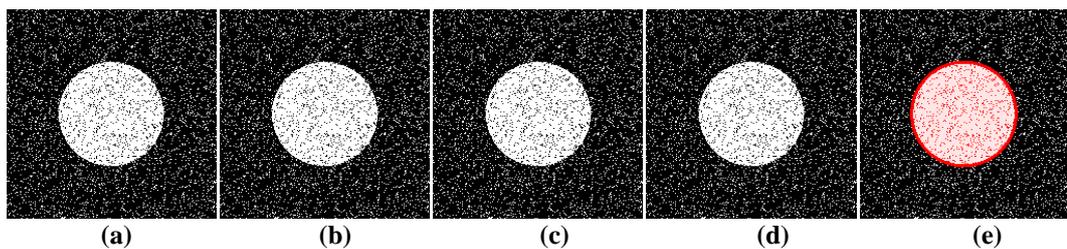


Figure 16 Detected circles by CDDI using (a) Robert's, (b) Prewitt, (c) Sobel, (d) LoG and (e) Canny edge detectors when $L_{\text{Noise}} = 1$.

The analyses are also made on some real images. In Figure 17 and Figure 18, two different POL examples, captured from Google EarthTM, can be seen. In the first example, the contrast difference between the POL and its surrounding is higher. Therefore all of the edge detectors produce similar results and CDDI can detect the POL precisely located by use of all of the edge detectors. For the second POL example, the scene is more complex and the

detectors produce different edge maps. By Robert's, Prewitt and Sobel operators, there are mainly three half circles on the edge map so CDDI cannot detect the POL with the given parameter set. By LoG and Canny approaches more detailed edge maps are obtained but Canny forces the edge to be continuous hence at some points of the POL (especially for its lower-left part) the edges can be detected even the contrast difference between the POL and its surrounding is as high as the other parts of the POL. Therefore, CDDI can only detect the POL by use of Canny edge detector.

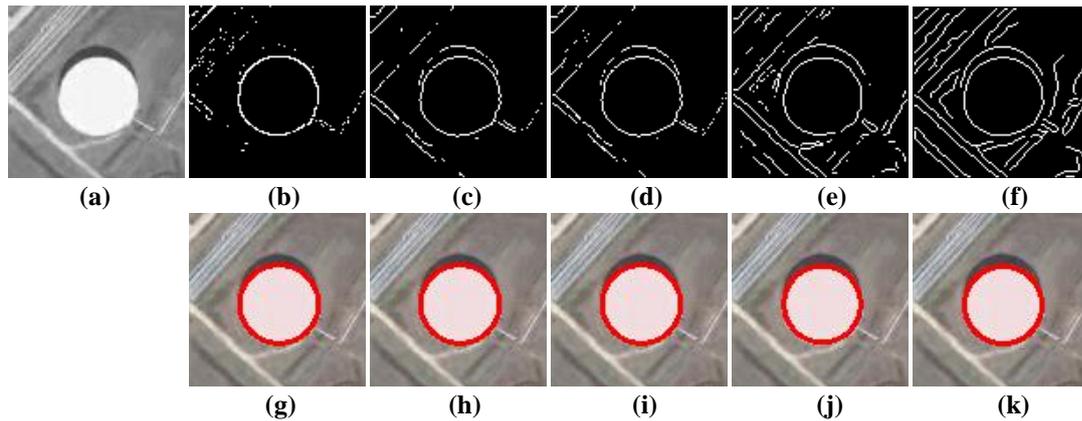


Figure 17 (a) A satellite image of a POL with its edge maps and detected circles by CDDI using (b), (g) Robert's; (c), (h) Prewitt; (d), (i) Sobel; (e), (j) LoG; (f), (k) Canny method.

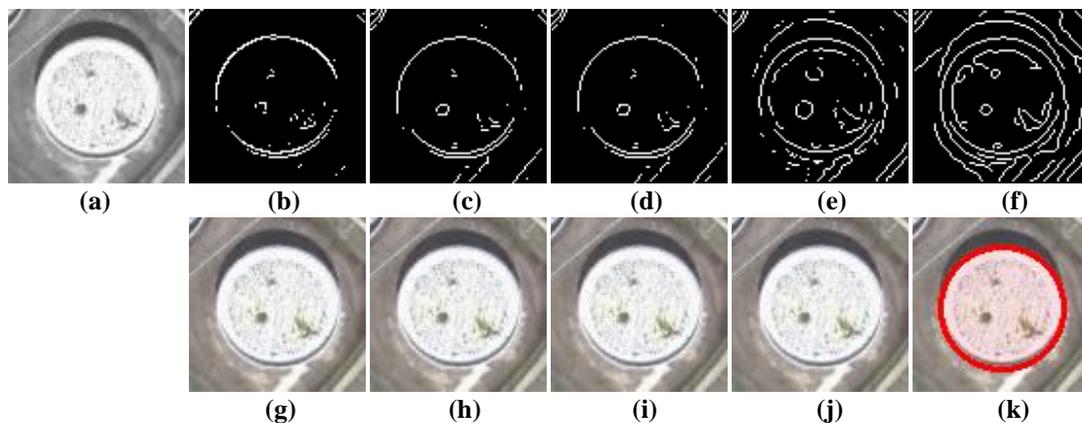


Figure 18 (a) A satellite image of a POL with its edge maps and detected circles by CDDI using (b), (g) Robert's; (c), (h) Prewitt; (d), (i) Sobel; (e), (j) LoG; (f), (k) Canny method.

To sum up, upon the commonly used edge detectors Canny edge detector is the best choice to utilize with CDDI method because of two main reasons. The first one is that Canny detector is less vulnerable to noise. Secondly, CDDI operates on the connected parts of the edge map and they are forced to be continuous and smooth by Canny detector.

2.5 Experimental Results

In this section, the results obtained by the proposed method are presented for synthetic and real world data. Moreover, the comparisons of the proposed method with the MHT and RCD algorithms are given in order to see the advantages and disadvantages of the method. All the methods are implemented in MATLAB R2009b and the experiments are done using a computer with Intel Core 2 Duo 2.4 GHz processor and 3 GB of RAM.

In order to test the robustness of the method against noise, a 256×256 image is created with 6 randomly located circles with randomly selected radii between 10 and 50. Then, ‘‘Salt & Pepper’’ type noise is added to the image at different levels where the noise level is evaluated using the formula given in [29] as:

$$L_{Noise} = \frac{\#of\ noise\ added\ pixels}{\#of\ all\ circles'\ pixels} \quad (67)$$

The experiments show that the method can detect circles up to $L_{Noise}=1$. An example of the test image, its edge map and the detected circles for this case are given in Figure 19 when noise level is 1. For this case, $\alpha=0.5$, $\beta=0.5$, $\gamma=0.5$, $\lambda=0.75$, $\mu_{min}=0.5$ and $\mu_{max}=1.5$, $n_n=0$, $n_p=0$.

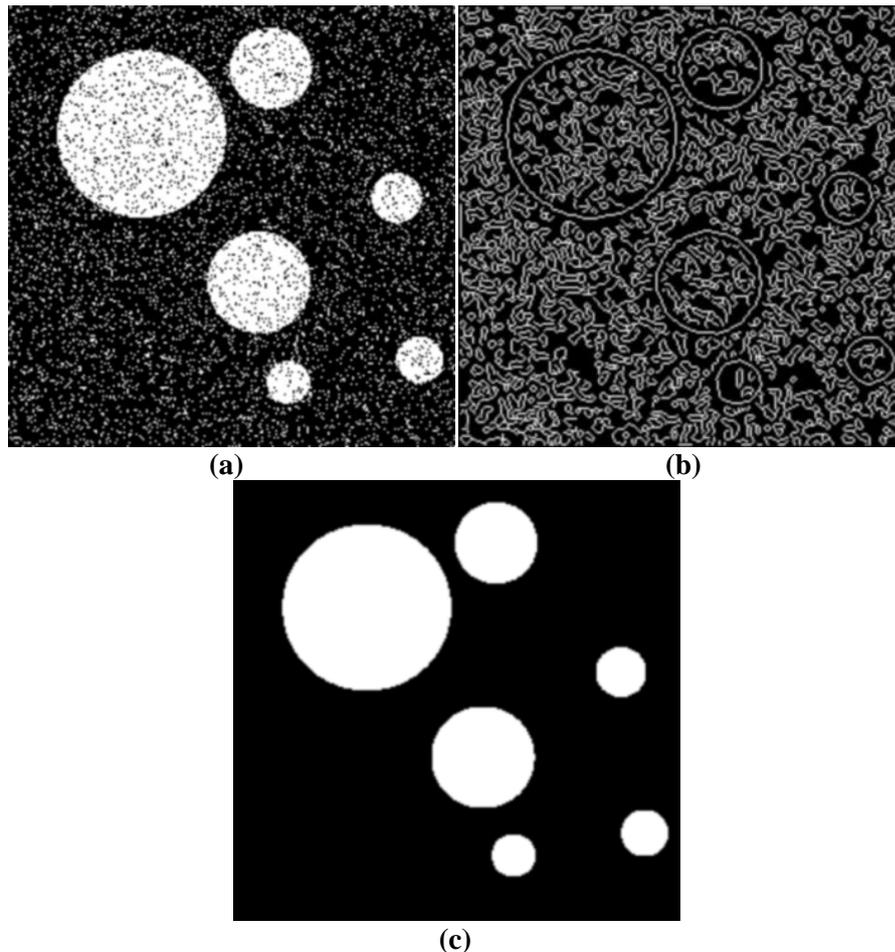


Figure 19 (a) An example for test images with Salt & Pepper noise added where $L_{Noise} = 1$, (b) edge map of the image with noise, (c) detected circles by CDDI.

The detection performance of the proposed method is also compared with MHT and RCD. The same data set is used and the same parameters are selected for the proposed method. For MHT r_{min} , r_{max} and r_{step} are selected as 10, 50 and 1 and for RCD T_f , T_{min} , T_a , T_d , T_r are 2×10^3 , 50, 10, 1 and 0.8, respectively. Detected circles are analyzed to evaluate the probability of detection (P_D) and probability of false alarm (P_{FA}) as in [43]. In this case, all the pixels in the detected circles are compared with the pixels in the original circles; so these measures can also give idea about the success of estimated parameters of circles. In order to define these measures the following sets of pixels are defined:

- True detections: pixels existing in both in the original and detected circles.
- Miss detections: pixels existing in one of the original circle but not in any of the detected circle.
- False detection: pixels existing in one of the detected circle but not in any of the original circle.

$$P_D = \frac{n_d}{n_d + n_m} \tag{68}$$

$$P_{FA} = \frac{n_f}{\# \text{ of all circles' pixels}}$$

In Figure 20, the detection performances of each of the approaches are presented, where up to $L_{Noise} = 0.75$ detection probability of the approaches are very close to 1. For more noisy images, the detection probability of the proposed approach is worse than the other approaches but it always produce least number of false circles.

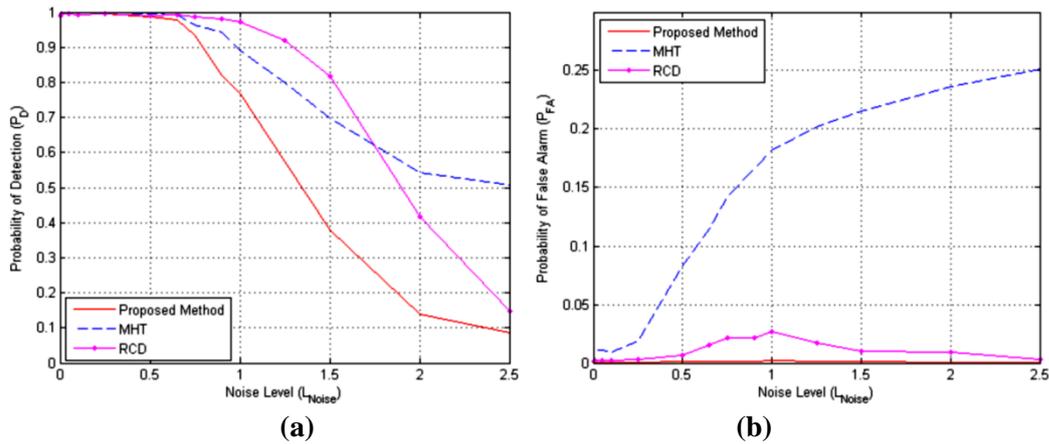


Figure 20 (a) Detection probability and (b) false alarm probability of the proposed method with respect to the added noise level.

More comprehensive analysis about their detection performance is made by generating the region of convergence (ROC) curves of the approaches, while changing the parameters of the approaches over a set of samples of their possible values. In this experiment, the same data set in the previous cases is used and the ROC curves are obtained for different noise levels (see Figure 21). To have better understanding, the ROC curves are plotted as precision vs. recall curves where precision and recall are defined as below:

$$\text{Recall} = P_D$$

$$\text{Precision} = \frac{n_d}{n_d + n_f} \quad (69)$$

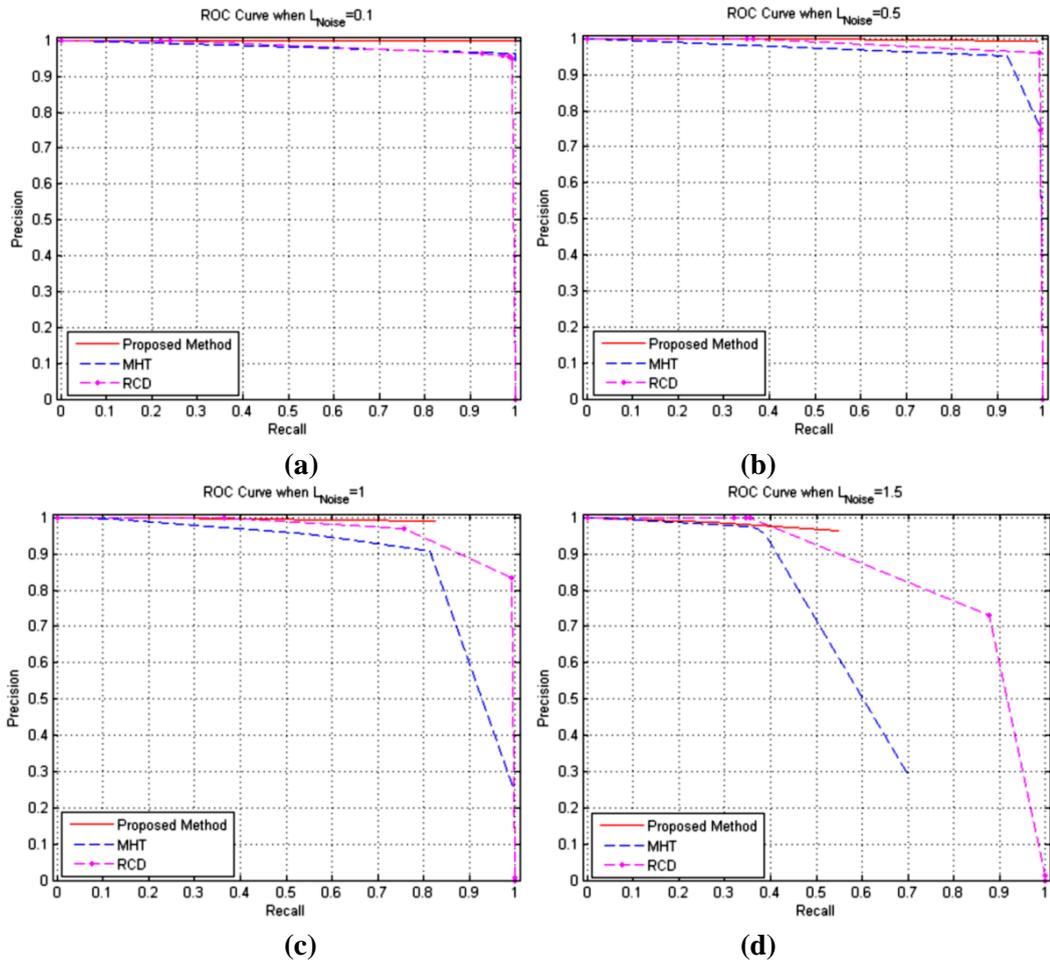


Figure 21 ROC curves of each method when L_{Noise} is equal to (a) 0.1, (b) 0.5, (c) 1, (d) 1.5.

As seen from the curves, the proposed approach is always precise. As the level of noise increases, it is not possible to detect circles by increasing the false alarm rate since the connected components of the edge map are ruptured into many small segments due to the noise. When compared with the curves of MHT and RCD it can be seen that, up to a certain

level of noise level, the proposed method has better performance and it can preserve its precision even for high detection rates.

In the next experiment, the performance of the algorithm is tested when the circles are not fully visible. As seen in Figure 22 (a), there are 12 circles with missing parts. The detected circles using this method when α, β, γ are 0.5, λ is 0.75, μ_{\min} is 0.25 and μ_{\max} is 1.75 are in Figure 22 (b). In this case, it is possible to detect circles when more than $\frac{3}{4}$ of its edges exist. However, as it can be seen clearly from Figure 22 (c), it is possible to detect a circle even if more than half of the circle is missing, when α, β, γ are 0.25, λ is 0.75, μ_{\min} is 0.25 and μ_{\max} is 1.75 since the bounds for the conditions given in the previous section are loosened. For each of the case, n_n and n_p are set as 0.

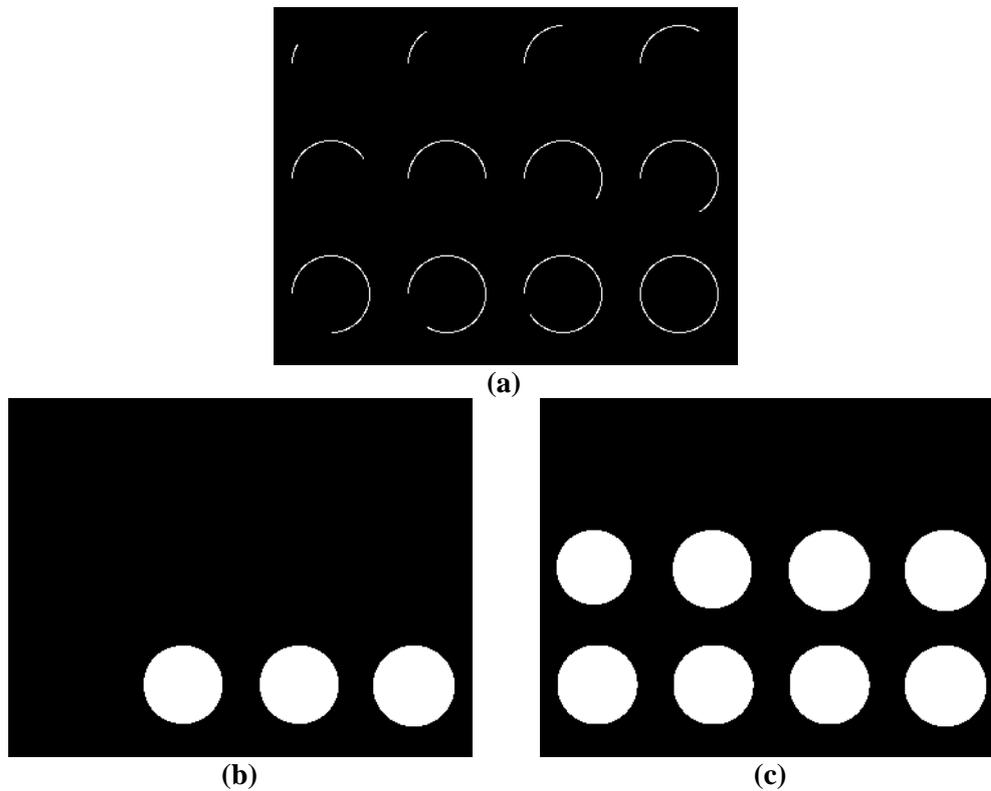


Figure 22 (a) Circle edges with missing parts and detection results of CDDI when (b) α, β, γ are 0.5, λ is 0.75, μ_{\min} is 0.25 and μ_{\max} is 1.75, (c) α, β, γ are 0.25, λ is 0.75, μ_{\min} is 0.25 and μ_{\max} is 1.75.

Another experiment is performed to understand whether it is possible to detect connected circles on an image. For this purpose, a synthetic image, given in Figure 23 (a), is generated. CDDI is run when α, β, γ are 0.75, λ is 0.75, μ_{\min} is 0.75 and μ_{\max} is 1.25. In this case, n_n is set to 2 and n_p is set to 5 since the connection points of the edges will be broken by the algorithm. As seen in Figure 23 (b), it is possible to detect all the circles in the image by CDDI approach.

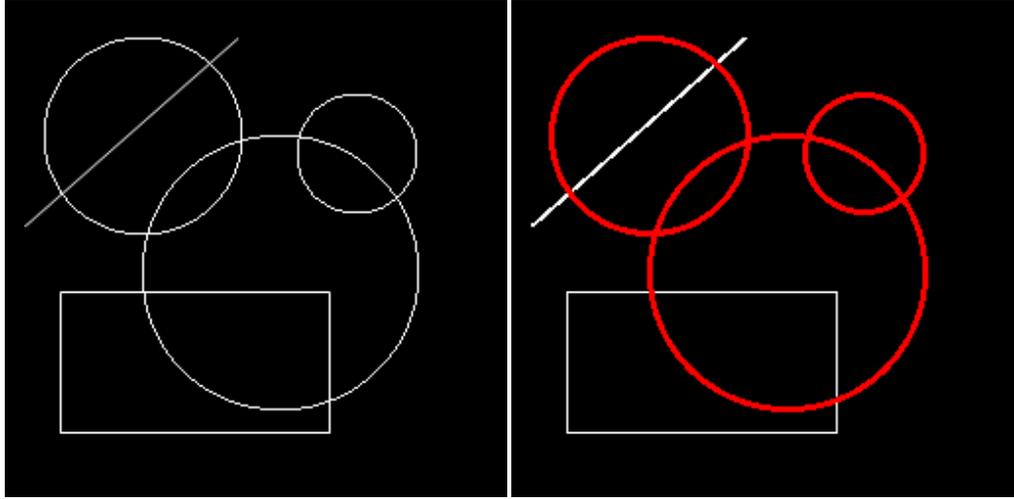


Figure 23 (a) Circles connected with other shapes and (b) detected circles using CDDI when $\alpha, \beta, \gamma, \lambda$ are 0.75, μ_{min} is 0.5, μ_{max} is 1.5, n_n is 2 and n_p is 5.

The comparisons of the proposed method with MHT and RCD methods are made in terms of the computation efficiency. For this purpose, a 1000x1000 image is created with randomly located circles with random radii between 10 and 50 and the number of circles is increased in order to analyze the computation efficiency of the approaches. The numbers of the circles are increased from 5 to 50 and the experiments are repeated for 100 times for each case. During the experiment, α, β and γ are set to 0.5, λ is set to 0.75, μ_{min} is set to 0.5 and μ_{max} is set to 1.5 where n_n and n_p are set as 0 for the proposed method; r_{min}, r_{max} and r_{step} are selected as 10, 50 and 1, respectively for MHT and $T_f, T_{min}, T_a, T_d, T_r$ are $2 \times 10^3, 50, 10, 1$ and 0.6, respectively for RCD method. Note that, for each case, P_D is above 99% and the P_{FA} is below 1%. Obtained results are given in Figure 24. It can easily be seen that the complexity of the proposed method is less than the other two methods hence the proposed method becomes much faster as the complexity of the scene (i.e. the number of edges) increases.

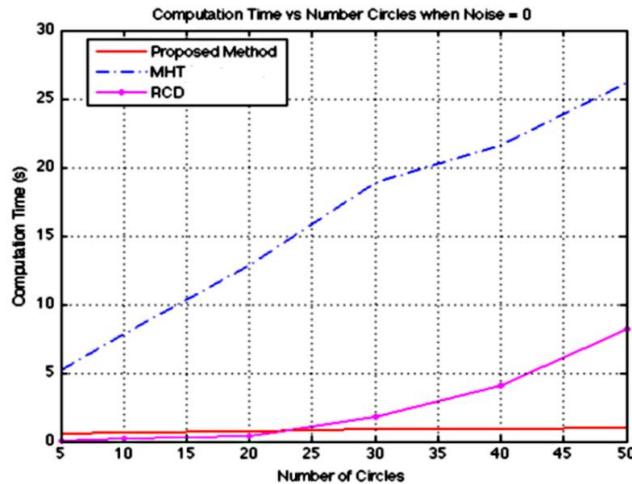


Figure 24 Computation durations of MHT, RCD and CDDI for different number of circles in the image.

Durations of MHT and proposed approach increase linearly with respect to the number of the circles where it increases exponentially for RCD. However, in the proposed method all the calculations are done for each of the connected component, independently. Hence, we also execute MHT and RCD for each of the connected component in the image with the same scenario described previously for a fair comparison. However T_f in RCD method is set to 20 for this case since there is no need to iterate much unless finding circle. For this case, the obtained results are given in Figure 25. The results show that RCD method has better performance when there is no noise in the image but it becomes comparable as the noise level increases and the proposed method is the best while the noise level increased and the scene becomes more complicated.

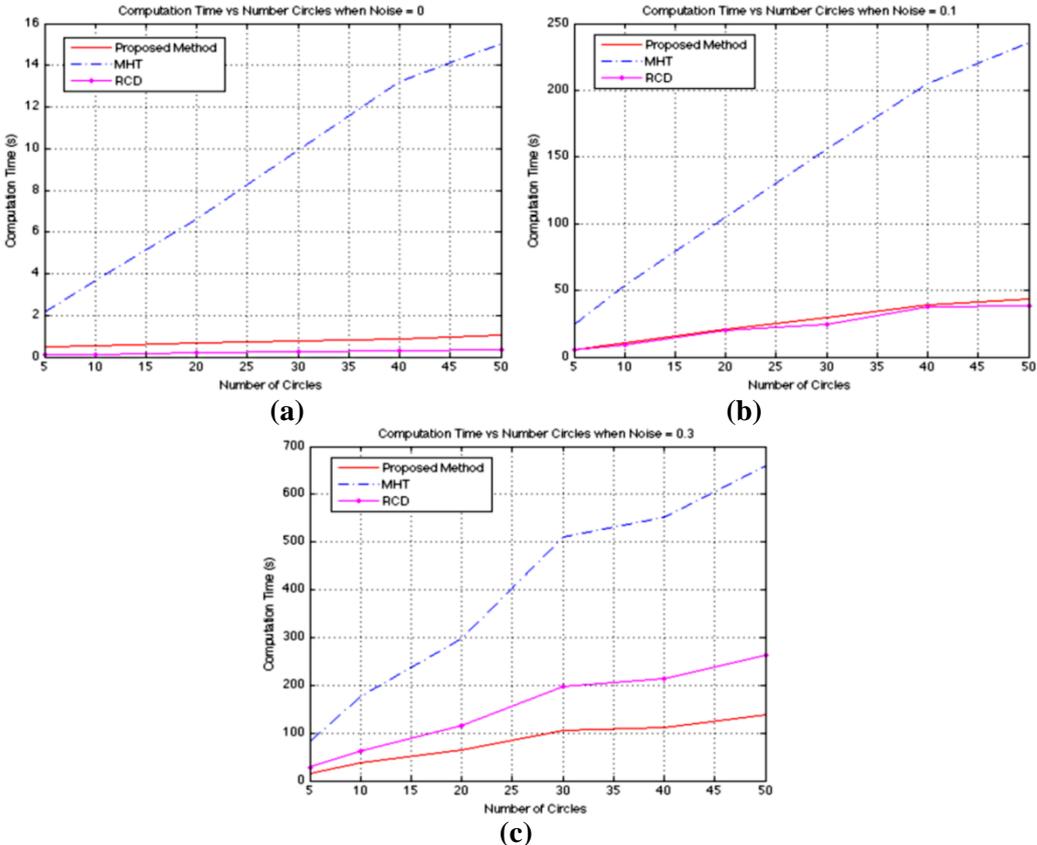


Figure 25 Computation durations of MHT and RCD when evaluated for each connected component and CDDI for different number of circles in the image when noise level is (a) 0, (b) 0.1, (c) 0.3.

The methods are also examined using real world data and the obtained results are presented in Figure 26 to Figure 29 and the durations of each approach for each of the images are given in Table 3.

Table 3 Execution durations of different methods in seconds for different images.

		MHT	MHT CC	RCD	RCD CC	CDDI
Image Name	Image Size					
Coin	300x246	4.986	2.537	2096.114	2.734	0.689
Eye	330x246	2.968	1.794	946.707	2.391	0.636
Football	470x480	17.083	8.567	2359.590	10.177	3.265
Oil Refinery	451x600	21.879	16.296	-	13.560	4.115

In this experiment, α , β and γ are set to 0.5, λ is set to 0.75, μ_{min} is set to 0.5 and μ_{max} is set to 1.5 where n_n and n_p are set as 0 for the proposed method; r_{min} , r_{max} and r_{step} are selected as 10, 50 and 1, respectively for MHT and T_f , T_{min} , T_a , T_d , T_r are 2×10^5 , 100, 10, 1 and 0.7, respectively for RCD method. In Figure 29, the detection result of RCD method is not given for a satellite image taken from Google Earth since the method cannot produce an output after several hours. As seen from these figures, the detection results are quite similar for such a complicated data; however, the computation duration is 21.879s for MHT, where it is 4.065s for the proposed method. These experiments are also repeated when MHT and RCD methods are executed for each connected component in the image and their outputs are also given in Figure 26 to Figure 29 and computation durations are given as MHT CC and RCD CC in Table 3. The results show that for this case, RCD method produces the outputs in comparable durations with the other methods but the proposed method is still the fastest.

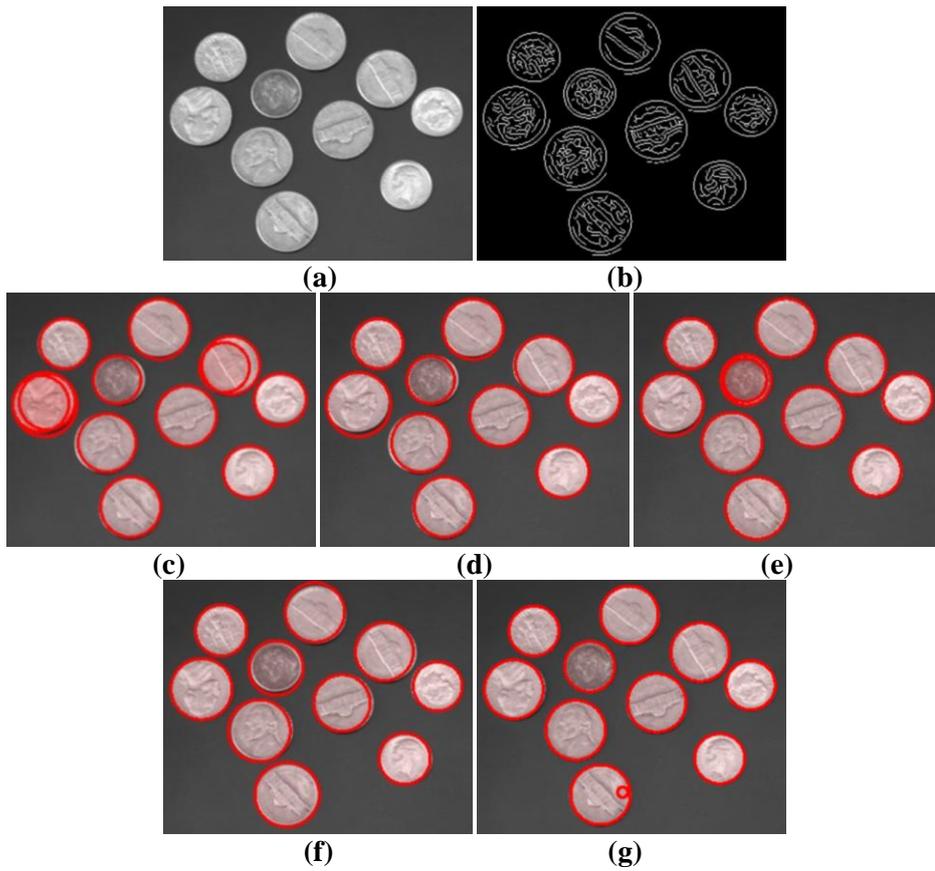


Figure 26 (a) Coin image, (b) its edge map and circle detections using (c) MHT, (d) MHT in connected components, (e) RCD, (f) RCD in connected components and (g) CDDI.

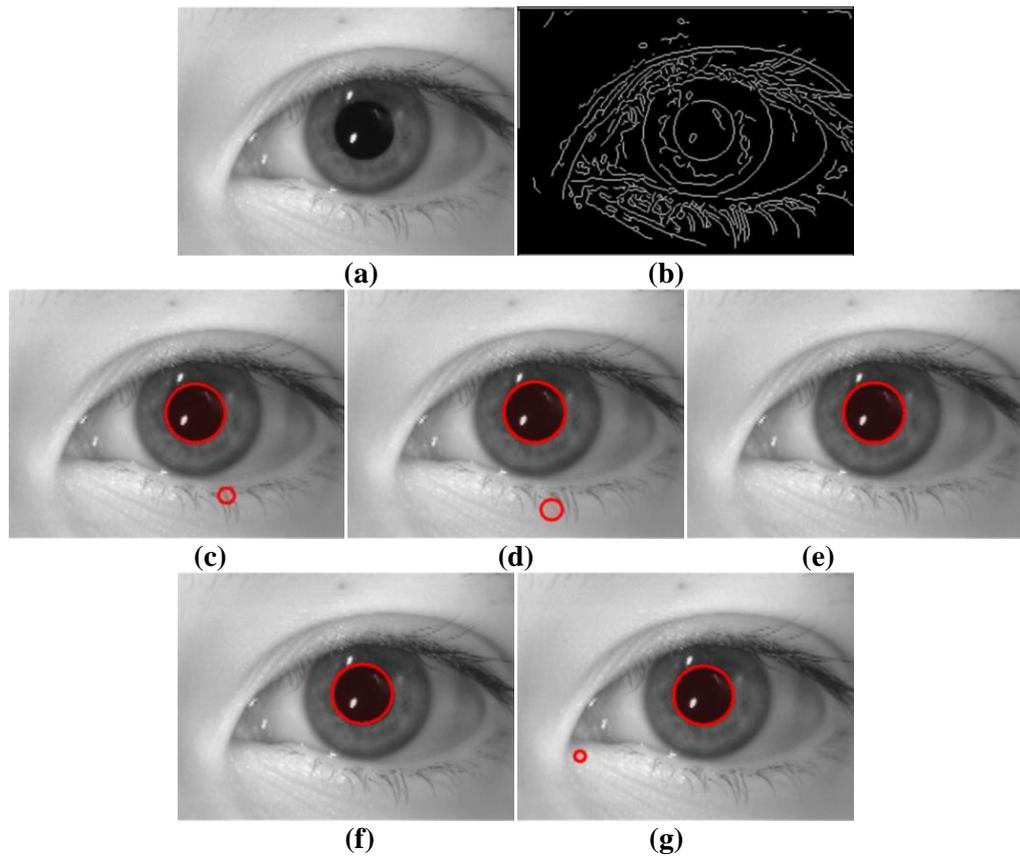


Figure 27 (a) Eye image, (b) its edge map and circle detections using (c) MHT, (d) MHT in connected components, (e) RCD, (f) RCD in connected components and (g) CDDI.

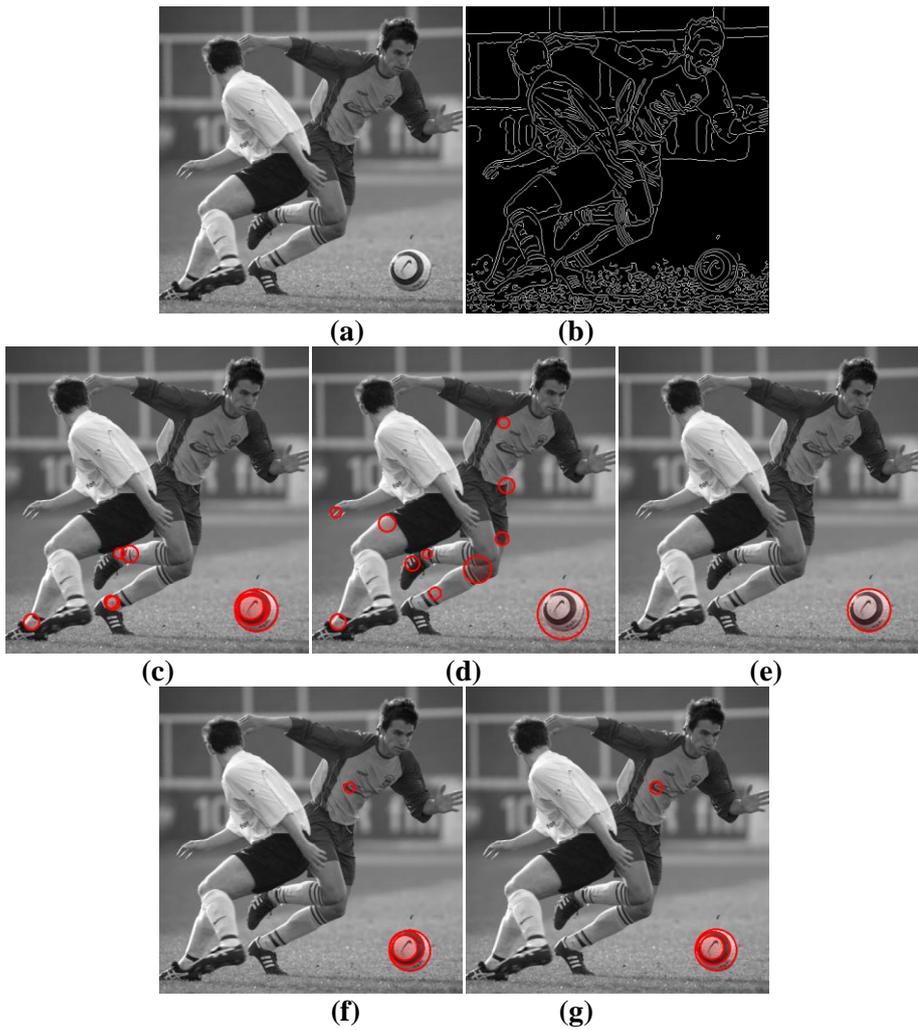


Figure 28 (a) Football image, (b) its edge map and circle detections using (c) MHT, (d) MHT in connected components, (e) RCD, (f) RCD in connected components and (g) CDDI.

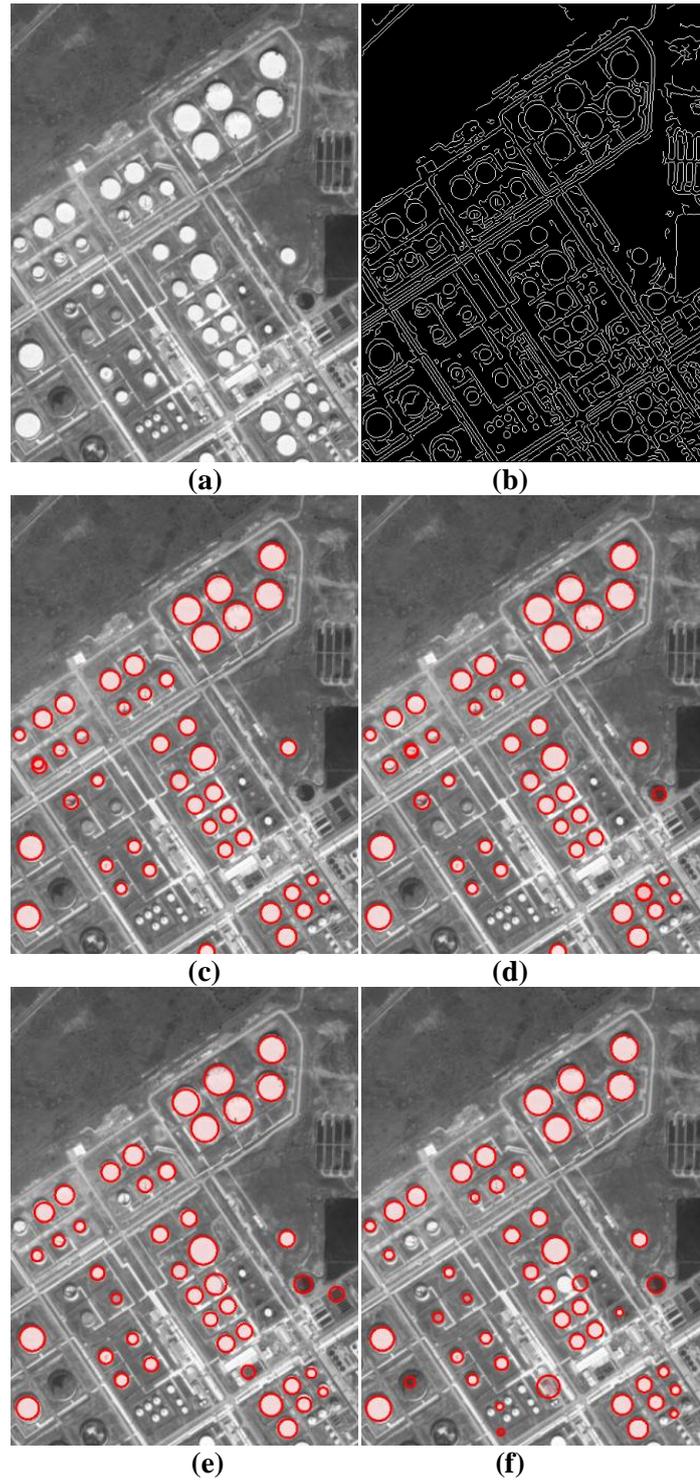
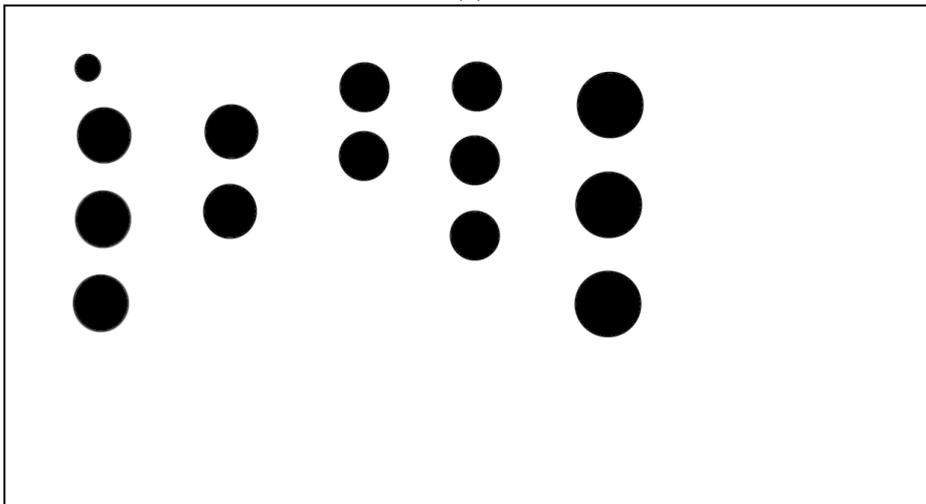


Figure 29 (a) A satellite image of an oil refinery taken from Google Earth, (b) its edge map and circle detections using (c) MHT, (d) MHT in connected components, (e) RCD in connected components and (f) CDDI.

Another experiment is performed for detection of POLs in many different oil refineries where the images are captured from Google Earth. In this experiment, the resolution of the images vary a lot hence the stability of the methods for only one parameter set is concerned. The data set is formed by selecting 71 images which contains 2057 POLs. Besides, the ground truth for the images is generated in order to compare the detection performance of the methods. Two examples from the data set are given in Figure 30 and Figure 31 where the resolution and scene variation can be seen easily.

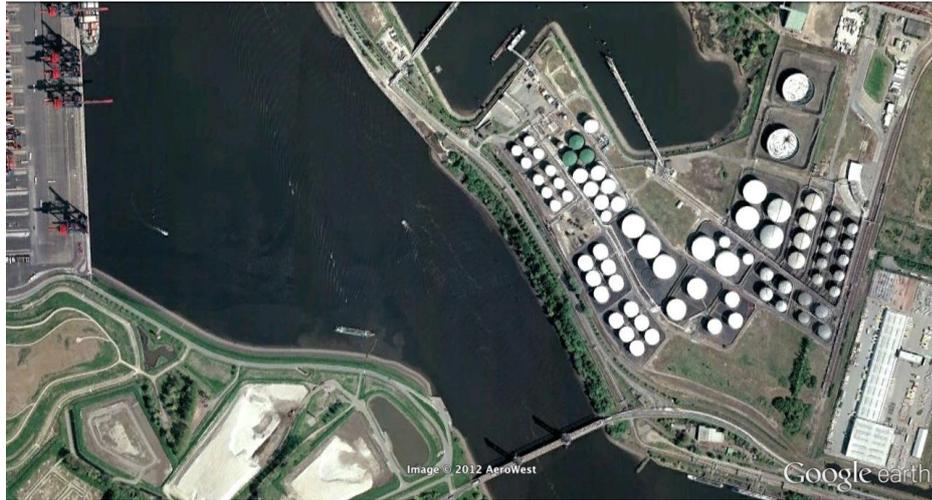


(a)

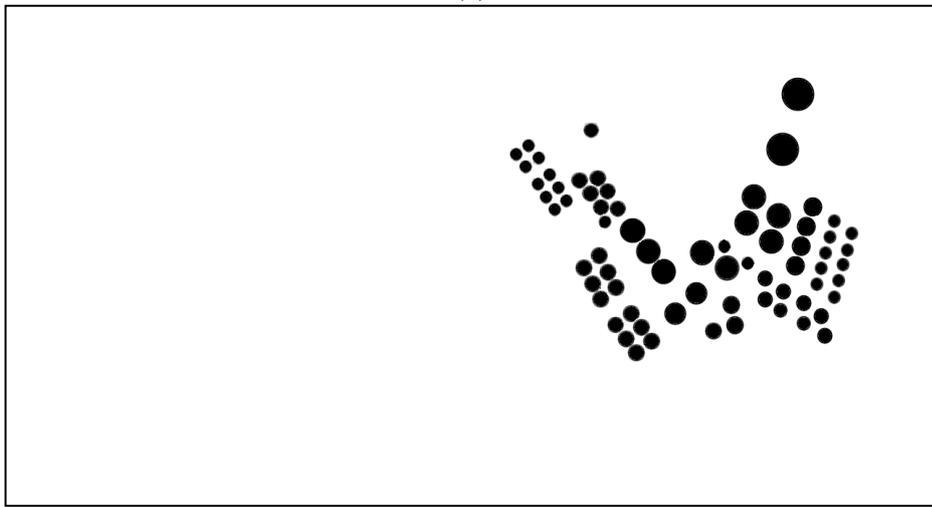


(b)

Figure 30 (a) An example image or POL test data set and (b) its ground truth image.



(a)



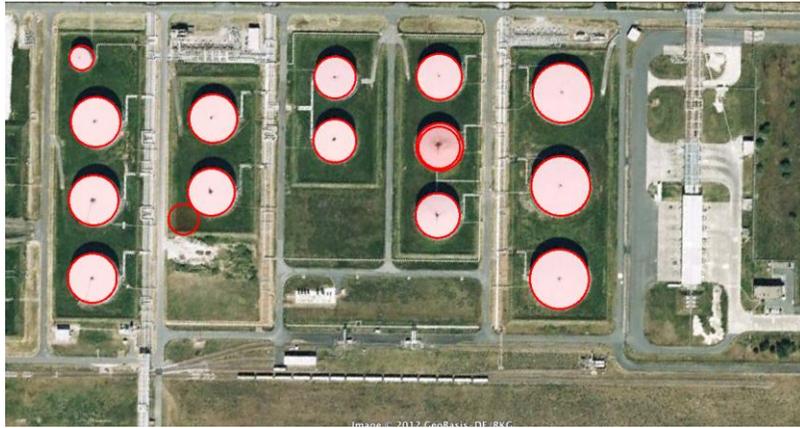
(b)

Figure 31 (a) An example image or POL test data set and (b) its ground truth image.

In these experiments the proposed method is only compared with MHT method since the scenes are very complex and RCD approach generates results in unpractical durations (i.e. more than an hour). Hence, RCD method is utilized on connected components as explained in the previous experiments. Recall and precision are utilized again as given in Equation (69) in order to measure the success of the methods but they are calculated by two different approaches. In the first approach detections are counted target by target which means that if at least half of a detected circle coincides with a POL on the image and number of true detections is increased by one and number of false detections is increased by one otherwise. Besides, number of misses is increased by one if there is no circle that coincides with a POL. Note that, if there are more than one circle coinciding with a POL, only one of them is counted for true detections and the others are counted for false detections. In the second approach the performance measures are evaluated pixel by pixel which means each pixel in the detected circle examined whether it is on a POL or not and true and false detections are counted accordingly. The POL pixels on ground truth are also examined to find the missed pixels to evaluate recall and precision.

For MHT r_{min} , r_{max} and r_{step} are selected as 10, 50 and 1; for RCD T_f , T_{min} , T_a , T_d , T_r are 100, 100, 10, 1 and 0.65, respectively and for proposed approach α , β and γ are set to 0.5, λ is set to 0.75, μ_{min} is set to 0.5 and μ_{max} is set to 1.5. But for this RCD is evaluated for connected components and n_p is set as 3 for RCD and CDDI. These methods are tested when n_n is 1 and 2 in order to see the effect of this parameter on the detection performance. Moreover, edge map of the images are extracted after applying bilateral filter [44] where size of the Gaussian bilateral filter is 31, standard deviation of the filter is 11 in spatial domain and 1 in intensity domain. Two examples of the results of these approaches for the data set are given in Figure 32 and Figure 33. As seen from these examples, the detection performances are very similar to each other. For all the data set, mean performance measures, i.e. recall, precision, and the mean processing durations are given in Table 4. As seen from this table, the POL detection performances of these methods are very close to each other. When the results are analyzed POL by POL when $n_n=2$ CDDI gives the best recall rate which means CDDI detects more POLs than the other ones where it is still two times faster than MHT approach. On the other hand, the precision of this method becomes worse which can be expected. An interesting point is that the circle wise recall rate is smaller than the pixel wise recall rate for MHT while it is just the opposite for CDDI method. When the results are analyzed it is shown that this is mainly because of MHT can hardly detect the small circles where CDDI has better performance for small circles. Besides, when the POL has a shadow, MHT detects some of them in addition to the POL itself, hence many of the POL pixels can be detected while CDDI can only detect the POL.

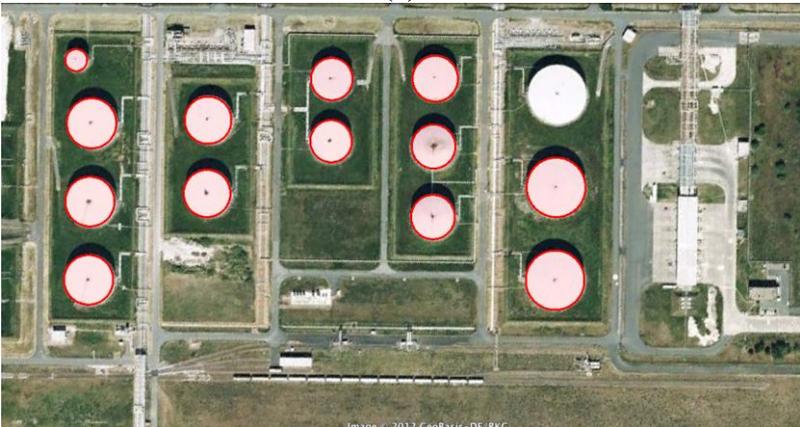
Note that, the performance of the RCD is still worse than CDDI especially for larger circles. This can be easily seen from the reduction of the recall rates from circle by circle evaluation to pixel by pixel evaluation. One of the major drawbacks of this method is a circle is detected more than once which is the reason of the big difference in precision rate between circle by circle evaluation and pixel by pixel evaluation. Hence, this method requires a post processing step to reduce repetitive detections.



(a)



(b)



(c)



(d)

Figure 32 Detected circles on the image given in Figure 30 by (a) MHT, (b) RCD in connected components when $n_n=2$, (c) CDDI when $n_n=1$ (d) CDDI when $n_n=2$.



(a)



(b)



(c)



(d)

Figure 33 Detected circles on the image given in Figure 31 by (a) MHT, (b) RCD in connected components when $n_n=2$, (c) CDDI when $n_n=1$ (d) CDDI when $n_n=2$.

Table 4 Execution durations and performance measures for POL images data set of different approaches

		MHT	RCD_CC $n_n=2$	CDDI $n_n=1$	CDDI $n_n=2$
Duration	seconds	151.1819	120.5431	35.4254	74.6334
Circle Wise Performance	Precision	0.85944	0.65954	0.73014	0.66462
	Recall	0.69385	0.72931	0.71158	0.76714
Pixel Wise Performance	Precision	0.84384	0.87204	0.82892	0.77944
	Recall	0.74418	0.57687	0.67941	0.72922

To sum up, CDDI method is faster than the two commonly used circle detection approaches especially when the scene is complicated. The experiments also show that when the parameters are set for similar performances, CDDI is robust and capable of detecting deformed circles in different size by use of the same parameter set.

2.6 Parameter Selection for CDDI

CDDI has eight parameters to adjust. α , β , μ_{min} and μ_{max} are used to decide whether the shape of the examined object is a circle or not. As these parameters get close to 1, the algorithm tends to detect only the perfect circles. So, in order to handle the distorted circles, α , β , μ_{min} should be decreased where μ_{max} should be increased.

As a circle is symmetric around the origin, any two profiles of a circle is equivalent to each other. The parameters γ , λ are used to determine the similarity of the two evaluated profiles so as these parameters get close to 1, the algorithm tends to detect only the perfect circles.

The last two parameters, n_n are n_p , are used to handle missing edges of the examined shapes. If the contrast of the objects are high enough and if unconnected objects are examined these parameters should be selected as 0. However, as the contrast gets lower or there exist connected objects these values should be increased.

As a guideline for the selection of these parameters, α , β , γ , λ can be initialized from 0.75. This value is seen as an optimal value for detection of the circles in real life scenes when the contrast is high enough. If the results are not good enough, these values can be decreased, together. As these parameters define the similarity of the shape to a circle and each of them is defined in (0,1], it is easier to set them to the same value. For most of the cases, 0.5 and 1.5 are very appropriate values for μ_{min} and μ_{max} , respectively. If the examined circles do not have huge missing parts, it is not required to be tuned. On the other hand, n_n are n_p depends only on the image. These values should be initialized as 0 and the results should be observed. If the image is noisy or the circles are intersects with other objects these values should be increased so it is better to analyze the edges on the images

before changing these parameters. Since increasing the values of these parameters induces false detections and causes increase in the operation duration of CDDI, n_n and n_p values should be selected optimally.

2.7 Discussion

In this chapter of the thesis a novel circle detection approach is proposed. The derivation of the method is given and the validity of the method is shown by the experiments. Since circles are detected by finding local peaks in 1-D transform domain, the proposed method requires much less memory and computation power compared to other methods in the literature. The experimental results performed using synthetic and real world data showed that the proposed method is robust against noise and capable of detecting circles with missing parts. Hence, by use of this method, it is possible to reduce the computation time required while preserving the detection performance of these shapes. Besides, since the method does not require a priori knowledge about the scene like the circles' size as in HT based approaches or scene complexity as in RCD, the parameters do not vary according to the scene.

The main drawback of the method is its dependency to the quality of the extracted connected components and for some cases it may fail due to the extraction of unconnected boundaries. From this perspective, the effects of the edge detection approaches are analyzed in terms of their effect on detection performances and Canny edge detection method is selected as the best of all since the continuity of the edges are taken into account and the extracted parts forced to be continuous by this method.

CHAPTER 3

RECTANGLE DETECTION

Rectangle detection is a common problem for many different ATR applications from aerial image analysis to license plate recognition. In order to solve this problem, different approaches have been proposed in the literature and many of these methods depend on detection of line and edge primitives on the image. The method proposed by Lagunovsky and Ablameyko [45, 46] is based on line primitives. First of all linear primitives are extracted to find the line segments, which are then grouped in straight line clusters. The length and orientation of these straight lines are compared and used to detect quadrangles that are further approximated as rectangles.

Lin and Nevatia [47] studied the problem of rectangle/parallelogram detection in aerial images and proposed an approach based on line detection, and selection of line segments within a range of values (determined by maximum and minimum building sizes). Once an initial line segment is found, anti-parallel lines are searched and a pair of anti-parallel lines is used to define a search region, where the remaining two sides of the rectangle are searched. Tao et. al. also proposed an approach for extracting rectangular buildings from aerial images [48]. In their approach, edge elements are found and linear elements are extracted using a splitting arithmetic. Start-point, end-point and orientation of each linear element are used to detect parallel lines, and pairs of parallel lines are used to form rectangular primitive structures. Finally, these primitives are merged to form rectangles.

Zhu et. al. used a different approach to detect rectangular particles in cryo-electron microscopy images, by proposing a Rectangular Hough Transform (RHT) which uses a 2-D accumulator array to detect the center and orientation of the rectangle [49-51]. Despite being fast and producing good results, it is applicable only when all rectangles in the image have the same dimensions, and these dimensions must be known.

In [52], another Hough Transform based approach, Windowed Hough Transform (WHT), is proposed to detect rectangles. In this method, rectangles are directly detected in that domain by examining the relative locations and values of the peaks on the HT domain. Wu et. al. proposed another method to detect large rectangular shapes under highly noisy conditions [53]. For this purpose, a HT based approach, Improved Progressive Probabilistic Hough Transform (IPPHT), which uses a kind of dilation to detect edges that are not exactly straight. Moreover, a fast rectangle fitting approach is developed that uses the extracted lines to detect rectangles.

Another line primitive based approach is proposed by Liu et. al. where the detected lines are used to identify whether there is a rectangle or not by use of a Markov Random Field model [54]. They stated in the manuscript that this method is capable of detecting multiple rectangles on the image with different size and orientation and it is shown that the approach is much faster than the Randomized Hough Transform (RHT) by the experiments performed.

3.1 Windowed Hough Transform (WHT)

As explained in Chapter 2.1, HT is in fact a voting procedure by which the edges on the image with similar geometric properties are grouped together. Hence, the local maxima on the accumulation space are detected as the desired object.

In linear HT, lines are represented in polar coordinates (ρ, θ) as seen in Figure 34 and for each edge point on the image at (x,y) all the lines passing through that point, i.e. satisfying the condition in Equation (70), is found [55].

$$\rho = x \cos \theta + y \sin \theta \quad (70)$$

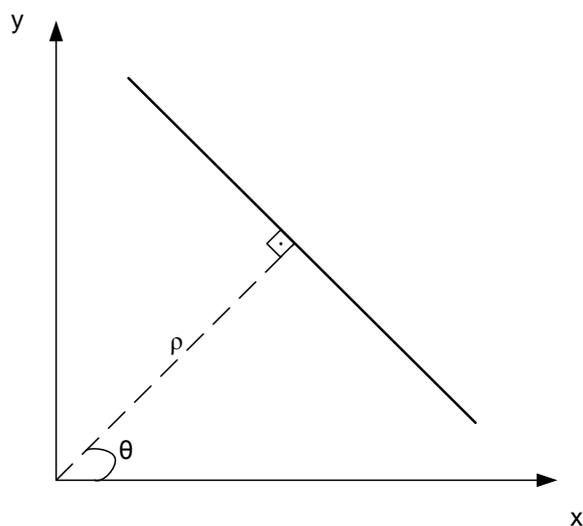


Figure 34 Representation of a line in polar coordinates.

WHT is based on detecting the pattern of rectangles on the Hough plane after the linear Hough Transform [52]. In other words, the aim is not detecting each line of a rectangle separately by HT but identifying the geometric characteristics of a rectangle in linear HT plane. These geometric characteristics can be easily seen in Figure 35 and can be listed as:

- There are four peaks in HT plane, i.e. H_1, H_2, H_3, H_4 .
- The peaks are in pairs, which have same angles, i.e. α_0 and α_1 .
- The pairs are 90° apart from each other, i.e. $|\alpha_0 - \alpha_1| = 90^\circ$.
- The peak pairs are symmetric w.r.t θ axis, which means their ρ values sum up to 0.
- The distance between the pairs in ρ -axis is equal to the length of one of the sides of the rectangle.
- The value of each peak in a peak-pair is equal to the other peak and this value is equal to the length of one of the sides of the rectangle, i.e. peak values of H_1 and H_2 are same and it is equal to the distance between H_3 and H_4 in ρ -axis and vice versa.

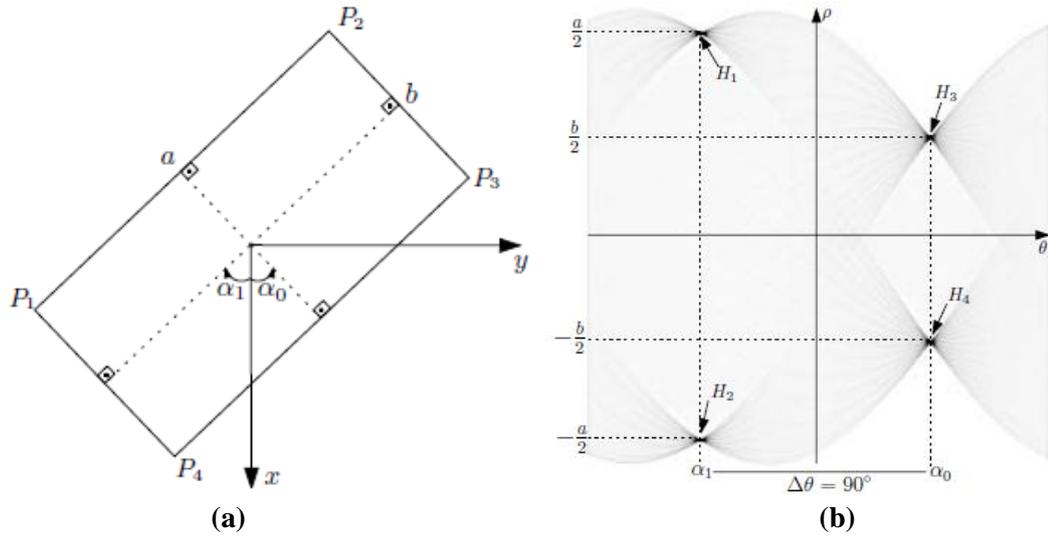


Figure 35(a) A rectangle centered at the origin and (b) its linear HT[52].

On the other hand, when there are many rectangles especially when the scene is very complicated, it is very hard to find those peak relations and detect the rectangles. Hence Jung et. al. proposed in WHT method to determine the existence of rectangles in a moving kernel with a predetermined size[52]. For each position of this kernel, HT is evaluated for the edge pixels inside the kernel and the peaks, H_i and H_j , satisfying the following conditions are selected as peak pairs:

$$\Delta\theta = |\theta_i - \theta_j| < T_\theta \quad (71)$$

$$\Delta\rho = |\rho_i - \rho_j| < T_\rho \quad (72)$$

$$|C(\theta_i, \rho_i) - C(\theta_j, \rho_j)| < T_L \frac{(C(\theta_i, \rho_i) + C(\theta_j, \rho_j))}{2} \quad (73)$$

where $C(\theta, \rho)$ is the Hough Transform, T_θ , T_ρ and T_L are the angular, distance and normalization thresholds, respectively. T_θ and T_ρ are selected as small values in order these peaks to correspond parallel lines symmetric with respect to the angular axis. T_L is also a small value, i.e. smaller than 0.5, to guarantee that the peaks have similar values which implies the corresponding lines have similar lengths.

From each of those peak pairs an extended peak pair, $P_k = (\alpha_k, \xi_k)$, is evaluated as:

$$\alpha_k = \frac{1}{2}(\theta_i + \theta_j) \quad (74)$$

$$\xi_k = \frac{1}{2}|\rho_i - \rho_j| \quad (75)$$

Finally a rectangle is detected at the center of the kernel with orientation of α_k and side lengths of ξ_k and ξ_l if there is a relation given in Equation (76) between two extended peak pairs P_k and P_l .

$$\Delta\alpha = \left| \alpha_k - \alpha_l - 90^\circ \right| < T_\alpha \quad (76)$$

Note that, the size of the moving kernel is determined according to the size limits of the desired rectangles to be found. This kernel is defined in ring shaped where the internal diameter is equal to the minimum and the external diameter is equal to the maximum edge size of the rectangles to be found, D_{min} and D_{max} , respectively.

3.2 Rectangle Detection by Directional Integrations (RDDI)

As it can be seen from Figure 36, a rectangle can be defined with five parameters x_c , y_c , a , b and θ and these parameters refer to its coordinates of center, width, height and orientation, respectively.

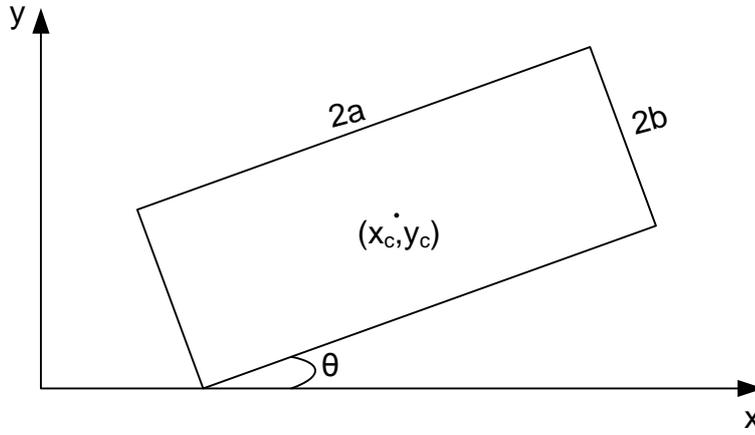


Figure 36 A rectangle and its parameters x_c , y_c , a , b and θ .

In this study, in order to detect a rectangle and extract its parameters it is also proposed to use directional integrations given in Equation (77), where f is defined as a rectangle defined in discrete spatial domain.

$$\begin{aligned} g_1[m] &= \sum_n f[m, n] \\ g_2[n] &= \sum_m f[m, n] \end{aligned} \quad (77)$$

When the orientation of the rectangle is equal to zero, the interpretation of the resultant functions is very easy since $g_1[m]$ would have two peaks at $m=x_c-a$ and $m=x_c+a$ and $g_2[n]$ would have two peaks at $n=y_c-a$ and $n=y_c+a$. The values of those peaks are also proportional to the side lengths of the rectangle and the resultant functions can be seen in Figure 37.

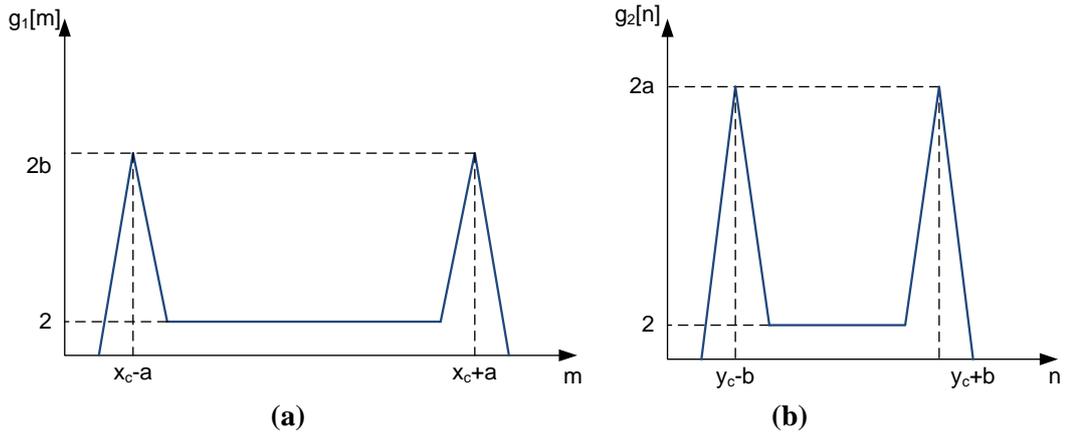


Figure 37 (a) $g_1[m]$ and (b) $g_2[n]$ for a rectangle when $\theta=0$.

On the other hand, when the orientation of the rectangle is nonzero, the evaluation of $g_1[m]$ and $g_2[n]$ become more complex which can still be done by use of differential length ds . As seen from Figure 38, $g_1[m]$ is composed of three parts from x_1 to x_2 , x_2 to x_3 and x_3 to x_4 , all of which are composed of sum of two different lines of the rectangle. For any line, the infinitely small length, ds , is evaluated by the following formula:

$$ds = \frac{dx}{\cos \phi} \quad (78)$$

where $\tan \phi$ is the slope of the line. From x_1 to x_2 and x_3 to x_4 $g_1[m]$ is equal to the sum of two small lengths on two lines perpendicular to each other and from x_2 to x_3 it is sum of the lengths on two parallel lines. Since, the slope of those lines are same as the orientation of the rectangle and dx is equal to 1 for discrete spatial domain, $g_1[m]$ can be found as given in Equation (79).

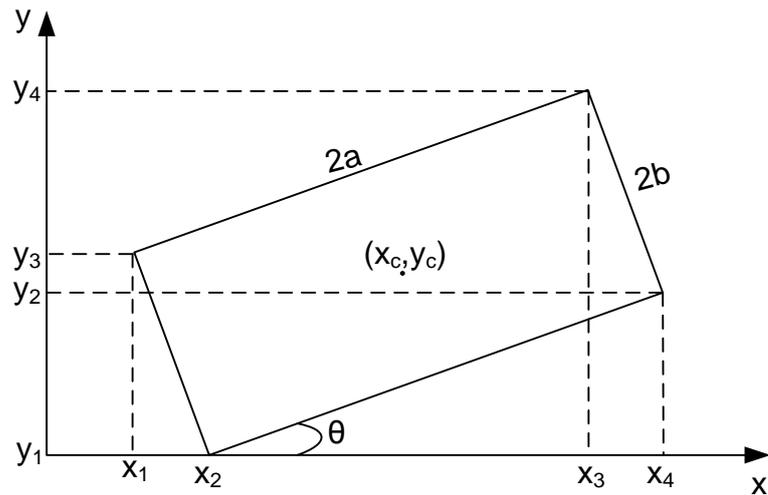


Figure 38 Rectangle with $\theta > 0$.

$$g_1[m] = \begin{cases} \frac{\cos \theta + \sin \theta}{\cos \theta \sin \theta}, & \text{if } x_1 \leq m < x_2 \text{ or } x_3 \leq m < x_4 \\ \frac{2}{\cos \theta}, & \text{if } x_2 \leq m < x_3 \\ 0, & \text{else} \end{cases} \quad (79)$$

$g_2[n]$ can be found after a very similar analysis as:

$$g_2[n] = \begin{cases} \frac{\cos \theta + \sin \theta}{\cos \theta \sin \theta}, & \text{if } y_1 \leq n < y_2 \text{ or } y_3 \leq n < y_4 \\ \frac{2}{\cos \theta}, & \text{if } y_2 \leq n < y_3 \\ 0, & \text{else} \end{cases} \quad (80)$$

From Equations (79) and (80) the graphs for $g_1[m]$ and $g_2[n]$ can be obtained as seen in Figure 39 and the values of the functions have the information about the orientation of the rectangle.

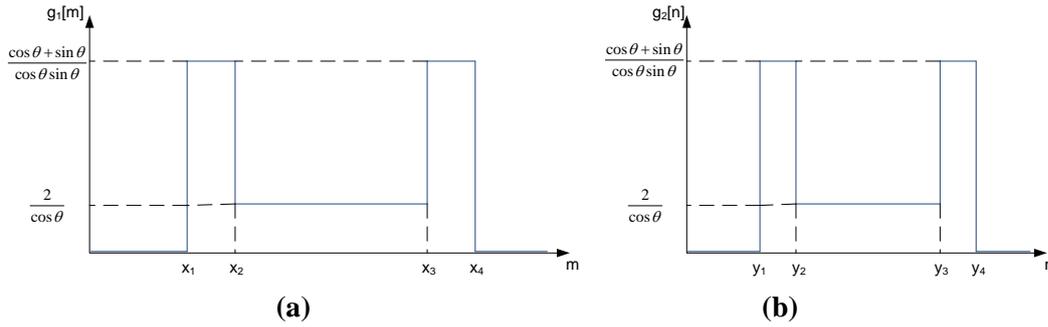


Figure 39 (a) $g_1[m]$ and (b) $g_2[n]$ for a rectangle when $\theta \neq 0$.

The side lengths, on the other hand, can be extracted by determination of their inflection points of the functions at where they are not continuous, i.e. x_1, x_2, x_3, x_4 and y_1, y_2, y_3, y_4 , since the distances between them are related to side lengths of the rectangle. These relations can be seen in Equations (81) and (82).

$$\begin{aligned} d_{11} &= x_2 - x_1 = 2b \sin \theta \\ d_{12} &= x_3 - x_2 = 2a \cos \theta - 2b \sin \theta \\ d_{13} &= x_4 - x_3 = 2b \sin \theta \end{aligned} \quad (81)$$

$$\begin{aligned} d_{21} &= y_2 - y_1 = 2b \sin \theta \\ d_{22} &= y_3 - y_2 = 2a \cos \theta - 2b \sin \theta \\ d_{23} &= y_4 - y_3 = 2b \sin \theta \end{aligned} \quad (82)$$

These results show that once the eight points given in Figure 38 can be detected, sufficient information about the parameters of a rectangle can be obtained, if there is one. Since they

are the inflection points in $g_1[m]$ and $g_2[n]$, in this study, it is proposed to detect these points using the zero crossings on the second derivatives of the functions and by determining a set of points that satisfy a set of condition and minimize a predefined cost function. However, the inflection points become undetectable, especially the ones at x_2 , x_3 , y_2 and y_3 , as the orientation of the rectangle increases since the derivative of the function become to have close values at each of the three regions (see Figure 40). Besides, depending on the side lengths ratio of the rectangle, the inflection points in the middle may scramble, i.e. x_3 becomes smaller than x_2 or y_3 becomes smaller than y_2 .

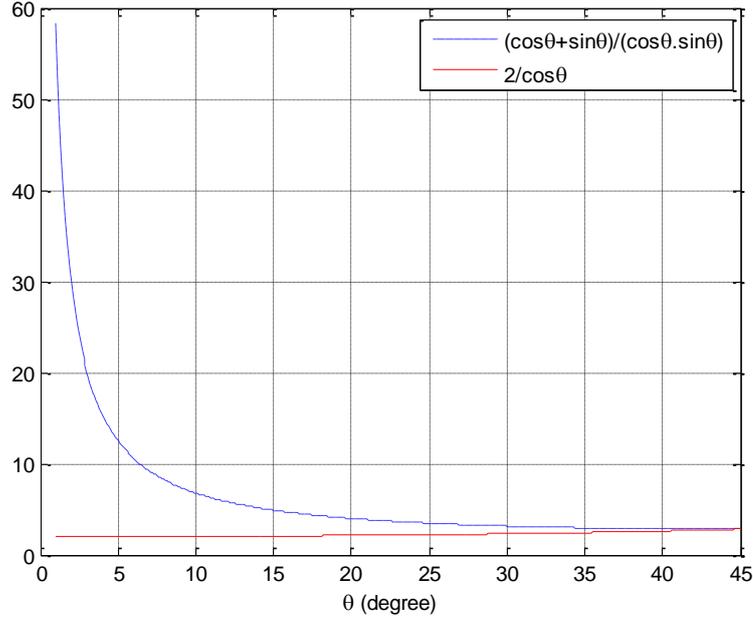


Figure 40 Possible values of integration of a rectangle in horizontal or vertical directions with respect to its orientation

Therefore, despite the orientation of a rectangle can be obtained by use of the integrations in horizontal or vertical direction, it is required to estimate the orientation at first in order not to miss a rectangle. Hence, in this study, it is proposed to estimate the orientation and it is obtained using the formula given in Equation (83) [56].

$$\tan 2\theta = \frac{2 \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left[(x_i - \bar{x})^2 - (y_i - \bar{y})^2 \right]} \quad (83)$$

where (x_i, y_i) 's are the locations of edges of the rectangle and \bar{x} and \bar{y} are the coordinates of center of mass of the rectangle which is evaluated as below:

$$x = \frac{1}{n} \sum_{i=1}^n x_i, \quad y = \frac{1}{n} \sum_{i=1}^n y_i \quad (84)$$

Then the directional integrations are evaluated along the lines in the direction of estimated orientation, θ , and in the direction of their normal lines $90+\theta$, to form $g_2[n]$ and $g_1[m]$,

respectively. Hence, if the orientation of the rectangle could be estimated exactly, the resultant $g_1[m]$ and $g_2[n]$ have the form as given in Figure 37 but because of the disturbances on the image or the distortions of the rectangle these functions are generally like as shown in Figure 39.

Therefore, it is still required to find the inflection points on $g_1[m]$ and $g_2[n]$ where the first derivatives of these functions are equal to zero. The sign of the first derivatives of $g_1[m]$ and $g_2[n]$ are also informative and an inflection point is a rising inflection point if the first derivative is positive where it is a falling inflection point if the first derivative is negative. Hence, the candidates of x_1, x_3, y_1 and y_3 are the rising inflection points in $g_1[m]$ and $g_2[n]$, where the candidates of x_2, x_4, y_2 and y_4 are the falling inflection points in $g_1[m]$ and $g_2[n]$. Note that, in this report, x_r and x_f are used to represent the set of rising and falling inflection points in $g_1[m]$ and y_r and y_f are used to represent the set of rising and falling inflection points in $g_2[n]$. Then, the distances between rising and falling edges and the mean values of $g_1[m]$ and $g_2[n]$, which are estimates of the possible values of the functions given in Equations (79) and (80), are evaluated to detect the rectangle.

3.2.1 Proposed Rectangle Detection Algorithm

The RDDI is very similar to described circle detection algorithm in Chapter 2.3 which is based on finding a set of points on two 1D functions satisfying a set of conditions. Equations (79) and (80) imply existence of only two peaks in each profile of the rectangle, which are also identical to each other. However, discretization, distorted shapes or noise on the image may cause other peaks to appear and/or desired peaks to be different from each other. So, some features are extracted from the profiles in order to detect the matching peaks in the profiles.

As the first step, the rising and falling inflection points are detected on the horizontal and vertical profiles as:

$$x_r := x \mid g_1[x] > g_1[x-1] \& g_1[x] \geq g_1[x+1] \quad (85)$$

$$x_f := x \mid g_1[x] > g_1[x+1] \& g_1[x] \geq g_1[x-1] \quad (86)$$

$$y_r := y \mid g_2[y] > g_2[y-1] \& g_2[y] \geq g_2[y+1] \quad (87)$$

$$y_f := y \mid g_2[y] > g_2[y+1] \& g_2[y] \geq g_2[y-1] \quad (88)$$

Secondly, the following features are extracted related to each hill in $g_1[m]$ and $g_2[n]$, given in Equations (89)-(92) in order to detect the similar peaks.

$$w_{1,i,j} = x_f(j) - x_r(i) \quad (89)$$

$$w_{2,k,l} = y_f(l) - y_r(k) \quad (90)$$

$$\mu_{1,i,j} = \frac{1}{w_{1,i,j}} \sum_{m=x_r(i)}^{x_f(j)} g_1[m] \quad (91)$$

$$\mu_{2,k,l} = \frac{1}{w_{2,k,l}} \sum_{n=y_r(k)}^{y_f(l)} g_2[n] \quad (92)$$

Based on Equations (79)-(82), if there is a rectangle on the image, for at least two of the peaks in each profile, the following conditions should be satisfied:

- Peak Similarity: The average value of two peaks in $g_1[m]$ and $g_2[n]$ are close to each other. Then,

$$\alpha \leq \frac{\mu_{1,i_1,j_1}}{\mu_{1,i_2,j_2}} \leq \frac{1}{\alpha} \quad (93)$$

$$\alpha \leq \frac{\mu_{2,k_1,l_1}}{\mu_{2,k_2,l_2}} \leq \frac{1}{\alpha} \quad \text{where } \alpha \in (0,1]$$

- Width Similarity: The width of two peaks in $g_1[m]$ and $g_2[n]$ are close to each other. Then,

$$\beta \leq \frac{w_{1,i_1,j_1}}{w_{1,i_2,j_2}} \leq \frac{1}{\beta} \quad (94)$$

$$\beta \leq \frac{w_{2,k_1,l_1}}{w_{2,k_2,l_2}} \leq \frac{1}{\beta} \quad \text{where } \beta \in (0,1]$$

Once the inflection points satisfying these conditions found, the parameters of the possible rectangles can be estimated. Different from the circle case, the conditions are set for the horizontal and vertical profile features independently, hence the rectangle parameters are found from each profile, separately. From the vertical profile, $g_1[m]$, the side lengths, the orientation and the horizontal component of the center of the rectangle are estimated as given in Equation (95).

$$\hat{a}_1(i_1, i_2, j_1, j_2) = \frac{x_f(j_2) - x_f(j_1)}{2 \cos \theta_{1,j_1,i_2}}$$

$$\hat{b}_1(i_1, i_2, j_1, j_2) = \frac{0.5(\mu_{1,i_1,j_1} w_{1,i_2,j_2} + \mu_{1,i_2,j_2} w_{1,i_1,j_1})}{\tan(\hat{\theta}(i_1, i_2, j_1, j_2)) + 1} \quad (95)$$

$$\hat{\theta}_1(i_1, i_2, j_1, j_2) = \tan^{-1} \left(\frac{0.5(\sin \theta_{1,i_1,j_1} + \sin \theta_{1,i_2,j_2})}{\cos \theta_{1,j_1,i_2}} \right)$$

$$\hat{x}_{c,1}(i_1, i_2, j_1, j_2) = \frac{x_f(j_2) + x_r(i_1)}{2}$$

where

$$\begin{aligned}
\cos \theta_{1,j_1,i_2} &= \frac{2}{\mu_{1,j_1,i_2}} \\
\sin \theta_{1,i_1,j_1} &= \frac{2}{2\mu_{1,i_1,j_1} - \mu_{1,j_1,i_2}} \\
\sin \theta_{1,i_2,j_2} &= \frac{2}{2\mu_{1,i_2,j_2} - \mu_{1,j_1,i_2}} \\
\mu_{1,j_1,i_2} &= \frac{1}{x_r(i_2) - x_f(j_1)} \sum_{m=x_f(j_1)}^{x_r(i_2)} g_1[m]
\end{aligned} \tag{96}$$

Similarly, the rectangle parameters are estimated from $g_2[m]$ features as given below:

$$\begin{aligned}
\hat{a}_2(k_1, k_2, l_1, l_2) &= \frac{0.5(\mu_{2,k_1,l_1} w_{2,k_1,l_1} + \mu_{2,k_2,l_2} w_{2,k_2,l_2})}{\tan(\hat{\theta}(k_1, k_2, l_1, l_2)) + 1} \\
\hat{b}_2(k_1, k_2, l_1, l_2) &= \frac{y_f(l_2) - y_f(l_1)}{2 \cos \theta_{2,l_1,k_2}} \\
\hat{\theta}_2(k_1, k_2, l_1, l_2) &= \tan^{-1} \left(\frac{0.5(\sin \theta_{2,k_1,l_1} + \sin \theta_{2,k_2,l_2})}{\cos \theta_{2,l_1,k_2}} \right) \\
\hat{y}_{c,2}(k_1, k_2, l_1, l_2) &= \frac{y_f(l_2) + y_r(k_1)}{2}
\end{aligned} \tag{97}$$

where

$$\begin{aligned}
\cos \theta_{2,l_1,k_2} &= \frac{2}{\mu_{2,l_1,k_2}} \\
\sin \theta_{2,k_1,l_1} &= \frac{2}{2\mu_{2,k_1,l_1} - \mu_{2,l_1,k_2}} \\
\sin \theta_{2,k_2,l_2} &= \frac{2}{2\mu_{2,k_2,l_2} - \mu_{2,l_1,k_2}} \\
\mu_{2,l_1,k_2} &= \frac{1}{y_r(k_2) - y_f(l_1)} \sum_{n=y_f(l_1)}^{y_r(k_2)} g_2[n]
\end{aligned} \tag{98}$$

Lastly, *Axis Similarity*, is checked to understand whether the estimated rectangle parameters from both profiles are equivalent or not, which is expressed as below:

$$1 - \frac{1}{3} \left(\frac{|\hat{a}_1 - \hat{a}_2|}{\hat{a}_1 + \hat{a}_2} + \frac{|\hat{b}_1 - \hat{b}_2|}{\hat{b}_1 + \hat{b}_2} + \frac{|\hat{\theta}_1 - \hat{\theta}_2|}{\hat{\theta}_1 + \hat{\theta}_2} \right) \leq \lambda \tag{99}$$

where $\lambda \in (0,1]$.

When this problem is tried to be solved for whole scene, it becomes very costly due to the response of possible different shapes, noise and ambiguities caused by the existence of

many rectangles. Thus, we propose to detect the rectangles after determination of the connected components in the image as explained in Chapter 2.3.2. Besides, two thresholds, n_n and n_p , are defined to process at most n_n neighbor connected components, which are at most n_p pixels away from one of the components, together. The details of the proposed algorithm are given as given in Table 5.

Table 5 RDDI rectangle detection procedure

1. Extract edge map $E[m,n]$ of the image $I[m,n]$, where $E[m,n] \in \{0,1\}$.
2. Extract connected components, c_k 's, in $E[m,n]$.
3. Obtain the objects (or blobs), b_k 's, satisfying the following conditions:
 - It should be composed of at most n_n connected components
 - If there are more than one connected component in the object, for each of c_u there should be another connected component c_v having a distance at most n_p pixels, where the distances of the connected components are defined as below:

$$\|c_u - c_v\| = \min \left(\sqrt{|m_u - m_v|^2 + |n_u - n_v|^2} \right) \quad (100)$$

where $(m_u, n_u) \in c_u \quad (m_v, n_v) \in c_v$

4. For each b_k
 - 4.1. Find the orientation of the object, θ_k , by as given in Equation (83).
 - 4.2. Compute $g_1[m']$ as

$$g_1[m'] = \sum_{(m',n') \in b_k} E[m',n']$$

$$\text{where } \begin{bmatrix} m' \\ n' \end{bmatrix} = \begin{bmatrix} \cos \theta_k & -\sin \theta_k \\ \sin \theta_k & \cos \theta_k \end{bmatrix} \begin{bmatrix} m \\ n \end{bmatrix} \quad (101)$$

- 4.3. Compute $g_2[n']$ as

$$g_2[n'] = \sum_{(m',n') \in b_k} E[m',n']$$

$$\text{where } \begin{bmatrix} m' \\ n' \end{bmatrix} = \begin{bmatrix} \cos \theta_k & -\sin \theta_k \\ \sin \theta_k & \cos \theta_k \end{bmatrix} \begin{bmatrix} m \\ n \end{bmatrix} \quad (102)$$

- 4.4. Find rising and falling inflection points on $g_1[m']$ (x_r and x_f) and $g_2[n']$ (y_r and y_f).
- 4.5. Solve Equation (103) to find the matching hills on the profiles that satisfy the Peak Similarity, Width Similarity and Axis Similarity conditions to detect largest possible rectangle.

$$(i_{1r}, i_{2r}, j_{1r}, j_{2r}, k_{1r}, k_{2r}, l_{1r}, l_{2r}) = \arg \max_{(i_1, i_2, j_1, j_2, k_1, k_2, l_1, l_2)} \left\{ \hat{a}_1(i_1, i_2, j_1, j_2) * \hat{b}_2(k_1, k_2, l_1, l_2) \right\} \quad (103)$$

- 4.6. A rectangle is detected if a set $(i_{1r}, i_{2r}, i_{3r}, i_{4r}, j_{1r}, j_{2r}, j_{3r}, j_{4r})$ is found and the parameters of the rectangle is estimated as below:

$$a = 0.5(\hat{a}_1(i_1, i_2, j_1, j_2) + \hat{a}_2(k_1, k_2, l_1, l_2))$$

$$b = 0.5(\hat{b}_1(i_1, i_2, j_1, j_2) + \hat{b}_2(k_1, k_2, l_1, l_2))$$

$$\theta = \theta_i + 0.5(\hat{\theta}_1(i_1, i_2, j_1, j_2) + \hat{\theta}_2(k_1, k_2, l_1, l_2)) \quad (104)$$

$$x_c = \hat{x}_{c,1}(i_1, i_2, j_1, j_2)$$

$$y_c = \hat{y}_{c,2}(k_1, k_2, l_1, l_2)$$

3.3 Experimental Results

In this section, the results obtained by the proposed method are presented for synthetic and real world data. Moreover, the comparisons of the proposed method with the WHT algorithm are given in order to see the advantages and disadvantages of the method. All the methods are implemented in MATLAB R2009b and the experiments are done using same hardware configuration given in Chapter 2.5.

In order to test the robustness of the method against noise, a 256x256 image is created with 6 randomly located rectangles with randomly selected side lengths between 10 and 75. The orientations of the rectangles are also randomly selected between 0° and 90°. Then, “Salt & Pepper” type noise is added to the image at different levels where the noise level is evaluated using the formula given in Equation (67) and detected rectangles are analyzed to evaluate the probability of detection (P_D) and probability of false alarm (P_{FA}) which are already defined by Equation (68).

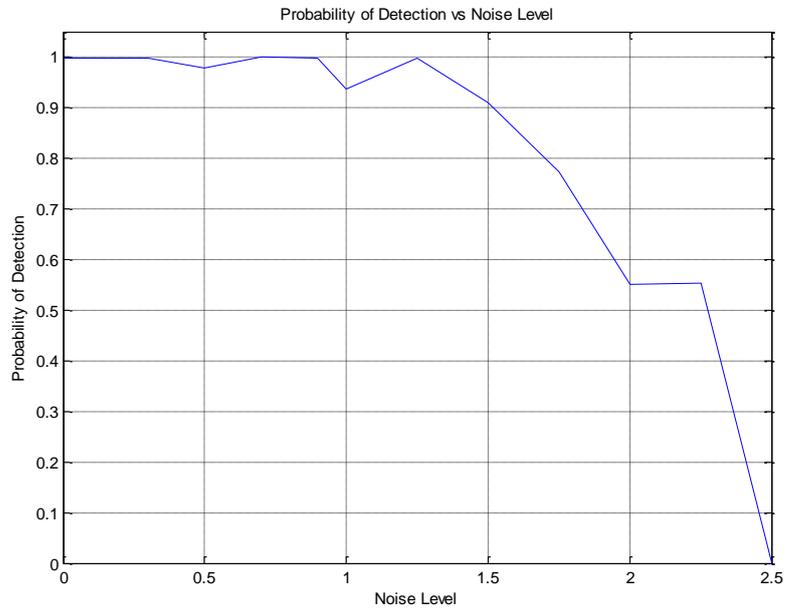


Figure 41 Detection probability of RDDI with respect to the added noise level.

In Figure 41, the change of P_D with the increase of the noise level is given. For this case, $\alpha=0.5$, $\beta=0.5$, $\lambda=0.5$, $n_n=0$, $n_p=0$. Note that, for each noise level the P_{FA} is evaluated below 0.1. An example of the test images, its edge map and the detected rectangle for this case are given in Figure 42 when noise level is 1.

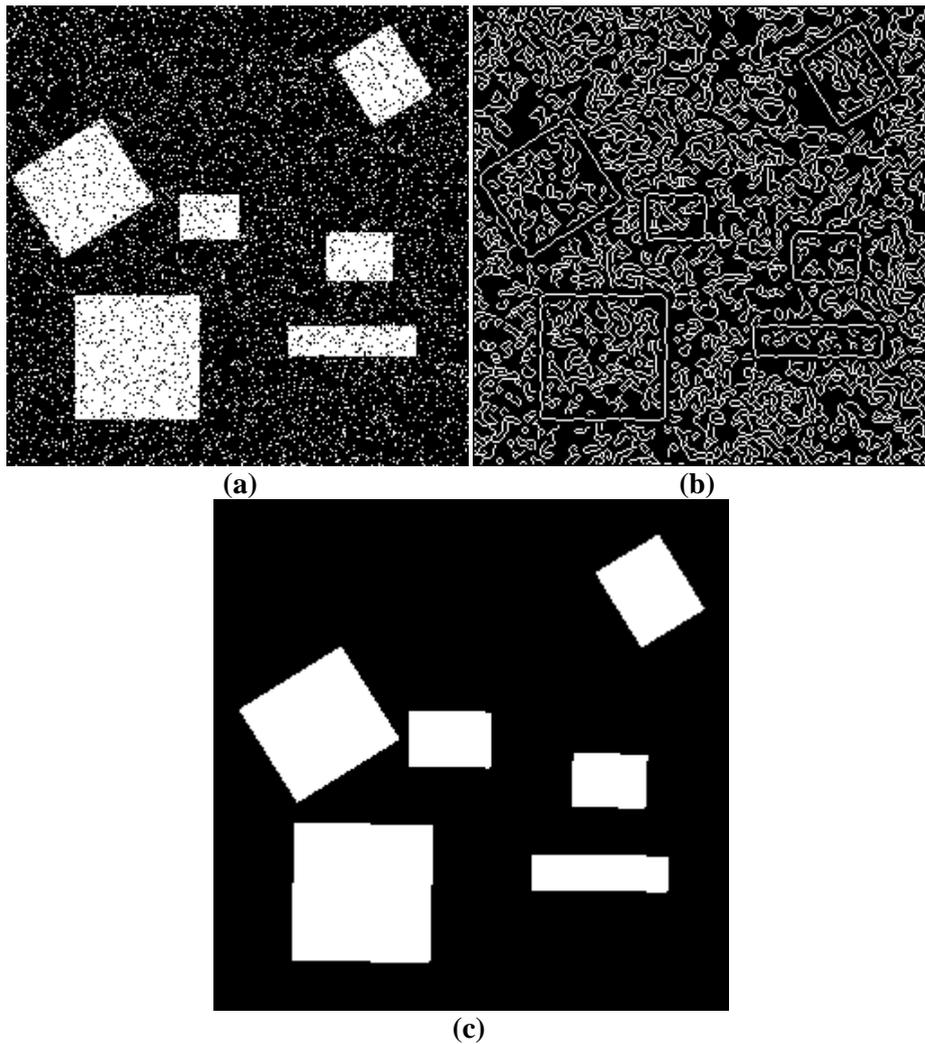


Figure 42 (a) An example for test images with Salt & Pepper noise added where $L_{\text{Noise}} = 1$, (b) edge map of the image with noise, (c) detected rectangles by RDDI.

The comparison of the proposed method with WHT is made in terms of the computation efficiency. For this purpose, a 1000x1000 image is created with randomly located rectangles with random side lengths between 10 and 75 and the number of rectangles is increased in order to analyze the computation efficiency of the approaches. The numbers of the rectangles are increased from 1 to 50 and the experiments are repeated for 100 times for each case. During the experiment, α , β and λ are set to 0.5, where n_n are n_p are set as 0 for RDDI; T_0 , T_p and T_L are equal to 1, 1 and 0.4 for WHT, respectively. However, during the evaluations of WHT method, the rectangles are searched in connected components rather than sliding windows in order to reduce the computation durations as in the circle detection case since the computation complexity is an exponent of the size of the image which makes the approach too slow to compare with the proposed method. Note that, for each case, P_D is above 99% and the P_{FA} is below 1%. Obtained results are given in Figure 43 and it can easily be seen that the complexity of the RDDI is less than the WHT hence the proposed method becomes much faster as the complexity of the scene (i.e. the number of edges) increases.

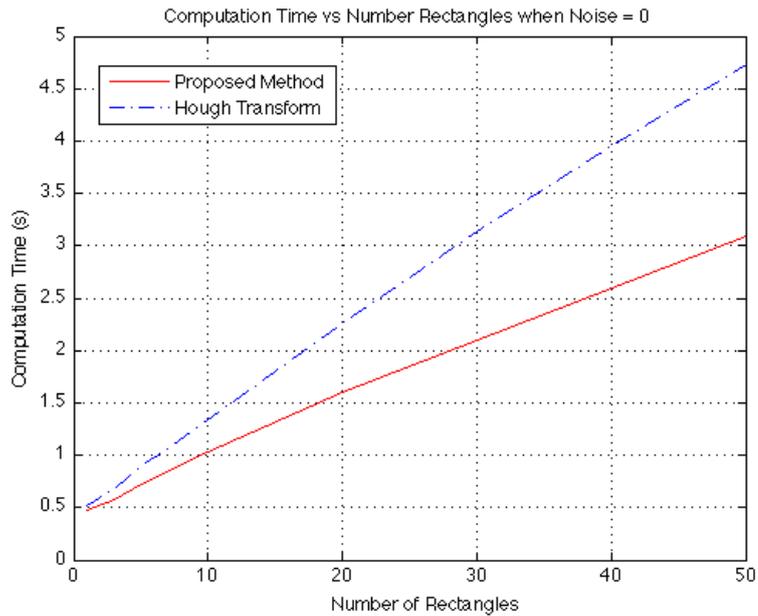


Figure 43 Computation durations of WHT and RDDI for different number of rectangles in the image.

The previous experiment is also performed for induced noise to simulate the performance of the methods for more complicated scenarios. Again Salt & Pepper type noise is added to the original image and the performance of the approaches are given in Figure 44 for different amount of noise. Note that, the parameters of the methods are kept same as in the previous experiment and the probabilities of detection of both approaches are above 98% where false alarm rates are below 20%.

The methods are also examined using real world data and the obtained results are presented from Figure 45 to Figure 47 and the durations of each approach for each of the images are given in Table 6. As seen from the results of the real world scenes and the computation durations given in Table 6, RDDI method has very similar detection performance with WHT approach, where it has less false positives and lower computation complexity. The main drawback of the proposed rectangle detection approach is its sensitivity to corrupted edges and can be seen in Figure 47 (d). The gray building in the middle of the image cannot be detected because of the missing edges on its left side. The white building on the upper and right side of the image cannot be detected since the shape is distorted from its lower side slightly.

Table 6 Execution durations of WHT in connected components and RDDI in seconds for different images.

		WHT CC	RDDI
Image Name	Image Size		
Car	500x358	4.237	3.696
Building	620x248	2.365	2.198
Industrial Region	988x689	17.751	16.364

Note that, during the experiment, α , β and λ are set to 0.5, where n_n are n_p are set as 0 for RDDI; T_θ , T_ρ and T_L are equal to 1, 1 and 0.2 for WHT, respectively. Again, WHT is computed for each of the connected components to reduce the computation complexity on the orders of RDDI.

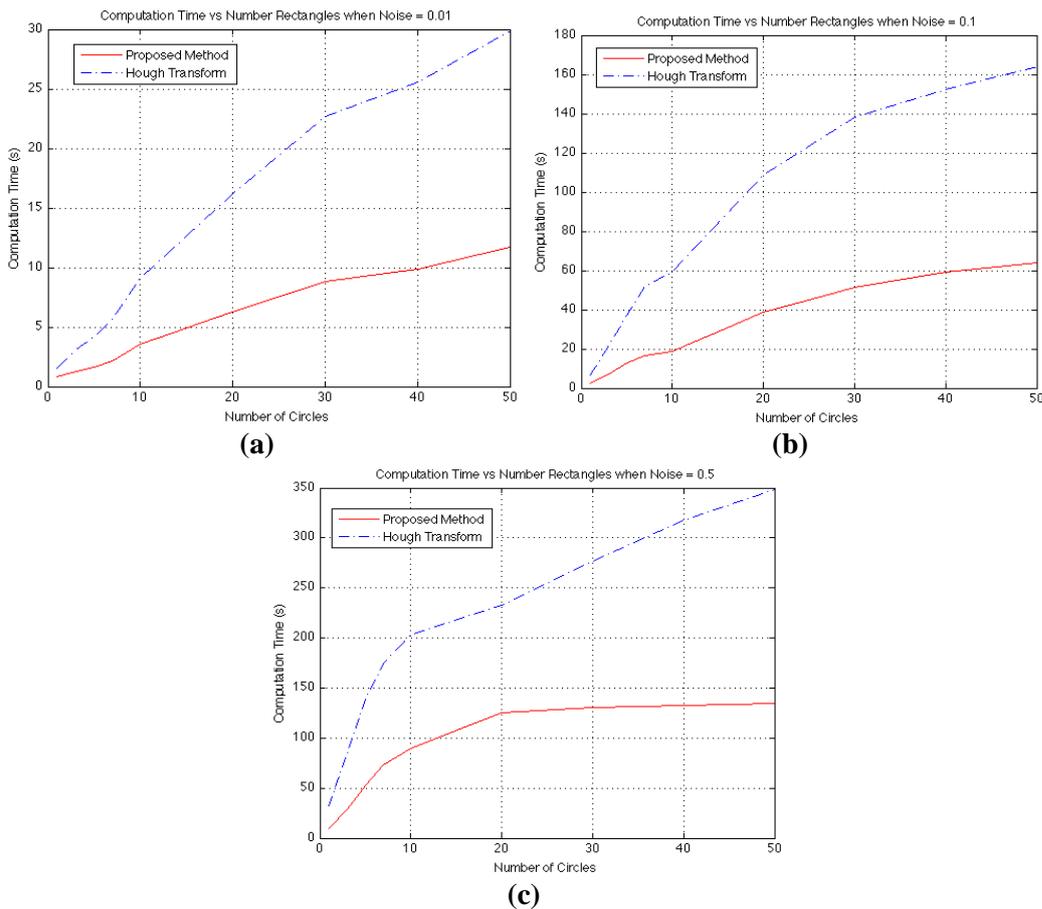


Figure 44 Computation durations of WHT when evaluated for each connected component and RDDI for different number of rectangles in the image when noise level is (a) 0.01, (b) 0.1 and (c) 0.5.

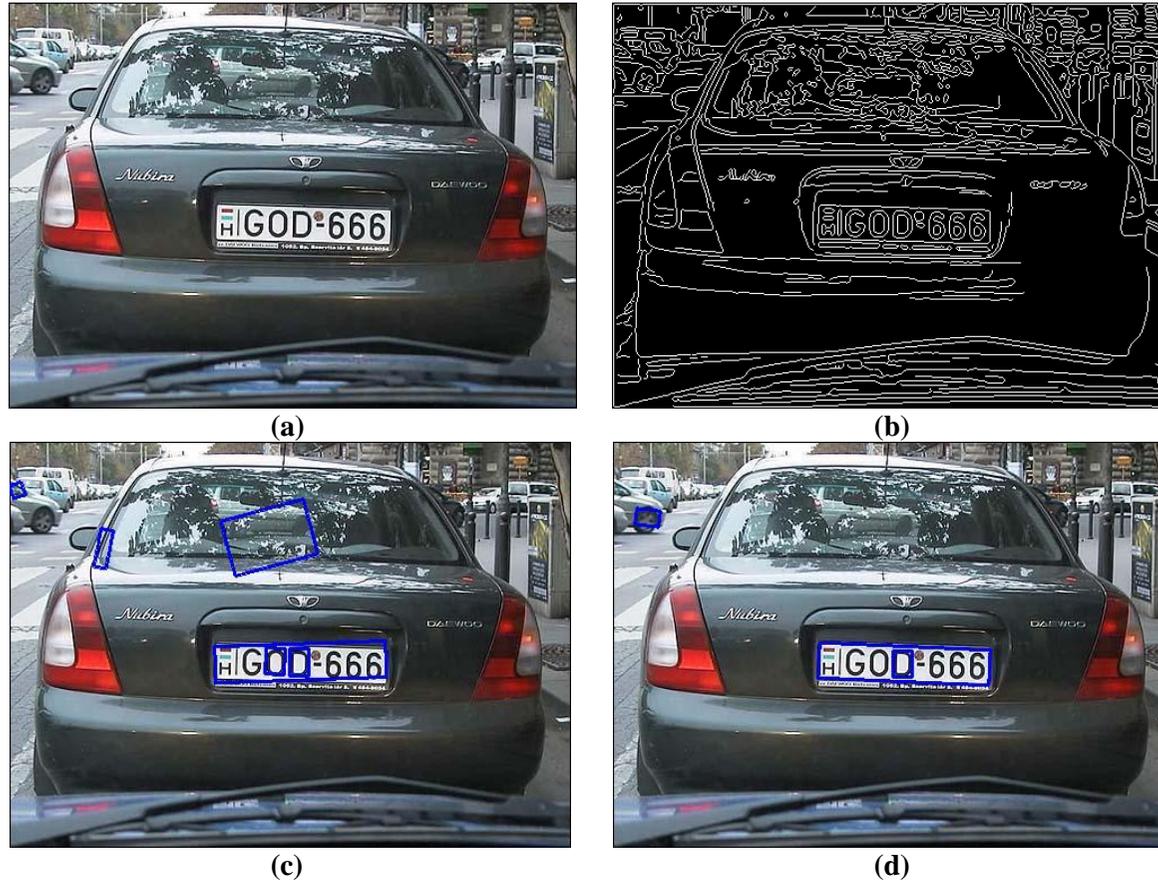


Figure 45 (a) A car image, (b) its edge map and rectangle detections using (c) WHT in CC and (d) RDDI.

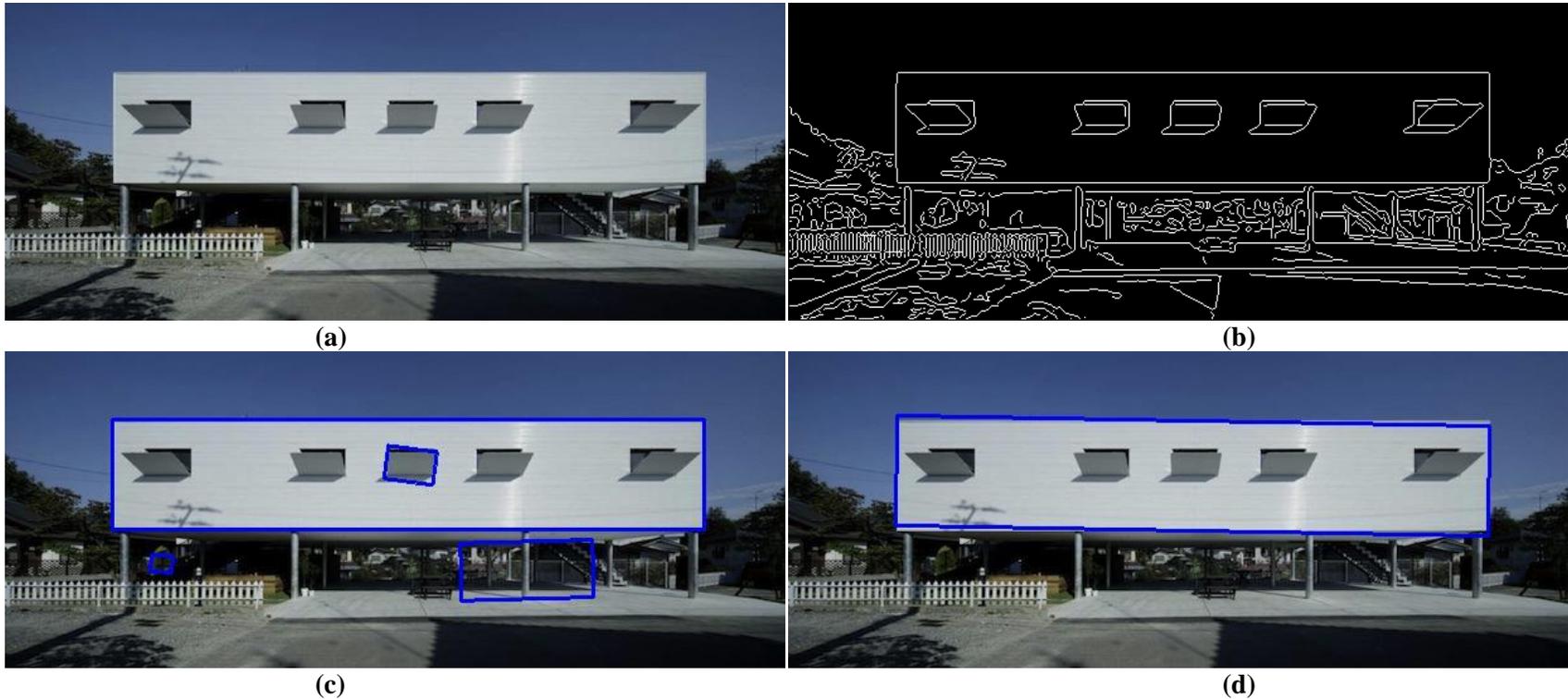
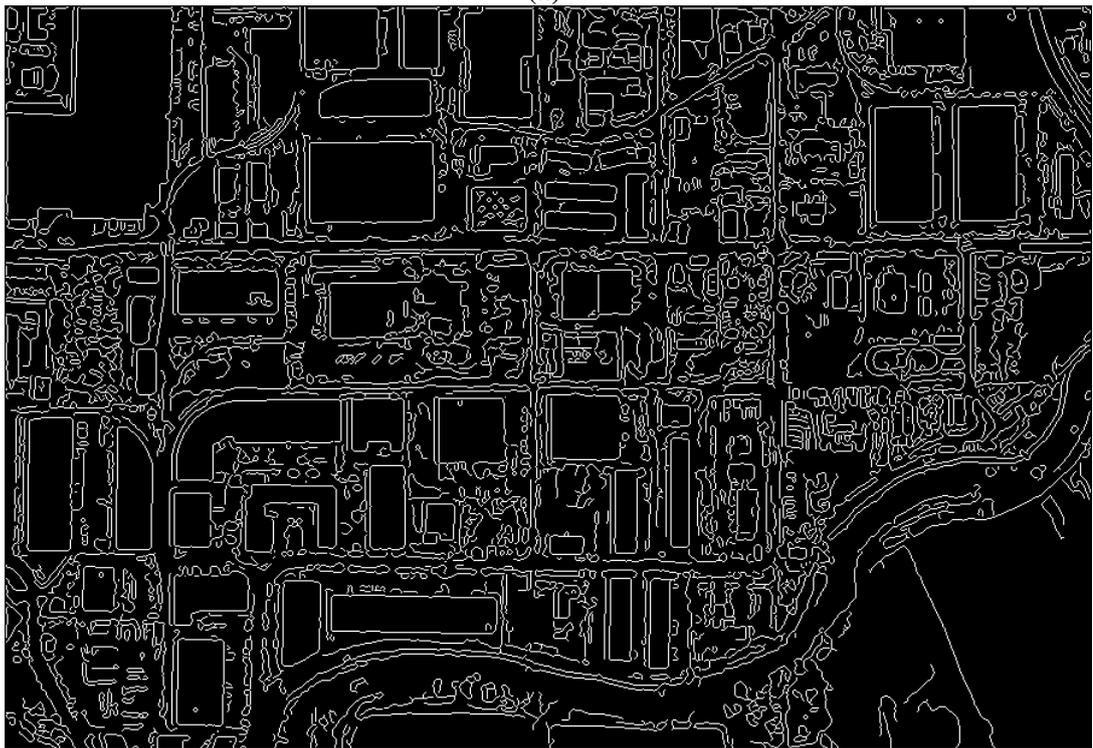


Figure 46 (a) A building image, (b) its edge map and rectangle detections using (c) WHT in CC and (d) RDDI.



(a)

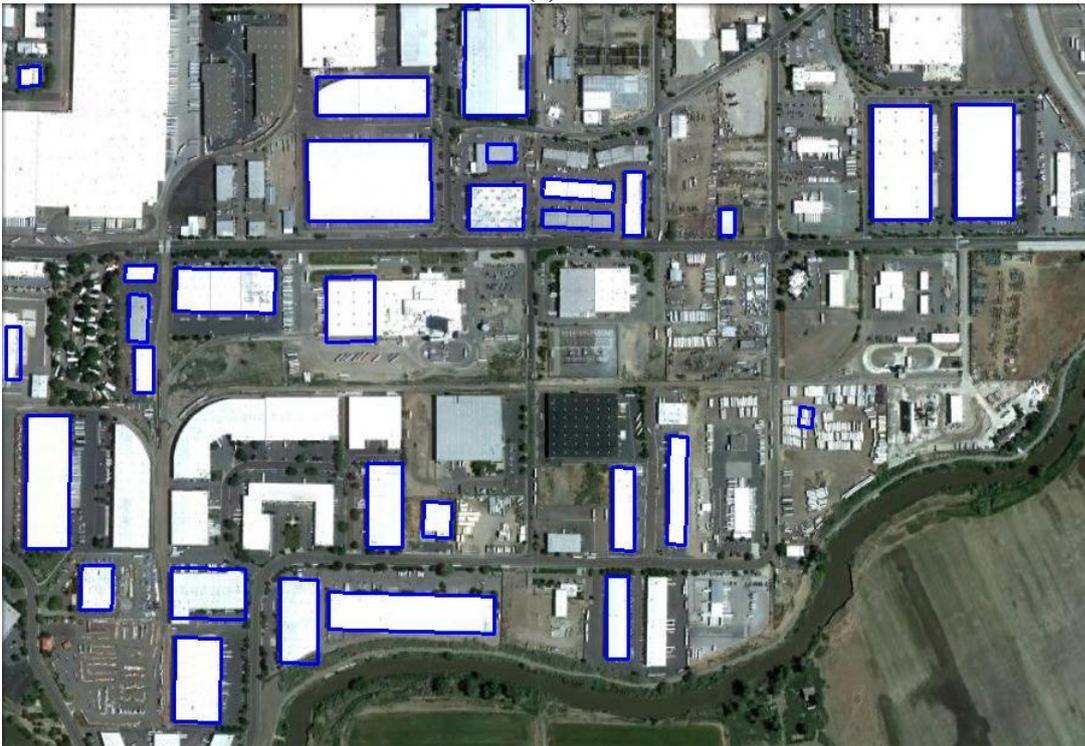


(b)

Figure 47 (a) A satellite image of an industrial zone taken from Google Earth, (b) its edge map and rectangle detections using (c) WHT in CC and (d) RDDI.



(c)



(d)

Figure 47 (cont'd) (a) A satellite image of an industrial zone taken from Google Earth, (b) its edge map and rectangle detections using (c) WHT in CC and (d) RDDI.

3.4 Parameter Selection for RDDI

RDDI has five parameters to adjust. α , β , are used to decide whether the shape of the examined object is a rectangle or not. As these parameters get close to 1, the algorithm tends to detect only the perfect rectangles. So, in order to handle the distorted rectangles, α and β , should be decreased. Since the parameters estimated from each profile should be similar λ should be close to 1. But this value can be decreased to detect distorted rectangles.

The last two parameters, n_n are n_p , are used to handle missing edges of the examined shapes. If the contrast of the objects are high enough and if unconnected objects are examined these parameters should be selected as 0. However, as the contrast gets lower or there exist connected objects these values should be increased.

As a guideline for the selection of these parameters, α , β , λ can be initialized from 0.75. This value is seen as an optimal value for detection of the rectangles in real life scenes when the contrast is high enough. If the results are not good enough, these values can be decreased, together. As these parameters define the similarity of the shape to a rectangle and each of them is defined in $(0,1]$, it is easier to set them to the same value. On the other hand, n_n are n_p depends only on the image. These values should be initialized as 0 and the results should be observed. If the image is noisy or the rectangles are intersects with other objects these values should be increased so it is better to analyze the edges on the images before changing these parameters. Since increasing the values of these parameters induces false detections and causes increase in the operation duration of RDDI, n_n and n_p values should be selected optimally.

3.5 Discussion

In this chapter a new rectangle detection approach is proposed. This approach has similarities with the circle detection approach explained in the previous chapter. The derivation of the method is given and the validity of the method is shown by the experiments. As in the circle detection case directional profiles of the edge map is evaluated and the detection is achieved by the analysis of the inflection points of these profiles. The experimental results performed using synthetic and real world data showed that the proposed method is robust against noise and. Hence, by use of this method, it is possible to reduce the computation time required while preserving the detection performance of Hough Transform based approaches. WHT is selected to be compared with and the experiments show that RDDI is much faster than this approach but the detection performances are similar. As in the circle case, the main drawback of the algorithm is its dependency to the quality of the extracted connected component.

CHAPTER 4

ELLIPSE DETECTION

Ellipse detection is another important problem in many of the applications of machine vision and pattern recognition some of which are face detection [57], iris recognition [58], remote sensing [59], [60].

As stated in the previous chapters, Hough Transform is a very popular approach once the closed form of the shape is obtained. By use of Generalized Hough Transform (GHT), it is possible to detect the ellipses on an image [8]. On the other hand, the complexity of this algorithm is very high and many different approaches are proposed to obtain more efficient ellipse detection method. Randomized Hough Transform (RHT) is one of the modifications made for GHT to detect ellipses faster as for circle detection [19, 20].

In the literature, most of the studies focused on decreasing the computation and memory requirements of the HT by reducing the accumulator dimension from five to less. In [61], the gradient direction of the edges are taken into account to reduce the memory requirements of the HT by decomposing the five dimensional parameter space into two independent accumulators and a histogram accumulator. In [62], a HT and Direct Least Square based ellipse detection approach is proposed starting from the approach given in [61]. The same two dimensional HT is used to guess the center locations of the ellipse but then the persistence of this location is analyzed by searching equidistant pair of edges around the possible center and the spurious points, which are too close to the center or too far from the center, are removed by the fitting a Normal distribution function to the Euclidian distances of the edge points to the center. Finally the remaining edge points are utilized to find the ellipse parameters by Direct Least Square. The synthetic and real images are utilized to analyze the performance of the method but this approach is only compared with the five dimensional HT based methods. Cheng and Liu proposed to reduce the number of edge pixels on the images to test in RHT approach by use of some preprocessing techniques [63]. This preprocessing is based on generating a linked list of the edge pixels according to their spatial coordinates and searching the ellipses on each of the list separately. Hence unnecessary edge pixels are not selected to be tested and the accuracy of the approach gets better by reducing the false alarms while decreasing the required computation.

In [64] a hybrid approach is proposed which combines the principles of hypothesis based and accumulator based approaches. In this method, first of all two edge pixels are selected assuming they are the pixels on the ellipse located on the major axis of this ellipse. The length of the minor axis is evaluated for each of the remaining edge pixels on the image assuming it is also on the same ellipse. The evaluated minor axis lengths are stored in an accumulator array and an ellipse is detected if the vote of a minor axis length exceeds a predefined threshold. Obviously, this process should be repeated for each edge pair on the image in order to find all the ellipses on the image. Despite the computation of the

approach is reduced by elimination of the tangent calculations this approach is still computationally intense for the complicated scenes and it may suffer for detection of the incomplete ellipses. Chia et. al. also proposed to use minor axis length as the only parameter of HT accumulator [65]. Hence, 1D accumulator is used for detection of the ellipses and the memory requirement of the method decreases a lot. The major difference of this method from the previous one is the refined decision strategy, which is defining a threshold according to the circumference of the ellipse rather than detection of the local peaks. This modification improves the performance of the method for detection of incomplete ellipses. Besides, in this study, the authors show how their algorithm can be parallelized and present the speed up after parallelization.

In [66], the symmetry property of the ellipses is used to reduce the dimension of the HT and the detection is achieved by a two dimensional HT. A similar approach is also utilized in [67], where the edge contours are grouped together according to some measures defined. The method is mainly composed of three main steps as extraction of smooth edges, generation of elliptical hypothesis and selection of salient hypothesis. Only the smooth edges are taken into consideration since the curvature cannot change abruptly on an ellipse. Then, for each of these smooth contours, other similar contours are determined in a search region and a two dimensional HT is applied for detection of the center of the elliptic hypothesis. Note that, this HT does not used for the extraction of the true parameters of the ellipse but for grouping the edge contours that possibly are the parts of the same ellipse. Finally, the ellipses on the image are detected using the selected groups of edge contours by the least squares approach. In this study, the method is compared with some other approaches in the literature using synthetic and real images and the presented results show that this approach has better detection performance but it is slower than some of the methods.

There are also some other ellipse detection approaches in the literature which has completely different perspective than the HT. Ji and Haralick are proposed to detect the ellipses by merging small arc segments according to some defined rules to form an ellipse [68]. Multisets mixture learning based detection is adopted for detection of ellipses in [69] and the advantages of the approach over RHT is shown by the experiments performed. Utilization of genetic algorithm is considered by Kawaguchi and Nagata [33] for detection of ellipses from the selected sets of line support regions, evaluating by the method in [70]. Nevertheless, these approaches are not analyzed deeply as the HT based approaches in order to understand their pros and cons except the lower computational load.

4.1 Ellipse Detection by Directional Integrations (EDDI)

In this chapter, the proposed circle and rectangle detection approaches are extended for detection of the ellipses. As explained in the previous chapter, the ellipse is defined using five parameters as shown in Figure 48 and it is expressed as given in Equation (105).

$$\frac{x'^2}{a^2} + \frac{y'^2}{b^2} = 1$$

$$\text{where } x' = (x - xc) \cos \theta - (y - yc) \sin \theta \quad (105)$$

$$y' = (x - xc) \sin \theta + (y - yc) \cos \theta$$

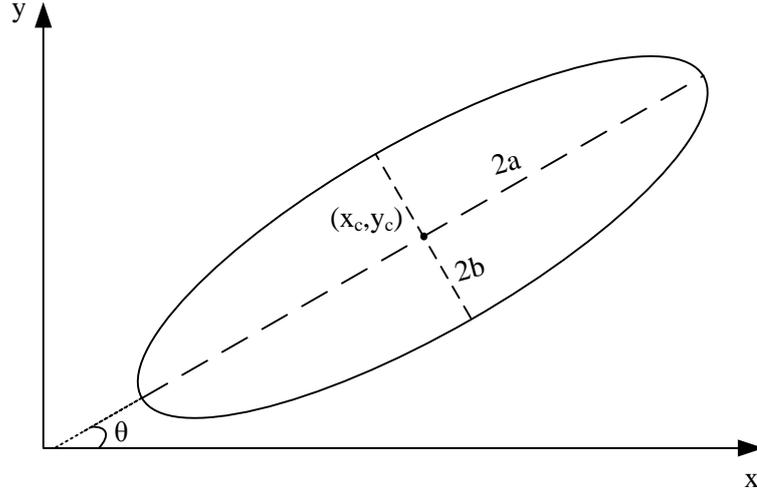


Figure 48 An ellipse and its parameters x_c, y_c, a, b and θ .

In order to simplify the derivations, let assume that there is only the edges of one ellipse, f , on the image where its center is at (x_c, y_c) and $\theta=0$. Then this image can be expressed by the following formula:

$$f(x, y) = \delta \left(\frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{b^2} - 1 \right) \quad (106)$$

As for the circle detection case, $g_1(x)$ is defined as in Equation (16) and it is evaluated as in Equation (18). Then, $g_1(x)$ is found as:

$$g_1(x) = 2 \sqrt{1 + \frac{b^2}{a^2} \frac{\left(\frac{x-x_c}{a}\right)^2}{1 - \left(\frac{x-x_c}{a}\right)^2}} \text{rect}\left(\frac{x-x_c}{2a}\right) \quad (107)$$

Similarly, $g_2(y)$ is found to be

$$g_2(y) = 2 \sqrt{1 + \frac{a^2}{b^2} \frac{\left(\frac{y-y_c}{b}\right)^2}{1 - \left(\frac{y-y_c}{b}\right)^2}} \text{rect}\left(\frac{y-y_c}{2b}\right) \quad (108)$$

In, $g_1(x)$ and $g_2(y)$ are plotted for $a=50, b=20$ where its center is at $(70, 100)$. From this figure and Equations (107) and (108), it can be easily seen that $g_1(x)$ is minimum at $x=x_c$ and has asymptotes at $x=x_c-a$ and $x=x_c+a$; $g_2(y)$ is minimum at $y=y_c$ and has asymptotes at $y=y_c-b$ and $y=y_c+b$. This result is very similar to the results obtained in circle case and the profiles have similar characteristics but they are not identical anymore.

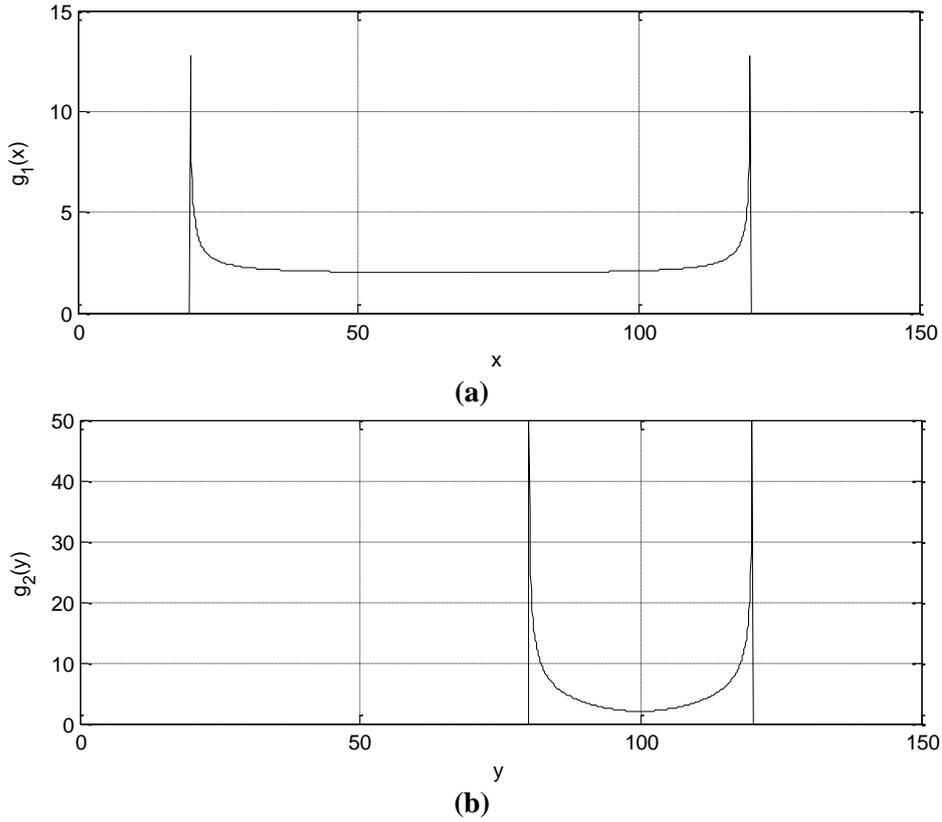


Figure 49 (a) $g_1(x)$ and (b) $g_2(y)$ for an ellipse with $a=50$, $b=20$ and centered at $(70, 100)$. Both x and y axes are sampled by 0.1 units.

While the orientation of the ellipse, θ , changes, the profiles differ a lot from the given in Equation (107) and (108) since both the distances between the asymptotes and the values of the functions in between change. The derivation of $g_1(x)$ and $g_2(y)$ and the solution of a , b and θ parameters are presented in APPENDIX A; however estimating the orientation of an ellipse as in Equation (83) is more robust against noise and shape disturbances. Hence, in this study, the orientation of the ellipse is estimated first and the profiles through this direction and its orthogonal direction are evaluated to find major and minor axis lengths using the relations given in Equations (107) and (108).

4.1.1 Analysis at Discrete Domain

For a discrete ellipse $f[m,n]$, $g_1[m]$ and $g_2[n]$ can be defined as in Equation (22). Then, for any m or n value, $g_1[m]$ and $g_2[n]$ are found by integrating $g_1(x)$ and $g_2(y)$ over 0.5 unit neighborhood as in Equation (109) assuming $\theta=0$.

$$g_1[m] = \int_{m-0.5}^{m+0.5} 2 \sqrt{1 + \frac{b^2}{a^2} \frac{\left(\frac{x-x_c}{a}\right)^2}{1 - \left(\frac{x-x_c}{a}\right)^2}} \text{rect}\left(\frac{x-x_c}{2a}\right) dx, \quad m \in Z$$

$$g_2[n] = \int_{n-0.5}^{n+0.5} 2 \sqrt{1 + \frac{a^2}{b^2} \frac{\left(\frac{y-y_c}{b}\right)^2}{1 - \left(\frac{y-y_c}{b}\right)^2}} \text{rect}\left(\frac{y-y_c}{2b}\right) dy, \quad n \in Z$$
(109)

Assuming x_c and a be integers, the maximum points of $g_1[m]$ are at $m=x_c-r$ and $m=x_c+r$.
When $m=x_c+r$:

$$g_1[x_c + a] = \int_{x_c+a-0.5}^{x_c+a+0.5} 2 \sqrt{1 + \frac{b^2}{a^2} \frac{\left(\frac{x-x_c}{a}\right)^2}{1 - \left(\frac{x-x_c}{a}\right)^2}} \text{rect}\left(\frac{x-x_c}{2a}\right) dx$$

$$= \int_{x_c+a-0.5}^{x_c+a} 2 \sqrt{1 + \frac{b^2}{a^2} \frac{\left(\frac{x-x_c}{a}\right)^2}{1 - \left(\frac{x-x_c}{a}\right)^2}} dx$$
(110)

By change of variable $t = \frac{x-x_c}{a}$ Equation (110) becomes:

$$g_1[x_c + a] = 2a \int_{1-\frac{1}{2a}}^1 \sqrt{1 + \frac{b^2}{a^2} \frac{t^2}{1-t^2}} dt$$
(111)

This equation can also be written as:

$$g_1[x_c + a] = 2\sqrt{b^2 - a^2} \int_{1-\frac{1}{2a}}^1 \sqrt{\frac{k^2 + t^2}{1-t^2}} dt,$$
(112)

$$\text{where } k^2 = \frac{a^2}{b^2 - a^2}$$

In Equation 3.169-4 of [71], this integral part of the equation is found to be

$$\int_u^1 \sqrt{\frac{k^2 + t^2}{1-t^2}} dt = \sqrt{k^2 + 1} E(\delta, r), \quad 1 > u \geq 0$$
(113)

where $E(\delta, r)$ is the elliptical function of the second kind, defined as:

$$E(\delta, r) = \int_0^{\sin \delta} \frac{\sqrt{1-rx^2}}{\sqrt{1-x^2}} dx \quad (114)$$

and

$$\begin{aligned} \delta &= \cos^{-1}(u) \\ r &= \frac{1}{\sqrt{1+k^2}} \end{aligned} \quad (115)$$

Then,

$$\begin{aligned} g_1[x_c + a] &= 2bE(\delta_1, r_1) \\ \text{where, } \delta_1 &= \cos^{-1}\left(1 - \frac{1}{2a}\right) \text{ and } r_1 = \sqrt{\frac{b^2 - a^2}{b^2}} \end{aligned} \quad (116)$$

When $m=x_c-a$

$$\begin{aligned} g_1[x_c - a] &= \int_{x_c-a-0.5}^{x_c-a+0.5} 2 \sqrt{1 + \frac{b^2}{a^2} \frac{\left(\frac{x-x_c}{a}\right)^2}{1 - \left(\frac{x-x_c}{a}\right)^2}} \text{rect}\left(\frac{x-x_c}{2a}\right) dx \\ &= \int_{x_c-a}^{x_c-a+0.5} 2 \sqrt{1 + \frac{b^2}{a^2} \frac{\left(\frac{x-x_c}{a}\right)^2}{1 - \left(\frac{x-x_c}{a}\right)^2}} dx \end{aligned} \quad (117)$$

By change of variable $t = \frac{x_c - x}{a}$ Equation (117) becomes:

$$\begin{aligned} g_1[x_c - a] &= -2a \int_1^{1-\frac{1}{2a}} \sqrt{1 + \frac{b^2}{a^2} \frac{t^2}{1-t^2}} dt \\ &= 2a \int_{1-\frac{1}{2a}}^1 \sqrt{1 + \frac{b^2}{a^2} \frac{t^2}{1-t^2}} dt \end{aligned} \quad (118)$$

Hence, $g_1[x_c-a]$ yields to same result with $g_1[x_c+a]$ (see Equation (111)) and the following expression can be obtained:

$$g_1[x_c - a] = 2bE(\delta_1, r_1) \quad (119)$$

Similar, operations can be repeated for the horizontal profile, $g_2[y]$, and the following results can be found for the maxima points:

$$\begin{aligned}
g_2[y_c - b] &= 2aE(\delta_2, r_2) \\
g_2[y_c + b] &= 2aE(\delta_2, r_2) \\
\text{where, } \delta_2 &= \cos^{-1}\left(1 - \frac{1}{2b}\right) \text{ and } r_2 = \sqrt{\left|\frac{a^2 - b^2}{a^2}\right|}
\end{aligned} \tag{120}$$

A synthetically generated ellipse can be seen in Figure 50 (a), where the image is 150x150, the ellipse is centered at (70, 100) with the parameters $a=50$, $b=20$. When the vertical and horizontal profiles are evaluated on this image resultant $g_1[m]$ and $g_2[n]$ can be seen in Figure 50 (b) and (c), respectively. These two graphs are symmetric around the x and y axis components of the ellipse's center. For $g_1[m]$ the distance between maxima is 100 ($=2a$) units and their values are equal to 5 ($2bE(\delta_1, r_1) = 5.6181$). For $g_2[n]$ the distance between maxima is 40 ($=2b$) units and their values are equal to 23 ($2aE(\delta_2, r_2) = 22.2362$). Like the circle case, there are lots of indices having a value of 2 around the symmetry axes of the functions and there are some locations having lower or higher values than the expectations due to the discretization.

In Figure 51 (a), a speaker image is presented with its edge map given in Figure 51 (b). The circle given in Figure 51 (c) is extracted manually from this edge map to evaluate the horizontal and vertical profiles, $g_1[m]$ and $g_2[n]$, as given in Figure 51 (e) and (f), respectively. When $g_1[m]$ is examined, its local peaks are found at $m=12$ and $m=170$. This result yields the x component of its center, x_c , to be at 91 and major axis length (a) to be 79. When $g_2[n]$ is examined the peaks are found at $n=48$ and $n=126$, which yields the y component of its center, y_c , to be at 87 and the minor axis length (b) to be 39. Then, the estimated values of the peaks of $g_1[m]$ and $g_2[n]$ should be 8.7476 and 25.2332 using the Equations (116), (119) and (120) and these values are observed as 10 and 23 from the given graphs. The consistency between the estimated and observed values validates the approach which can be also seen by the drawn ellipse in Figure 51 (d) with the parameters estimated from the graphs.

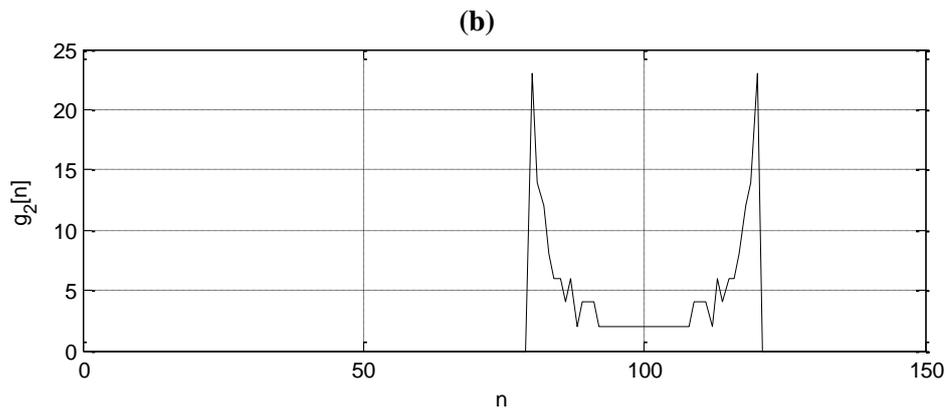
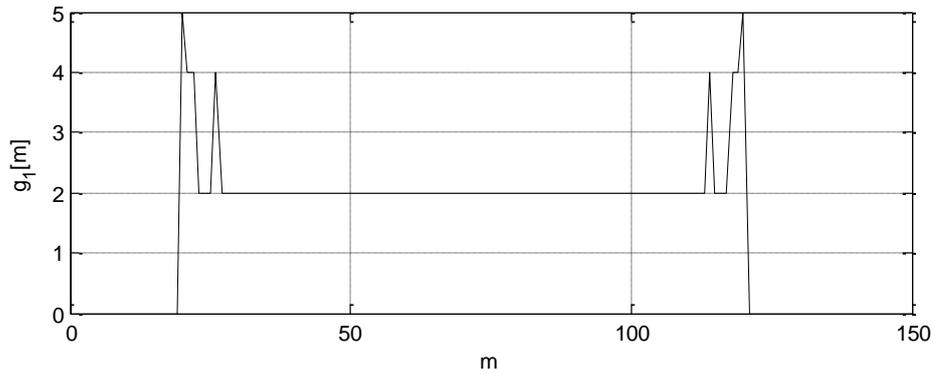
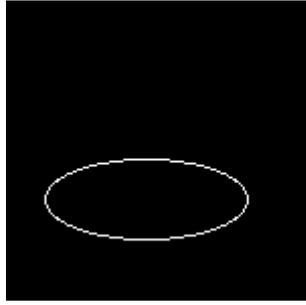


Figure 50 (a) A synthetic ellipse image with parameters $a=50$, $b=20$, centered at (70, 100) and evaluated directional summations (b) $g_1[m]$ and (c) $g_2[n]$ from this image.

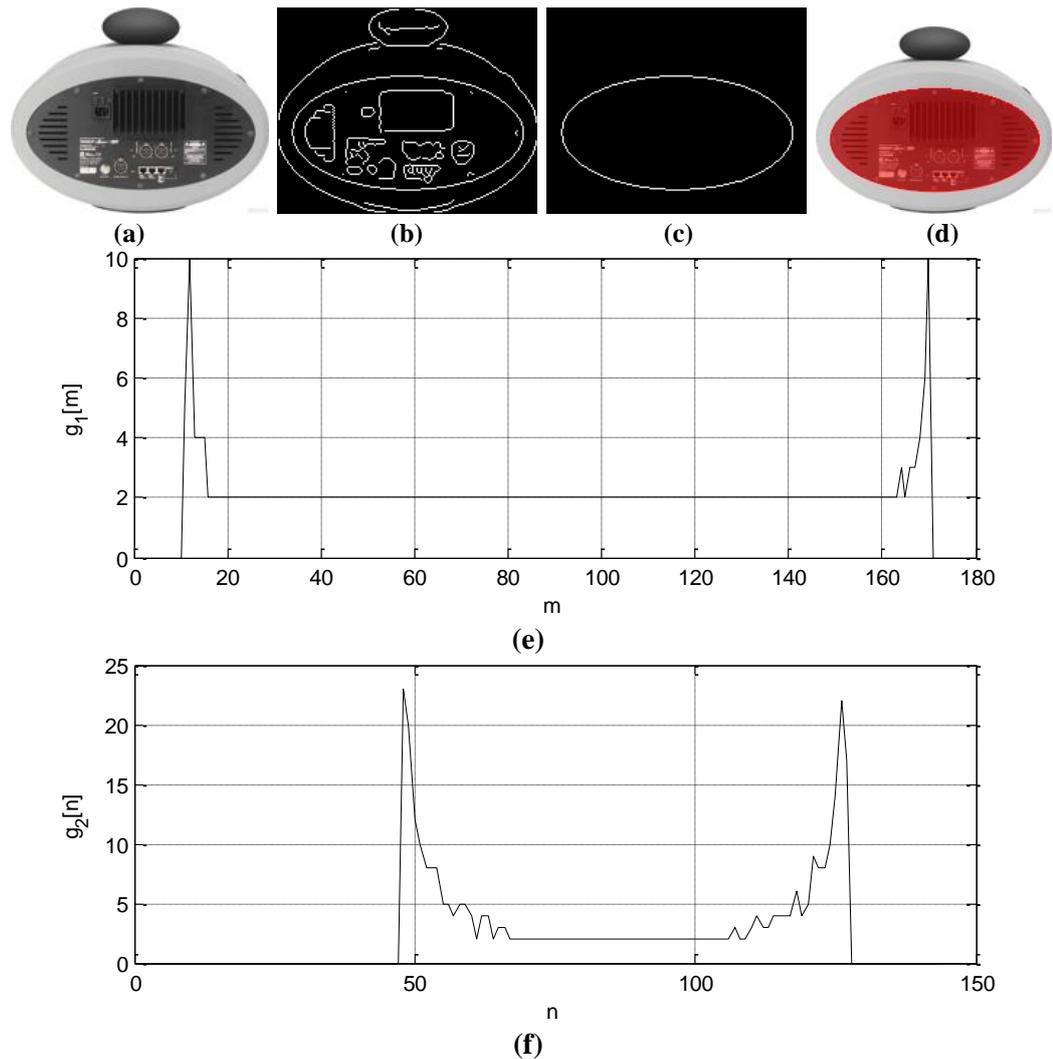


Figure 51 (a) A speaker image, (b) its edge map using Canny detector, (c) manually extracted elliptical structure from the edge map (d) estimated ellipse from (e) vertical profile, $g_1[m]$, and (f) horizontal profile $g_2[n]$.

Another real life example is given for a dish antenna in Figure 52 (a), where the ellipse in this case is rotated. As stated in the previous section, in this study the orientation of the ellipse is estimated first by Equation (83) and the profiles are evaluated accordingly. In Figure 52 (b) the edge map of the image and in Figure 52 (c) manually extracted and rotated (9.4°) edge contour of one of the antennas are given. When the profiles given in Figure 52 (e) and (f) are analyzed, the local peaks of $g_1[m]$ are found at $m=273$ and $m=338$, with values of 8 and 4; the local peaks of $g_2[n]$ are found at $n=95$ and $n=115$, with values of 13 and 19, respectively. By use of these information, an ellipse with major axis length $a=32.5$, minor axis length $b=10$ and orientation $\theta=9.4$ can be drawn as seen in Figure 52 (d). According to Equations (116), (119) and (120), the peaks of $g_1[m]$ and $g_2[n]$ should be 3.41 and 20.31 which are different from the observed values due to the distortion on the observed shape. However, it is still possible to find an ellipse from those profiles, which fits with the shape, despite deformations on the elliptical shape.

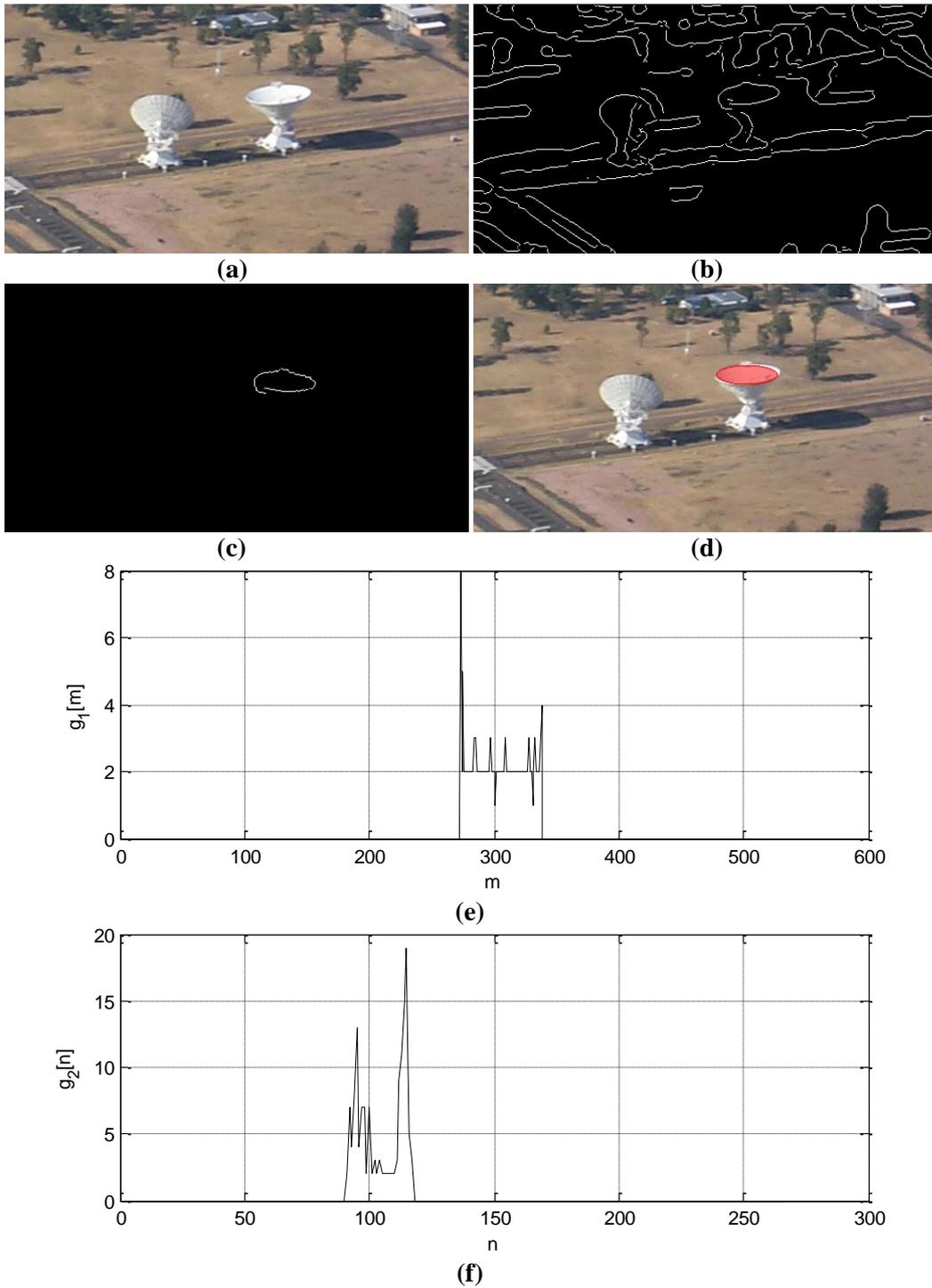


Figure 52 (a) A sample image containing a dish antenna, (b) its edge map using Canny detector, (c) manually extracted and rotated elliptical structure from the edge map (d) estimated ellipse from (e) vertical profile, $g_1[m]$, and (f) horizontal profile $g_2[n]$.

4.1.2 Proposed Ellipse Detection Algorithm

In the previous section, it is shown that information about the parameters of an ellipse can be extracted from its horizontal and vertical profiles. According to the obtained results, there has to be two peaks in both of the profiles, and these peak pairs in each profile have to possess similar characteristics. Therefore, in this study, the problem of detecting ellipses is reduced to identifying matching peak pairs in the horizontal and vertical profiles of the ellipse as in the circle case.

However, the derivations in the previous section are for an ellipse with orientation of 0° . Then the first step of the approach is estimating the orientation of the ellipse, θ_r , as in explained in Chapter 3.2.1 by Equation (83). Then the ellipse is rotated by this amount back to evaluate the horizontal and vertical profiles, $g_1[m]$ and $g_2[n]$, as in Equation (22).

Equations (116)-(120) imply existence of only two peaks in each profile of the ellipse, which are also identical to each other. However, discretization, distorted shapes or noise on the image may cause other peaks to appear and/or desired peaks to be different from each other. So, a two step process, very similar to CDDI method, is proposed to detect the matching peaks in the profiles. The first step is finding the global maxima of $g_1[m]$ and $g_2[n]$, $g_{1,\max}$ and $g_{2,\max}$, respectively, by:

$$\begin{aligned} m_{\max} &= \arg \max_m \{g_1[m]\} \\ g_{1,\max} &= g_1[m_{\max}] \end{aligned} \quad (121)$$

$$\begin{aligned} n_{\max} &= \arg \max_{n_{\max}} \{g_2[n]\} \\ g_{2,\max} &= g_2[n_{\max}] \end{aligned} \quad (122)$$

In the second step, features related to each local peak, m_i and n_j , in $g_1[m]$ and $g_2[n]$, given in Equations (123)-(130).

$$p_{1,i} = g_1[m_i] \quad (123)$$

$$d_{1,i} = |m_i - m_{\max}| \quad (124)$$

$$b_{1,i} = L \left(\frac{d_{1,i}}{2}, g_{1,\max} \right) \quad (125)$$

$$\mu_{1,i} = \frac{1}{d_{1,i}} \sum_{m=\min\{m_i, m_{\max}\}}^{\max\{m_i, m_{\max}\}} g_1[m] \quad (126)$$

$$p_{2,j} = g_2[n_j] \quad (127)$$

$$d_{2,j} = |n_j - n_{\max}| \quad (128)$$

$$b_{2,j} = L \left(\frac{d_{2,j}}{2}, g_{2,\max} \right) \quad (129)$$

$$\mu_{2,j} = \frac{1}{d_{2,j}} \sum_{n=\min\{n_j, n_{\max}\}}^{\max\{n_j, n_{\max}\}} g_2[n] \quad (130)$$

where $p_{1,i}$'s and $p_{2,j}$'s denote the values of local peaks of $g_1[m]$ and $g_2[n]$; $d_{1,i}$'s and $d_{2,j}$'s denote the distances between the local peaks and global maxima of $g_1[m]$ and $g_2[n]$; $b_{1,i}$ and $b_{2,j}$ are the estimated vertical and horizontal axes lengths of the ellipse from $g_1[m]$ and $g_2[n]$; $\mu_{1,i}$'s and $\mu_{2,j}$'s denote the average values of $g_1[m]$ and $g_2[n]$ in between their global maxima and peaks and L is a look up table such that

$$\begin{aligned} L(a, p) &= b \\ p &= 2bE(\delta, r) \end{aligned} \quad (131)$$

$$\text{where } \delta = \cos^{-1}\left(1 - \frac{1}{2a}\right), \quad r = \sqrt{\frac{b^2 - a^2}{a^2}}$$

Based on Equations (116)-(120), if there is an ellipse in the image, for at least one of the local peak in each profile, the following conditions should be satisfied:

- **Peak Similarity:** The value of the local peaks in $g_1[m]$ and $g_2[n]$ are close to the global maxima of that profile. Then,

$$\begin{aligned} \frac{p_{1,i}}{p_{1,\max}} &\geq \alpha \\ \frac{p_{2,j}}{p_{2,\max}} &\geq \alpha \quad \text{where } \alpha \in (0, 1] \end{aligned} \quad (132)$$

- **Mean Validity:** The average values of $g_1[m]$ and $g_2[n]$ between the matching peak pairs is related to the circumference of the ellipse, which is computed by a complete elliptical integral of second kind. This integral can only be evaluated numerically but Ramanujan approximated the circumference of the ellipse by Equation (133) [72] and this approximation is used in this study to have an efficient implementation.

$$C_e(a, b) \approx \pi(a+b) \left(1 + \frac{3h^2}{10 + \sqrt{4 - 3h^2}} \right), \quad (133)$$

$$\text{where } h = \frac{a-b}{a+b},$$

a: major axis length

b: minor axis length

Then, $\mu_{1,i}$ and $\mu_{2,j}$ should satisfy:

$$\beta \leq \frac{d_{1,i} \mu_{1,i}}{C_e(0.5d_{1,i}, b_{1,i})} \leq \frac{1}{\beta}$$

$$\beta \leq \frac{d_{2,j} \mu_{2,j}}{C_e(0.5d_{2,j}, b_{2,j})} \leq \frac{1}{\beta} \quad \text{where } \beta \in (0,1]$$
(134)

- Axis Similarity: The estimated circle parameters from each profile should be similar. Then,

$$\lambda \leq \frac{0.5d_{1,i}}{b_{2,j}} \leq \frac{1}{\lambda}$$

$$\lambda \leq \frac{0.5d_{2,j}}{b_{1,i}} \leq \frac{1}{\lambda} \quad \text{where } \lambda \in (0,1]$$
(135)

Once the local peaks satisfying all the conditions above, m_i and n_j , are detected, the parameters of the circle can be estimated by:

$$x_c = \frac{m_{\max} + m_i}{2}$$

$$y_c = \frac{n_{\max} + n_j}{2}$$

$$a = \frac{|m_{\max} - m_i|}{2}$$

$$b = \frac{|n_{\max} - n_j|}{2}$$

$$\theta = \theta_r$$
(136)

Note that, since many peak pairs may appear that satisfy these conditions, we propose to get the most distant pairs in each profile.

When this problem is tried to be solved for whole scene, it becomes very costly due to the response of possible different shapes, noise and ambiguities caused by the existence of many ellipses. Thus, we propose to detect the ellipses after determination of the connected components in the image. Besides, two thresholds, n_n and n_p , are defined to process at most n_n neighbor connected components, which are at most n_p pixels away from one of the components, together. Then, the overall procedure for the proposed method is given in Table 7.

Table 7 EDDI ellipse detection procedure

<p>1. Extract edge map $E[m,n]$ of the image $I[m,n]$, where $E[m,n] \in \{0,1\}$.</p> <p>2. Extract connected components, c_k's, in $E[m,n]$.</p> <p>3. Obtain the objects (or blobs), b_k's, satisfying the following conditions:</p> <ul style="list-style-type: none"> • It should be composed of at most n_n connected components • If there are more than one connected component in the object, for each of c_u there should be another connected component c_v having a distance at most n_p pixels, where the distances of the connected components are defined as below: $\ c_u - c_v\ = \min \left(\sqrt{ m_u - m_v ^2 + n_u - n_v ^2} \right) \quad (137)$ <p style="text-align: center;">where $(m_u, n_u) \in c_u$ $(m_v, n_v) \in c_v$</p> <p>4. For each b_k</p> <p>4.1. Find the orientation the object, θ_k, as given in Equation (83) using the edge points $(m,n) \in b_k$.</p> <p>4.2. Compute $g_1[m']$ as in Equation (101).</p> <p>4.3. Compute $g_2[n']$ as in Equation (102).</p> <p>4.4. Detect the global maximum points in $g_1[m']$ (m_{max}) and $g_2[n']$ (n_{max}).</p> <p>4.5. Detect the local maximum points in $g_1[m']$ (m_i's) and $g_2[n']$ (n_j's).</p> <p>4.6. Solve Equations (138) and (139) to find the matching peaks on the profiles that satisfy the "Peak Similarity", "Mean Validity" and "Axis Similarity" conditions to detect the largest possible ellipse.</p> $i_r = \arg \max_i \{ m_{max} - m_i \} \quad (138)$ $j_r = \arg \max_j \{ n_{max} - n_j \} \quad (139)$ <p>4.7. An ellipse is detected if i_r and j_r are found and the parameters of the circle are estimated using Equation (136).</p>
--

4.2 Experimental Results

In this section, the results obtained by the proposed ellipse detection method are presented for synthetic and real world data. All the methods are implemented in MATLAB R2009b and the experiments are done using same hardware configuration given in Chapter 2.5.

In order to test the robustness of the method against noise, a 256x256 image is created with 6 randomly located ellipses with randomly selected minor and major axis length between 20 and 100; with randomly selected orientation between 0° and 90°. Then, "Salt & Pepper" type noise is added to the image at different levels where the noise level is evaluated using the formula given in Equation (67) and detected ellipses are analyzed to evaluate the

probability of detection (P_D) and probability of false alarm (P_{FA}) which are already defined by Equation (68).

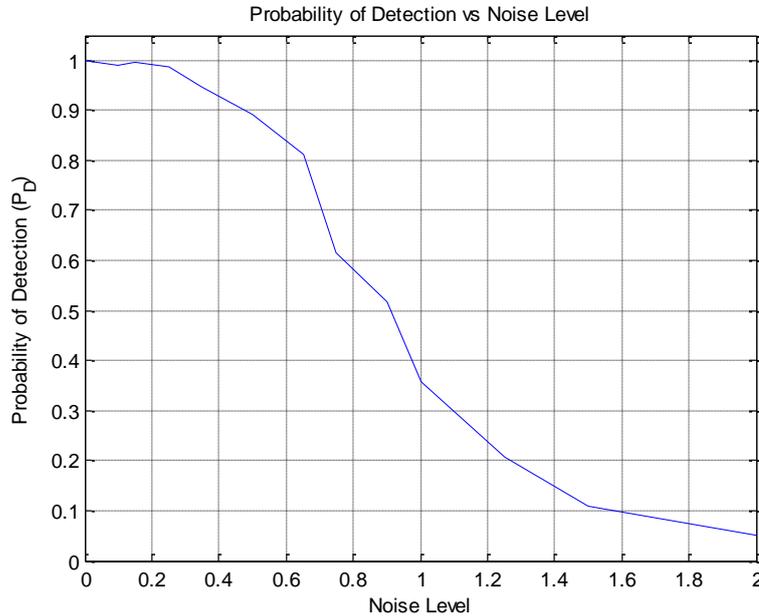
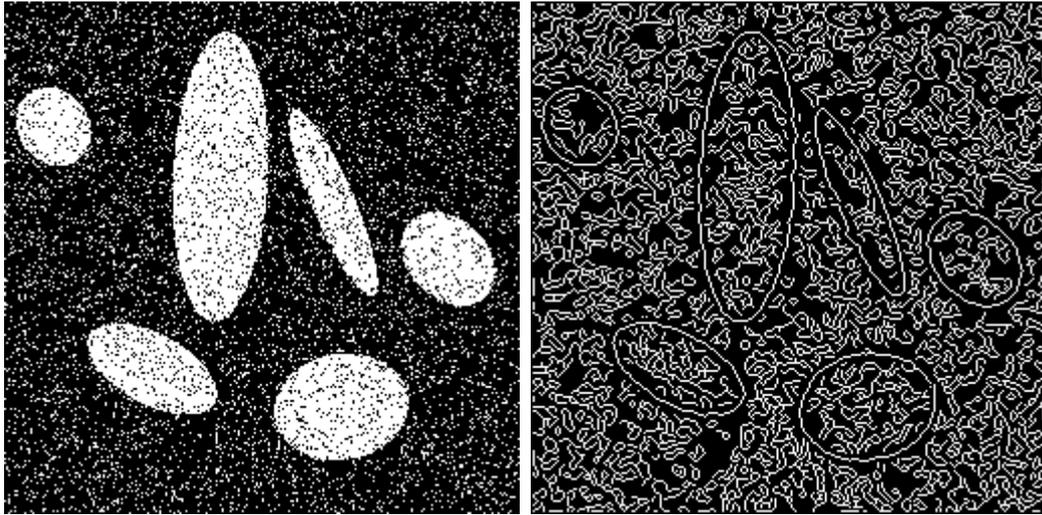


Figure 53 Detection probability of EDDI with respect to the added noise level.

Figure 53 the change of P_D with the increase of the noise level is given. For this case, $\alpha=0.5$, $\beta=0.7$, $\lambda=0.5$, $n_n=0$, $n_p=0$. Note that, for each noise level the P_{FA} is evaluated below 0.02. An example of the test images, its edge map and the detected ellipses using EDDI are given in Figure 54 when noise level is 1. As seen from this figure, all the ellipses can be detected by the approach in this case but the probability of detection of the approach is about 0.4 for $L_{Noise}=1$.

The proposed ellipse detection method is also analyzed in terms of the computation efficiency. For this purpose, a 1000x1000 image is created with randomly located ellipses with random major and minor axis lengths between 20 and 200; random orientation between 0° and 90° ; and the number of ellipses is increased from 1 to 50 in order to analyze the computation efficiency of the approach. The experiments are repeated for 100 times for each case. During the experiments, α , β and λ are set to 0.5, where n_n and n_p are set as 0 for EDDI. Note that, for each case, P_D is above 99% and the P_{FA} is below 2%. Obtained results are given in Figure 55 and it can easily be seen that the complexity of the EDDI is logarithmically increases as the complexity of the scene (i.e. the number of edges) increases. It is an evident for this approach to detect ellipses in reasonable durations even for complex scenes. This experiment is also performed for noisy cases and the results are presented in Figure 56. These results also show the convergence of the computational complexity of the method.



(a) (b)



(c)

Figure 54 (a) An example for test images with Salt & Pepper noise added where $L_{\text{Noise}} = 1$, (b) edge map of the image with noise, (c) detected ellipses by EDDI.

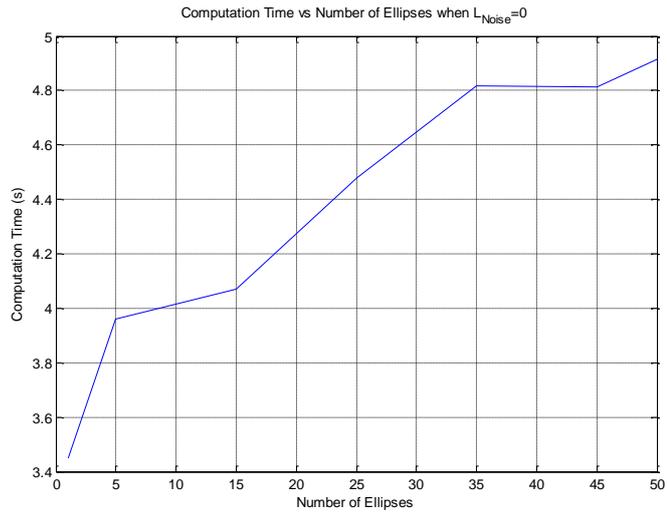


Figure 55 Computation duration of EDDI for different number of ellipses in the image.

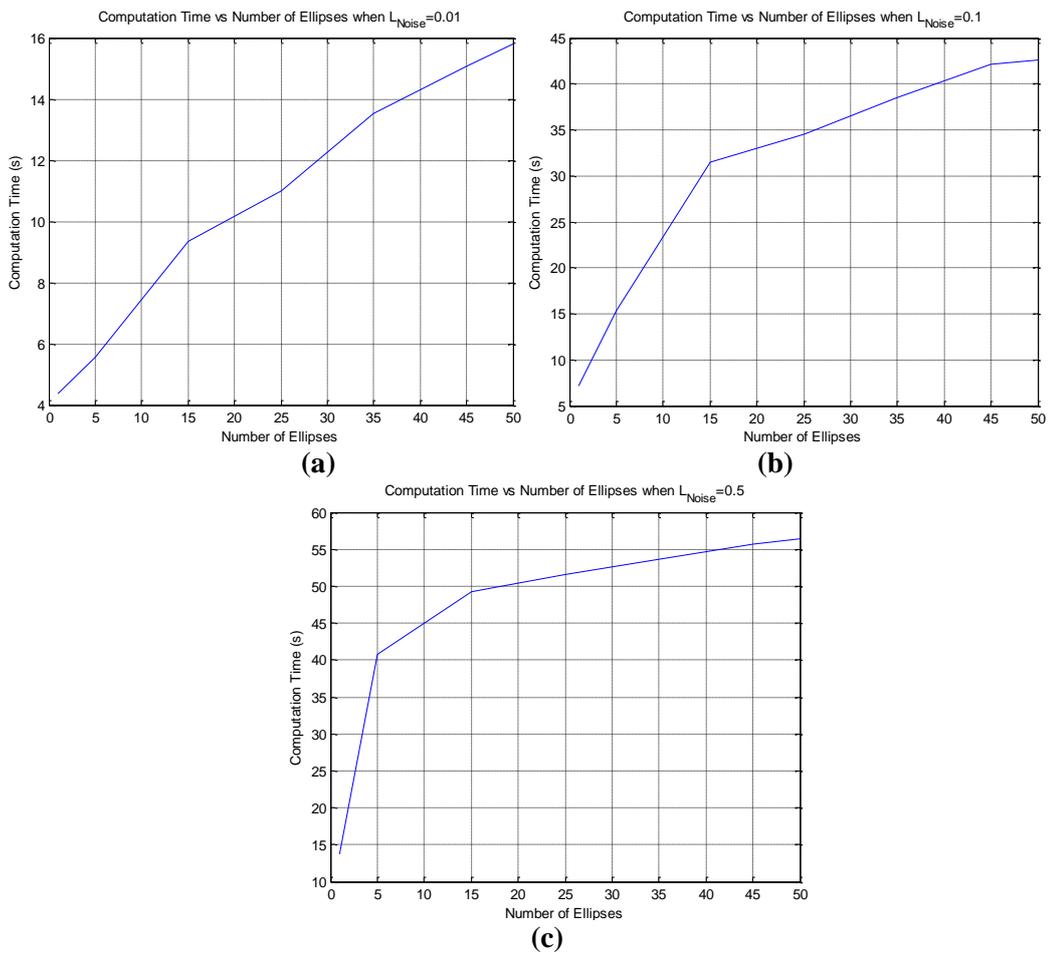


Figure 56 Computation duration of EDDI for different number of ellipses in the image when noise level is (a) 0.01, (a) 0.1 and (c) 0.5.

EDDI approach is also examined using real world data and the obtained results are presented from Figure 57 to Figure 59 and the durations for each of the images are given in Table 8. The first two examples are automatically detection of the ellipse given in Figure 51 and Figure 52, respectively. As seen from the result in Figure 57, a very similar ellipse is detected with the manual case given Figure 51 (d) with a small orientation error. The results given in Figure 52 (d) and Figure 58 are also very close to each other and for both cases the dish antenna can be detected as an ellipse. Note that, each of the image is filtered by a Gaussian bilateral filter, [44], before detection of the edges with different parameters according to their characteristics. This filter has three parameters as the size of the filter, standard deviation in spatial domain and standard deviation in intensity domain. For “Speaker” image; these parameters are selected as 15, 7 and 1, respectively. They are selected as 7, 5 and 1 for “Dish Antenna” image and 11, 7 and 2 for “Garden” image. On the other hand, for all of the images the EDDI parameters α , β and λ are set to 0.5, where n_n are set as 0.

Table 8 Execution durations of EDDI in seconds for different images.

		EDDI
Image Name	Image Size	
Speaker	180x143	3.374
Dish Antenna	507x266	5.163
Garden	1280x717	32.536

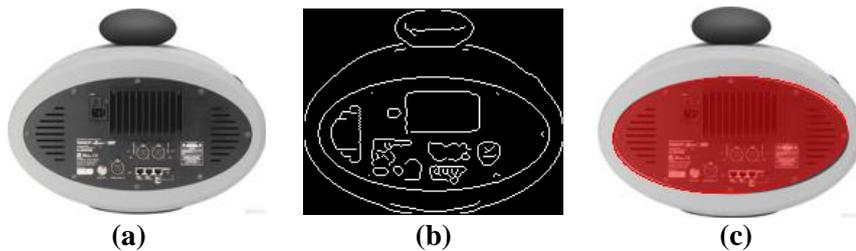


Figure 57 (a) The image in Figure 51(a), (b) its edge map and (c) automatically detected ellipse by EDDI.

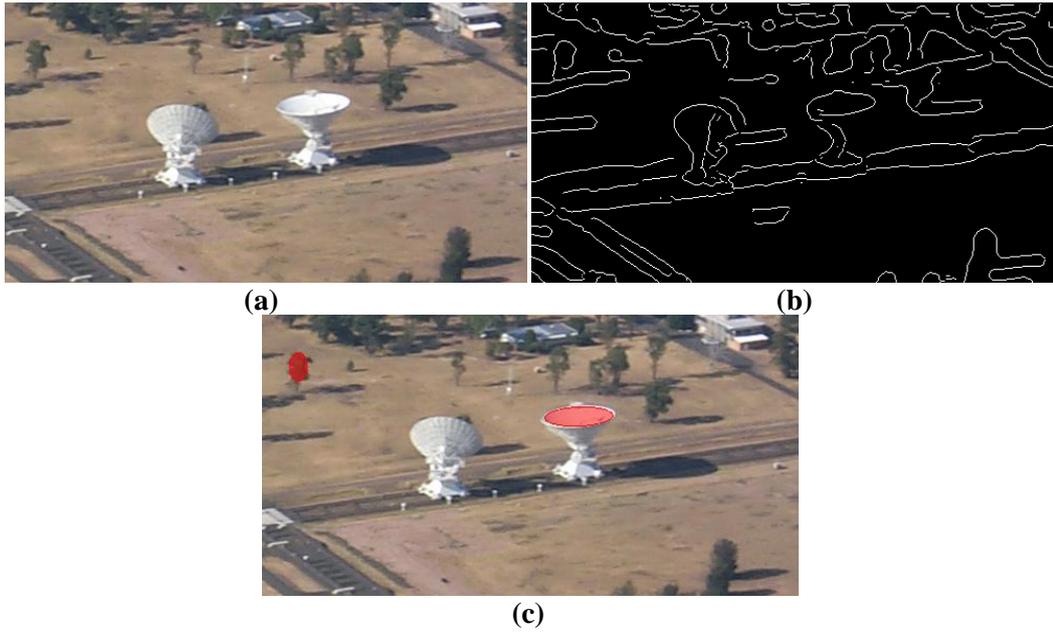
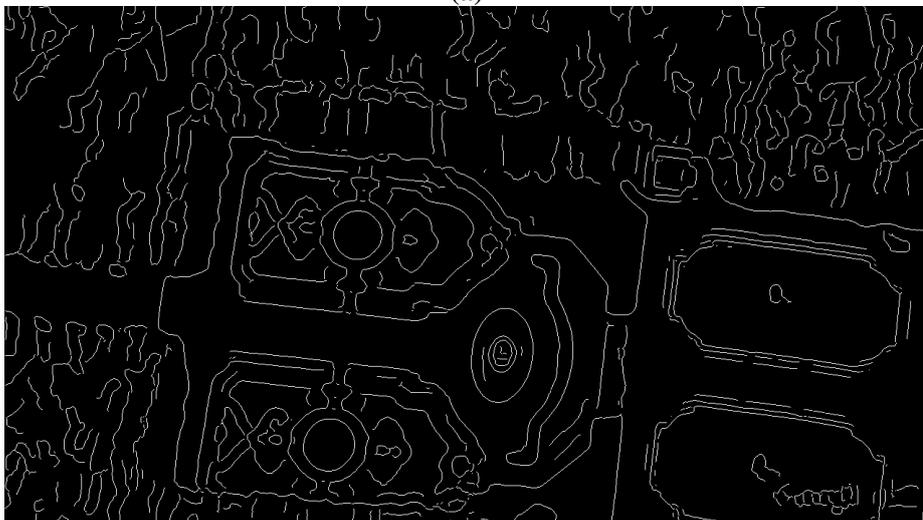


Figure 58 (a) The image in Figure 52 (a), (b) its edge map and (c) automatically detected ellipses by EDDI.



(a)



(b)



(c)

Figure 59 (a) A satellite image of a garden taken from Google Earth, (b) its edge map and ellipse detections using (c) EDDI.

4.3 Parameter Selection for EDDI

EDDI has five parameters to adjust. α , β , are used to decide whether the shape of the examined object is an ellipse or not. As these parameters get close to 1, the algorithm tends to detect only the perfect ellipses. So, in order to handle the distorted ellipses, α and β , should be decreased. Since the parameters estimated from each profile should be similar λ should be close to 1. But this value can be decreased to detect distorted ellipses.

The last two parameters, n_n are n_p , are used to handle missing edges of the examined shapes. If the contrast of the objects are high enough and if unconnected objects are examined these parameters should be selected as 0. However, as the contrast gets lower or there exist connected objects these values should be increased.

As a guideline for the selection of these parameters, α , β , λ can be initialized from 0.75. This value is seen as an optimal value for detection of the ellipses in real life scenes when the contrast is high enough. If the results are not good enough, these values can be decreased, together. As these parameters define the similarity of the shape to a rectangle and each of them is defined in (0,1], it is easier to set them to the same value. On the other hand, n_n are n_p depends only on the image. These values should be initialized as 0 and the results should be observed. If the image is noisy or the ellipses are intersects with other objects these values should be increased so it is better to analyze the edges on the images before changing these parameters. Since increasing the values of these parameters induces false detections and causes increase in the operation duration of EDDI, n_n and n_p values should be selected optimally.

4.4 Discussion

In this chapter, CDDI and RDDI approaches are adapted for detection of the ellipses on the images. As in these approaches, the directional profiles of the ellipse edges are analyzed and a similar algorithm is obtained. The validity of the approach is presented by experiments using synthetic and real images.

The results of Chapters 2-4 show that, the algorithm can easily be generalized for any other shape once the closed form expression of the shape is known. Starting from these expressions, 1-D integrations over different directions can be derived according to the parameters of the shape. Note that, it is also possible to approximate these integrations by evaluating these profiles numerically for very complicated shapes. Finally, a detection procedure can be determined after finding the critical points of obtained functions.

Despite the proposed method can be applied both for concave and convex shapes, it is easier to modify for convex shapes since the critical points are on the boundaries of the profile. Moreover, for convex shapes, the critical points are distant from each other which induce robustness against shape distortions since it is harder to confuse the locations of those points. However, as the complexity of the shape (i.e. number of parameters to define the shape) increases, the number of critical points to be detected increases which also cause higher computational load.

CHAPTER 5

AN APPLICATION: RECOGNITION OF GEOMETRICAL OBJECTS IN THE INDUSTRIAL REGIONS

In the previous section, one of the most important steps of the man-made structure detection is examined and a fast approach for identification of the structures in different geometrical shapes is proposed. In this chapter, the motivation is presenting an application area for the proposed shape detection method and show how this method can be used for practical problems.

Remote sensing applications is one of the possible application areas since there are many different types of geometrical structures on remotely sensed images and the processing should be fast due to the mass of collected images. The analysis show that there are different types of buildings in the industrial areas; hence detection and recognition of these buildings are examined in this chapter. POLs are one of the commonly observed, circularly shaped buildings in the industrial regions. Other buildings are generally in rectangular shapes and in the following sections a schema for detection and recognition of these structures is explained.

5.1 Detection and Recognition of POLs

The aim of this section is to develop a schema for recognition of POLs in the industrial regions. During the experiments performed in the previous chapter, some examples are given about the detection of the man-made structures in the industrial regions. These efforts showed that pre-processing for noise reduction and determination of the region of interests (ROI) where the circular structures are searched, and post-processing to identify the circles as the desired objects are also very important in order to increase the detection probability and reduce the false detections. The proposed schema can be seen in Figure 60 and it is basically composed of land cover/use classification, industrial zone determination, circle detection and POL identification steps. In the following parts of this chapter, the details of these steps are explained.

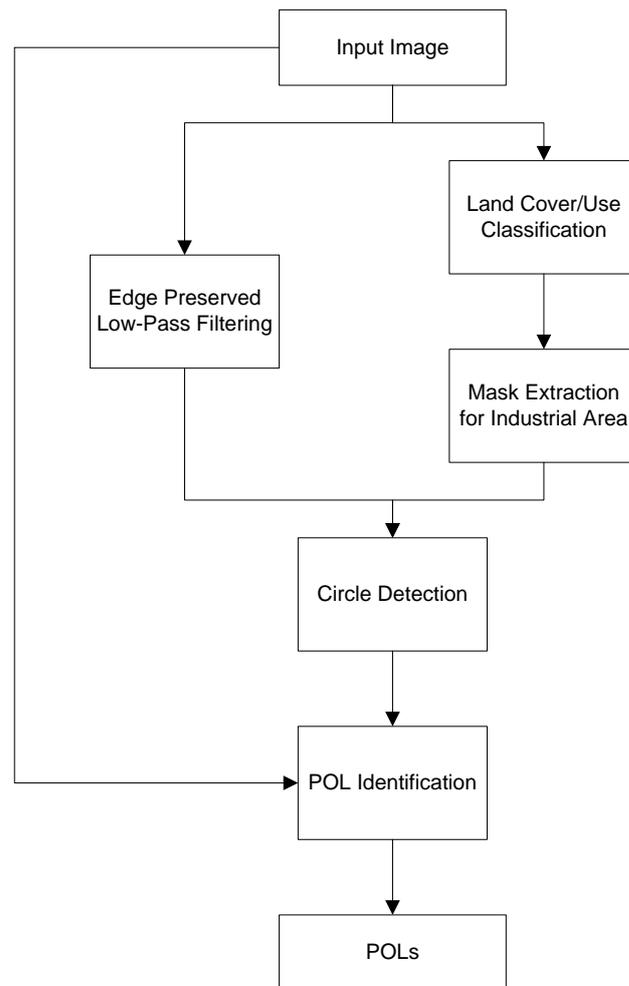


Figure 60 Block diagram for POL recognition process.

5.1.1 Land Cover/Use Classification

Land cover/use classification is the key step for reducing the search space and determining the ROI for the desired targets in remote sensing applications. However, this issue is not in the scope of the thesis and there are numerous studies in the literature to overcome the difficulties of this step. Hence, in this thesis, the state-of-the-art strategies are followed to classify the earth surface and determine the industrial regions in where desired buildings are searched.

5.1.1.1 Literature Survey

There are various researches about the processing of aerial images, [1, 73], with the tremendous increase in the collected data. The purpose and the scope of those studies vary a lot but very similar techniques are used in order to achieve their goals.

On the other hand there are very few works in the literature about detecting the industrial areas using aerial images. These works are generally about detecting the deformations or change on the earth surface which aims to find the mines, [74-79]. In these studies, the change of the terrain around the mines or the coal fires frequently observed in the coal mines are acquired for determining the mines.

For determining the change of the terrain some image processing techniques like clustering, estimation, image differencing and vegetation differencing. In [80], the feasibility of the usage of LANDSAT and SPOT images and some image analysis techniques. In this study, the ability of multi-temporal satellite data for making rapid classification to detect the temporal and spatial changes is analyzed. Moreover, a variety of standard image processing routines were run on the satellite data set like preprocessing and image enhancement, unsupervised and supervised classifications, merging of SPOT, LANDSAT and airborne imageries, image transforms and filtering techniques.

In [77], JERS-1 SAR multi-temporal images are used to analyze the deformations around the gold mines according to the mining activities. In spite of the limitations in the image processing techniques due to the nature of the nature of SAR images, the normalized difference index (NDI) images allowed identification of cover changes related to mining activities. Another study was presented by Wang et. al. which aims to detect the POLs by fusing the EO and SAR information taken from SPOT-5 and RadarSat-1 images, respectively [81]. In this method, bright points on the SAR image are detected as the candidate regions and around those regions, the extracted edges from EO images used to detect the circular objects which are finally labeled as the POLs on the region.

Despite the limited number of studies about industrial area detection in the literature, there are many studies about the automatic detection of man-made objects, especially buildings and roads, [82]. These studies are also important since it is thought that the proposed approaches can be adopted or a part of it will be very beneficial to reach the goal.

The existing studies about man-made object detection can be classified according to very different aspects like the methods used, resolution of the images used, complexity, etc. but one of the distinctive classifications is made according to the required human interaction. Semi-automatic systems requires human guidance in a level, [83-91], whereas the automatic systems don't. Since the aim of this study is to propose a fully automated system, only the survey only deals with the automatic systems or the automatic parts of the systems.

The building detection methods generally seek for rectilinear shapes, [47, 92-97], and most of them are proposed for very high resolution images since they are generally used for reconstruction or geographic information system (GIS) update. In [47], the linear edges are detected first and next parallelogram hypothesis are formed that are consistent with the projective constraints given by the viewing geometry. Hence, this system is limited to detect only rectilinear buildings with flat roofs. The promising hypotheses are selected according to some 2-D and 3-D cues like walls and shadows. Since these cues give information about the height of the building, the 3-D model of the building is obtained. After that, building data extracted from multiple images of the same scene are integrated to obtain better and more robust performance. Moreover, an interactive editing system is proposed to correct the possible errors of the automatic system. An improved snake model is proposed in [96] for detecting building contours in complicated urban environment in high-resolution aerial images. Combining with the radiometric and geometric characteristics of buildings, snake model is modified on the initiation criteria and the external energy function. In [94] and [98], an approach for the extraction of trees and building shapes from color infrared image and height data in an urban environment is proposed. For detecting the trees, a model using geometric and radiometric features as well as neighboring relations between trees is used where the buildings are detected using the

analysis of invariant moments. The buildings are assumed to be decomposed into some rectangular shaped objects; hence only angled shapes can be detected as buildings. A hierarchical approach is used to automatically interpret the landscapes.

The algorithms proposed in [92] also assume generic class of flat roofed, rectilinear buildings. In [97], the extraction of the buildings are achieved by the fusion of the LIDAR data and aerial images using polyhedral building methods. Multiple images of the same scene are also utilized in [93]. The approach is based on surface reconstruction, followed by extraction of visible building facades. Then different facades of the same objects are merged to detect the buildings. The algorithm proposed in [99] is also limited to detect rectilinear shapes since the lines in the aerial images are extracted and the buildings are detected according to the relations between those lines.

The method in [95], which uses photometric and geometric features for detection, is not limited to detect only the rectilinear buildings. It starts with a seeded region growing algorithm to segment the entire image and a numerical classification approach is performed to differentiate the classes as building and non-building.

Land cover/use classification is another important issue for interpretation of the images and object detection in the aerial images since the landscape may be very large where the acquired objects are in a very condensed region.

Early applications for the segmentation of the aerial images were used only the gray scale values (intensity) of the pixels, [100]. However, the color image segmentation becomes popular with the increase in the computational powers of the devices; since more detailed information can be obtained by use of the color information, [101]. These traditional pixel based classification approaches show meaningful results when two or three land cover classes are in considerations. On the other hand additional information is required to separate more complex regions (i.e. man-made regions like residents, industries, etc.) which become possible with the increase in the image resolution. Hence, information in optical bands are started to be supported by the texture analysis, [102]. In the texture analysis, the pixels are assumed to be located according to some rules which are generally periodic or completely random in the spatial domain. Some of the widely used are co-occurrence matrices, fractal dimension, autocorrelation, Markov Random Fields (MRF), Voronoi neighborhood models, Gabor and other Wavelet filters, Zernike moments, etc. [103]. In [104] a fast and accurate texture descriptor is proposed which utilizes the orientation and the local variance of the regions. Moreover, many researchers examined the performance of those texture features and compare them for different applications. Madasu and Yarlagaadda compared four texture features, descriptors, heuristic function, fuzzy logic and Mask based features for segmentation by use of Fuzzy C Means classification [105]. In [106] and [107] Gabor, pyramidal and tree structured wavelet based textures are compared. Some of the texture features are also compared according to the effectiveness for different applications that uses different imaging modalities [108, 109].

In the literature, those different texture features are widely used for land cover/use classification purposes. One of the very first texture analysis of the aerial images is given in [110] and a method for aerial image classification is proposed in [111] which clusters the aerial images using a fuzzy schema. In [112] and [113] the fusion of color and texture information is analyzed and the performance increase is proved with respect to the separate

usage of those information. In [114] SPOT and ERS satellite images are utilized for determination of the residential regions using a Markovian model and in [115] these regions are classified by the fractal textural structures. Texture analysis also used for forest classification and flora mapping [116, 117].

Another issue about the aerial image interpretation is the method used to classify the pixels or sub-blocks of image. For this purpose, many different approaches were proposed. Many of those methods are unsupervised techniques, [100, 118], which is not applicable for the problem of this thesis since the contents of the clustered parts should be designated in order to interpret the landscape. In [113] a maximum likelihood classification (MLC) algorithm is applied to aerial images for supervised segmentation of the image. Since support vector machine (SVM) is the “best” generalized solution to the boundary decision problem [119], it is applied to the supervised segmentation of the aerial images. From those works [120] integrates SVM with pyramid image and decision tree. In [121], the visible and infrared aerial images are segmented using SVM and the performance of multi-category SVM techniques are analyzed. In [122], SVM is compared with three other popular classifiers including the MLC, neural network classifiers (NNC) and decision tree classifiers (DTC). The impacts of the kernel configuration on the performance of the SVM techniques are also analyzed in this study. In [123], it is proposed to use active SVMs for segmentation of aerial images while reducing the number of labeled points. SVMs were also used on satellite images to detect hotspots on the Jharia coal filed region of India, [79].

5.1.1.2 Feature Selection

As explained in the previous part, for segmenting the industrial areas in an aerial scene, it is firstly desired to classify the image into some parts having similar patterns. For this purpose, each pixel of the image is analyzed and a vector of descriptors is obtained that represents the characteristics of the pattern at that pixel location.

In this study, the elements of the descriptors vector contain the information about the color and the texture. In the following sections the method for the descriptor evaluation methods are explained.

5.1.1.2.1 Color Descriptors

The color images are represented in red (R), green (G) and blue (B) space hence the color descriptors are selected as the normalized R, G and B components of the pixel which are defined in [0,1].

5.1.1.2.2 Texture Descriptors

In this study, four texture features are selected to use during the land cover/use classification of the 3-band satellite images. Since textural features aims to represent the distribution of the pixels in a region, in this study all the defined texture features are evaluated for a region $B_{x,y}$ around each of the spatial pixel coordinate (x, y) .

First one of these features is called entropy and it is defined as a measure of the randomness of a random variable by the given formula:

$$H = \sum_{\forall x} p(x) \log(p(x)) \quad (140)$$

where $p(x)$ is the probability distribution function (pdf) of the random variable x . Then, before evaluation of this feature, the intensity distribution in the region $B_{x,y}$ is estimated and this descriptor, $d_1(i,j)$, is evaluated as given in the equation below:

$$H(x, y) = \sum_{(m,n) \in B_{x,y}} p(I[m, n]) \log(p(I[m, n])) \quad (141)$$

Entropy is defined between zero and logarithm of the number of possible discrete values of pixel intensities. In this study, the intensity values are digitized into N_L levels during the entropy calculations in order to reduce the computation load and normalized into [0,1] to get better performance during the classification procedure. Then, the defined entropy feature used in this study can be described as follows:

$$\hat{H}(x, y) = \frac{1}{\log(N_L)} \sum_{(m,n) \in B_{x,y}} p(\bar{I}[m, n]) \log(p(\bar{I}[m, n])) \quad (142)$$

The next three texture features are extracted using the Gray Level Co-Occurrence Matrix (GLCM). GLCM is the two dimensional matrix of joint probabilities of the pixel pairs separated by a distance of d and an angle of θ . It is first proposed by Haralick [124] and widely used in many of the pattern recognition and computer vision applications [55, 119]. GLCM, $P_{d,\theta}$, can be defined as in Equation (143):

$$C_{d,\theta}(i, j) = \sum_{p,q \in B_{x,y}} \begin{cases} 1, & \text{if } \bar{I}(p, q) = i \text{ \& } I(p + \Delta x, q + \Delta y) = j \\ 0, & \text{otherwise} \end{cases} \quad (143)$$

$$P_{d,\theta}(i, j) = \frac{1}{K} C_{d,\theta}(i, j)$$

where $\bar{I}(p, q)$ are the quantized intensity values, $\Delta x = d \cos \theta$, $\Delta y = d \sin \theta$ and K is the normalization constant which is evaluated by the following formula:

$$K = \sum_i \sum_j C_{d,\theta}(i, j) \quad (144)$$

During the calculations of GLCM, the intensity values are quantized into N_L levels ($N_L < 256$) since the resultant matrix is $N_L \times N_L$, and especially when the bit depth of the image is high, i.e. 16 bits, the resultant matrix occupy large space in the memory and the computation load for the feature extraction step is so high.

The GLCM computation procedure is also clarified using Figure 61. As seen from this figure, when $d=1$ and $\theta=0$, horizontally adjacent pixel intensities are examined and the corresponding index of $C_{1,0}$ matrix is increased by one. When this matrix is analyzed it is shown that $C_{1,0}(1,1)$ is equal to one since there are one horizontally adjacent pixel both having values of 1. Similarly, $C_{1,0}(2,4)$ is equal to two since there are two instances of horizontally adjacent pixel on the image with the intensity values 2 and 4.

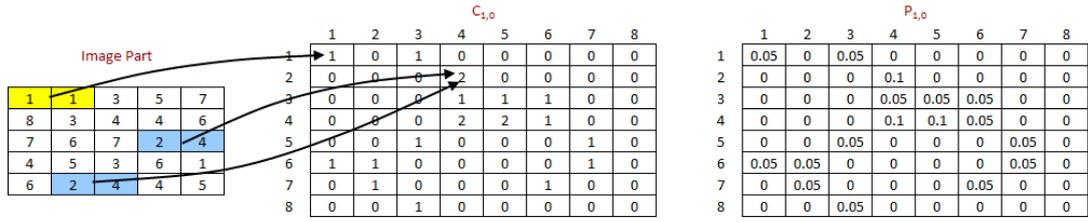


Figure 61 Procedure for GLCM computation when $d=1$ and $\theta=0$.

In this study, the aim is classify the earth surface independent of the look angle and orientation of the structures in the scene. Hence it is essential to use a set of rotation invariant texture descriptors. As result, the GLCM features that will be explained in the following paragraphs are extracted from the GLCM averaged by four GLCMs obtained in directions 0, 45, 90 and 135 with same parameter of d as given below:

$$P_d(i, j) = \frac{P_{d,0^\circ}(i, j) + P_{d,45^\circ}(i, j) + P_{d,90^\circ}(i, j) + P_{d,135^\circ}(i, j)}{4} \quad (145)$$

From extracted GLCM, Haralick proposed to use fourteen features in [124]. However, in the literature less number of features are widely used and in this study it is preferred to use the most common four of them; energy, contrast, homogeneity and entropy [125, 126].

Homogeneity feature is a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal and evaluated by the given formula:

$$H_{P_d} = \frac{P_d(i, j)}{1 + |i - j|} \quad (146)$$

The second GLCM feature used in this study is contrast. Contrast feature is a measure for the local variations in the GLCM and evaluated by the following formula:

$$C_{P_d} = \frac{1}{N_H} \sum_i \sum_j |i - j|^2 P_d(i, j) \quad (147)$$

where N_H is the normalization constant and equals to $(N_L - 1)^2$ hence the range of this feature is in $[0,1]$ and for a diagonal GLCM its value is equal to 0.

The last GLCM feature, energy, is used to measures the overall probability of having distinctive grayscale patterns in the image and it is evaluated by the square sum of the GLCM elements as given in Equation (148), which is again defined in $[0,1]$.

$$E_{P_d} = \sum_i \sum_j P_d^2(i, j) \quad (148)$$

5.1.1.3 Aerial Image Classification

As the second step of the proposed method, each pixel of the image is classified as a predefined type according to its feature vector. For the purpose of this classification, Support Vector Machines (SVMs) will be used.

SVMs are mainly linear discriminant machines. The motivation of these types of machines is representing the patterns in higher dimension, typically in much higher dimension than the original feature space. The reason of this transformation is that it was proved that with an appropriate nonlinear mapping of the original pattern to a sufficiently higher dimension, data from two categories can be separated by a hyperplane, [119].

In order to explain the details of SVM let x_k be a pattern in an N-element dataset and y_k be corresponding transformed pattern in the higher dimension using the appropriate nonlinear mapping, $\varphi()$. Hence, the relationship between y_k and x_k can be given as:

$$y_k = \varphi(x_k) \quad \text{for } k = 1, 2, \dots, N \quad (149)$$

For each of the pattern let $z_k = \pm 1$ according to whether the k^{th} pattern is in the set of the first type of clusters or in the set of the second type of clusters. Since for a linear discriminant function $g()$ in a space y can be expressed as $g(y) = a^t y$, a separating hyperplane guarantees that:

$$z_k g(y_k) \geq 1 \quad \text{for } k = 1, 2, \dots, N \quad (150)$$

In [119], the main goal of SVMs are described as finding the separating hyperplane with the largest margins since the larger margin implies better generalization for the solution of the problem.

The distance between the transformed pattern y_k and the hyperplane is evaluated as [119]:

$$d_k = \frac{g(y_k)}{\|a\|} \quad (151)$$

Using this distance and assuming the existence of a positive b , the Equation (150) implies that

$$z_k d_k \geq b \quad \text{for } k = 1, 2, \dots, N \quad (152)$$

Hence the aim of the SVM becomes finding the weight vector, a , that maximizes b .

The support vectors (SVs) are defined as the most informative patterns in the transformed dataset. They are also the hardest patterns to classify since they are the closest ones to the separating hyperplane. Hence, these patterns are used in training process and define the separating hyperplane.

As a result, in SVM, for determining the separating hyperplane between any two types of pattern clusters, the following steps are followed:

1. Obtain the features, x_k 's, of the patterns in each of the clusters
2. Obtain the z_k 's according to the clusters that x_k 's belongs to
3. Obtain support vectors and the separating hyperplane between the two type of clusters

Note that, the SVM has a very complicated algorithm and there exist numerous open source implementations of the algorithm. Hence, in this study one of these implementations described in [127] is used to accomplish the work describe in step 3.

After obtaining the separating hyperplane, for the classification of a pattern, the feature vector of this pattern, say f , is transformed to the higher order plane using the mapping function $\varphi()$ and the obtained vector, say h , is multiplied by a^t from right. Then, the pattern is classified as from one of the two types of the predefined clusters according to the sign of the result of $a^t h$.

So forth, a technique for classifying the dataset into two categories has been explained. On the other hand, in this study, it is planning to classify the patterns in the aerial images as one of the five predefined categories:

- Industrial Areas
- Residential Areas
- Water
- Forest
- Bare Land (soil, rock, green spaces, etc.)

In the literature, there are different approaches for multi-category classification using SVMs, [120, 122, 123, 127, 128]. Among the methods used in those studies two of them, called “one-against-all” and “one-against-one” approaches, are the most commonly used ones.

In the first approach, M different SVMs, say S_1, S_2, \dots, S_M , are used for solving the M -category problem. During the evaluation of m^{th} SVM, S_m , z_k is set to 1 if x_k belongs to the m^{th} category and -1 if x_k belongs to one of the remaining categories. Hence S_m is used to decide whether a pattern is from m^{th} category or not, which is the reason why this method is called “one-against-all” approach. To sum up, M -class problem is solved using the results of the M two-class SVMs, [123].

In the second approach, $C \binom{M}{2} = \frac{M(M-1)}{2} = M_C$ different SVMs are used for solving

the M -category problem. In this case, during the evaluation of any of the set of SVs, only the patterns from two categories are used. Hence these SVs are used to classify as one of these two categories. Hence this approach is called “one-against-one” approach. Using a voting strategy, M -class problem is solved using the results of M_C two-class SVMs and the class having maximum votes indicates the category of the pattern, [127].

In this study, “one-against-one” approach will be used since in [128] it was found that this approach is more practical and successful than the “one-against-all” approach. In [127], if there is two classes with maximum number of votes, the pattern is classified as from the class with smaller index.

5.1.1.4 Experimental Results

For the training of the SVMs numerous of data from each of the clusters were extracted. For this purpose 142 different scenes are captured from Google Earth with 1m of resolution. In Table 9 the number of pixels for each class in this dataset and the area covered by these regions are given.

In order to decide on a successful classification different sets of feature vectors are selected and classification performances are examined. In

In Table 11, Table 12 and Table 13 the confusion matrices for each feature set are presented. In the given matrices the columns represent the instances in the predicted class and each row represent the instances in the actual class. Hence, the diagonal elements show the probability of true classification of a pixel in the corresponding category where the other elements show the probability of confused classifications. Note that, in these matrices the capitals R, I, F, W and L are used to represent the categories “Residential”, “Industrial”, “Forest”, “Water” and “Bare Land”, respectively.

As seen from Table 11 only the color information is not discriminative especially for “Residential” and “Industrial” classes. It is able to classify the “Forest” pixels quite accurately where for pixels in “Industrial” and “Bare Land” categories the accuracy is about 75%.

Table 10, the tested feature sets can be seen.

The classification procedure is achieved as indicated in Chapter 5.1.1.3 which is implemented in C++ using LIBSVM library [129]. Since the training of an SVM is a very complicated procedure, it is almost impossible to train the machine with such a huge amount of data. Hence, before the training of the SVM, these data in each type of category are clustered using Fuzzy C-Means (FCM) technique, and only 2500 discriminant vectors are obtained for each category. The details of this clustering approach are given in APPENDIX B. Hence for each class defined in Table 9 there are 2500 vectors with length of D_i is used to generate the SVM models, and all the pixels counted in the second column of Table 9 are used for classification to measure performance of the feature sets. Since the FCM converges to different result for different initial cluster centers and they are determined randomly in these experiments, the experiments are performed 10 times and the results are averaged in order to measure the performance of the classifications. Then the results are presented using the confusion matrix which shows the probabilities of true and false classifications for each class.

Table 9 Number of pixels used in training of SVMs and their coverage areas according to the cluster types.

Type of the Cluster	Number of Images	Number of Pixels in the Dataset	Total Area (km ²)
Residential	55	14575550	~14.58
Industrial	71	11789175	~11.79
Water	38	12306100	~12.31
Forest	60	12432025	~12.43
Bare Land	70	14618025	~14.62

In Table 11, Table 12 and Table 13 the confusion matrices for each feature set are presented. In the given matrices the columns represent the instances in the predicted class and each row represent the instances in the actual class. Hence, the diagonal elements show the probability of true classification of a pixel in the corresponding category where the other elements show the probability of confused classifications. Note that, in these matrices the capitals R, I, F, W and L are used to represent the categories “Residential”, “Industrial”, “Forest”, “Water” and “Bare Land”, respectively.

As seen from Table 11 only the color information is not discriminative especially for “Residential” and “Industrial” classes. It is able to classify the “Forest” pixels quite accurately where for pixels in “Industrial” and “Bare Land” categories the accuracy is about 75%.

Table 10 Feature sets used for classification experiments.

Set No.	Feature Vectors	Dimension (D_f)
1	RGB	3
2	Entropy, GLCM Features, $B_{x,y}$: 31x31, d=1	4
3	Entropy, GLCM Features, $B_{x,y}$: 201x201, d=1	4
4	Entropy, GLCM Features, $B_{x,y}$: 31x31, d=10	4
5	Entropy, GLCM Features, $B_{x,y}$: 201x201, d=10	4
6	RGB Entropy, GLCM Features, $B_{x,y}$: 31x31, d={1,10} Entropy, GLCM Features, $B_{x,y}$: 51x51, d={1,10} Entropy, GLCM Features, $B_{x,y}$: 75x75, d={1,10} Entropy, GLCM Features, $B_{x,y}$: 101x101, d={1,10} Entropy, GLCM Features, $B_{x,y}$: 151x151, d={1,10} Entropy, GLCM Features, $B_{x,y}$: 201x201, d={1,10}	45

Table 11 Confusion matrix when only color features are used (Feature Set 1).

	R	I	F	W	L
R	0.42	0.22	0.02	0.12	0.22
I	0.17	0.65	0.04	0.03	0.11
F	0	0	0.94	0.06	0
W	0.01	0.01	0.14	0.72	0.11
L	0.13	0.06	0.01	0.04	0.75

Table 12 Confusion matrices when only texture features are used for (a) Feature Set 2, (b) Feature Set 3, (c) Feature Set 4, (d) Feature Set 5.

	R	I	F	W	L
R	0.86	0.10	0.00	0.04	0.01
I	0.42	0.38	0.01	0.09	0.10
F	0.00	0.00	0.87	0.03	0.10
W	0.05	0.03	0.00	0.83	0.08
L	0.01	0.05	0.09	0.21	0.64

(a)

	R	I	F	W	L
R	0.86	0.10	0.00	0.04	0.00
I	0.38	0.37	0.01	0.20	0.04
F	0.00	0.00	0.86	0.00	0.13
W	0.07	0.03	0.00	0.70	0.20
L	0.01	0.03	0.09	0.34	0.53

(c)

	R	I	F	W	L
R	0.80	0.18	0.00	0.02	0.00
I	0.28	0.65	0.00	0.02	0.06
F	0.00	0.01	0.92	0.02	0.06
W	0.09	0.10	0.00	0.72	0.08
L	0.02	0.08	0.04	0.15	0.70

(b)

	R	I	F	W	L
R	0.90	0.09	0.00	0.00	0.00
I	0.14	0.78	0.00	0.04	0.05
F	0.00	0.00	0.93	0.01	0.06
W	0.05	0.13	0.01	0.70	0.12
L	0.00	0.08	0.06	0.16	0.70

(d)

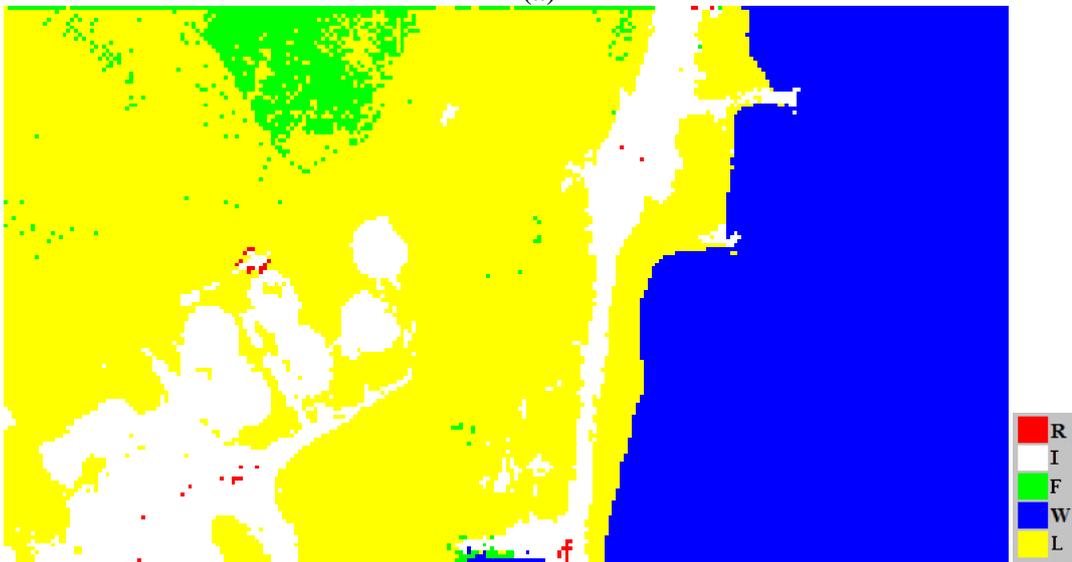
Table 13 Confusion matrix when color features and texture features with different parameters are used (Feature Set 6).

	R	I	F	W	L
R	0.92	0.08	0.00	0.00	0.00
I	0.10	0.87	0.00	0.00	0.02
F	0.00	0.00	0.99	0.00	0.00
W	0.02	0.01	0.00	0.96	0.01
L	0.01	0.02	0.01	0.02	0.94

When only the texture features are utilized the performance depends on the parameters used. It can be clearly seen that when the texture features are obtained for larger neighborhood, the pixels in the “Industrial” and “Bare Land” categories are classified more accurately. Since the structures in the “Industrial” category generally include large buildings, which have similar statistics in a small region with the small buildings, pixels in “Industrial” categories are classified as “Residential” when the neighborhood is selected as 31x31 block. Similarly, better performance is obtained for “Bare Land” classification when a larger area is selected for texture feature extraction.



(a)



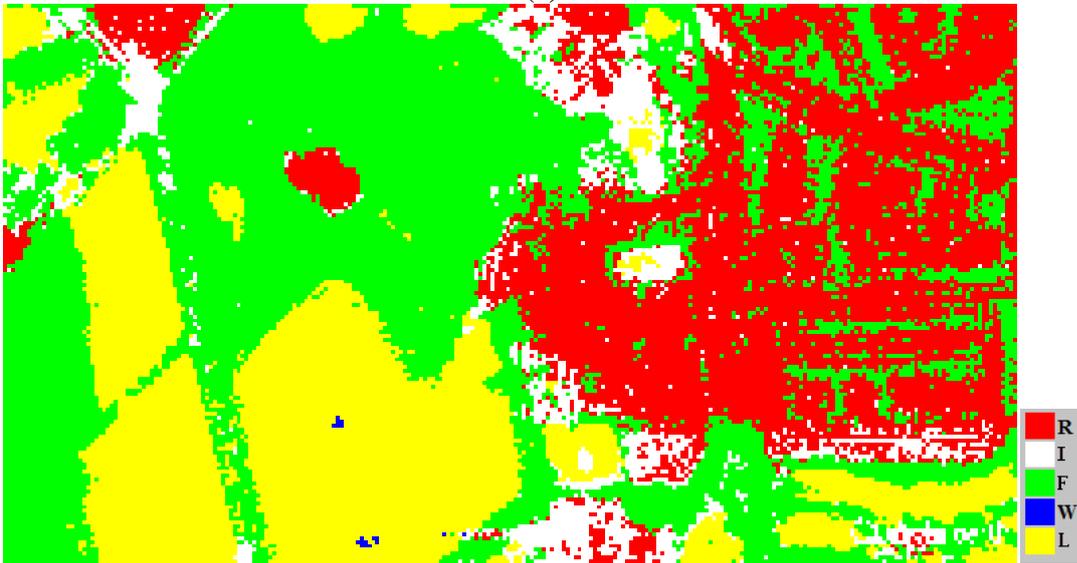
(b)

Figure 62 An example for (a) a satellite image and (b) its classification result.

Despite the performances for other three categories are independent of the texture feature parameters, correctly classified pixels differ from each other for each case. Hence, in this study it is proposed to fuse the color features with a larger set of texture features, evaluated for different neighborhood size and d parameters as given in the last row of Table 10. So, the performance of the approach is better and the overall accuracy is about 94% as seen in Table 13. For this feature set some classification results can be seen in Figure 62 and Figure 63 where the red pixels represent the regions classified as “Residential”, white pixels represent the regions found in “Industrial” class, blue pixels are used to represent the regions classified in “Water”, green is used to show the pixels found in “Forest” class and lastly yellow pixels represent the regions classified as “Bare Land”.



(a)



(b)

Figure 63 An example for (a) a satellite image and (b) its classification result.

5.1.2 Mask Extraction for Industrial Regions

The mask for industrial regions where the POLs will be searched in is determined by the result of the classification process given in the previous part. The parts of the images classified as “Industrial” are selected as the basis of this mask and a binary image is composed with the same size of the original image as given below:

$$M(i, j) = \begin{cases} 1, & \text{if } C(i, j) = \text{"Industrial"} \\ 0, & \text{otherwise} \end{cases} \quad (153)$$

where $M(i, j)$ is the mask for industrial region and $C(i, j)$ is the matrix that shows the classification results of the pixels of the image.

After that, the mask is dilated 10 times with a 3x3 square kernel to reduce the errors on the edges of the regions. Lastly, the areas classified as “Residential” completely covered by “Industrial” regions are added to mask since the analysis of the results show that there is a possibility of existence of small POLs inside these regions due to false classifications.

5.1.3 Prefiltering

Due to noise on the images and unwanted tiny details existing in the scene, the edge map of the images are quite complicated which requires quite long computations for detection of the desired objects. Hence, a low pass filtering is required to reduce the noise and eliminate the tiny structures on the images while preserving the edges of the desired shapes. For this purpose, bilateral filtering [44] is adopted as in the previous chapter.

Bilateral filter is an edge preserving smoothing filter by which the intensity values of each pixel is replaced by a weighted average of the pixel intensities around its neighborhood. The weighting is achieved by a Gaussian function and the distances between the pixels are evaluated in a 3-D space composed of the two spatial axes and the intensity. The bilateral filter can be expressed by the following equation:

$$I_{BF}(x, y) = \frac{1}{W_p} \sum_{\substack{\forall i \\ \{x_i, y_i\} \in S}} G_{\sigma_s} \left(\sqrt{(x-x_i)^2 + (y-y_i)^2} \right) G_{\sigma_r} \left(|I(x, y) - I(x_i, y_i)| \right) I(x, y) \quad (154)$$

where S is the defined neighborhood, G_{σ} is a Gaussian function given in Equation (155) and W_p is the normalization term evaluated as in Equation (156).

$$G_{\sigma}(t) = e^{-\frac{t^2}{2\sigma^2}} \quad (155)$$

$$W_p = \sum_{\substack{\forall i \\ \{x_i, y_i\} \in S}} G_{\sigma_s} \left(\sqrt{(x-x_i)^2 + (y-y_i)^2} \right) G_{\sigma_r} \left(|I(x, y) - I(x_i, y_i)| \right) \quad (156)$$

In this study, 7x7 square neighborhood around each pixel is used as the search region S ; σ_s and σ_r are selected as 5 and 1 for detection of POLs from 1m resolution satellite images.

After some trials, an additional high-pass filtering provides better edge detections hence the image is filtered with an unsharp contrast enhancement filter given in Equation after the bilateral filtering in the proposed approach to emphasize the region boundaries much better.

$$h = \begin{bmatrix} -1/6 & -2/3 & -1/6 \\ -2/3 & 13/3 & -2/3 \\ -1/6 & -2/3 & -1/6 \end{bmatrix} \quad (157)$$

5.1.4 Circle Detection

For detection of the circles for POL recognition, the proposed method given in the previous chapter is utilized. But in this case, the industrial mask is utilized through the edge detection process and only the edges in these regions are taken into account.

5.1.5 Post Processing and POL Recognition

The preliminary trials clearly showed that the parameters of the circle detection algorithms should be adjusted to loosen the thresholds used in taking decisions in order to detect the POLs successfully. The main reason for that is the noise on the images, existence of the other structures around the POLs that makes the edge maps more complex and of course the shape distortions due to the looking angle of the camera, illumination and the shadows. For this reason, the circle detectors produce outputs with high false alarm rates. Hence, the properties of the POLs should be analyzed and the false detections should be filtered out by using some rules.

When these structures are analyzed three main characteristics can be observed:

- At least half of their edges appear continuously on the edge map.
- Their surface is smooth.
- Their most common color is white

As a result, a circle, $c(r, x_c, y_c)$, is recognized as a POL if it satisfies one of the following rules:

$$1. \quad R_{edges,c} \geq \tau_R \quad (158)$$

where R_{edges} is used for “ratio of edges” and calculated as given below:

$$R_{edges,c} = \frac{1}{2\pi r} \sum_{\forall \{i,j\}: \frac{|r - \sqrt{(i-x_c)^2 + (j-y_c)^2}|}{r} \leq d} E(i, j) \quad (159)$$

After performed trials, the optimum τ_R values are found to be 0.75 when the radius of the circle is smaller than 10 and 0.5 for other cases; the optimum d is found to be 1.

$$2. \quad c_{v,c} \leq \tau_{cv} \quad (160)$$

where $c_{v,c}$ stands for the coefficient of variation and it is defined as the ratio the standard deviation of the intensities in the circle to the square of the mean intensity in the circle. By experiments τ_R is selected as 0.0005 if the radius of the circle is smaller than 10 and 0.00075 for the rest of the radius values.

$$3. \quad R_{color,c} \geq \tau_{color} \quad (161)$$

where $R_{color,c}$ is the ratio of the number of pixels, having red component greater than 160, green and blue components greater than 180 inside the circle, to the area of the circle. The trials yields with the optimum threshold value τ_{color} as 0.7 if the radius of the circle is smaller than 10 and 0.65 if it is greater than 10.

A next processing step is also required after the elimination of the possible false circles since there are some circles connected to each other or some circles in another. Their existence mainly depends on two main reasons:

- noise or freckles on the surface of the POLs
- shadows inside or outside of the POLs especially when the illumination angle is high.

In order to remove these false detection, first of all the circles are analyzed and the circles which have other circles inside are determined. Generally, these larger circles are tried to be preserved due to false detection by the noise or freckles on the surface of POLs hence they are preserved in the detection list if one of the conditions given in Equation (162) is satisfied; and they are removed from the list otherwise.

$$\begin{aligned}
\text{Cond 1: } & \min(r_{c_{i,j}}) \geq 0.85r_{c_i} \\
\text{Cond 2: } & R_{edges,c_i} \geq 0.9 \max(R_{edges,c_{i,j}}) \\
\text{Cond 3: } & R_{color,c_i} \geq 0.9 \max(R_{color,c_{i,j}}) \\
\text{Cond 4: } & c_{v,c_i} \leq 1.1 \min(c_{v,c_{i,j}})
\end{aligned} \tag{162}$$

where c_i is the circle having other circles inside and $c_{i,j}$ is the circles inside c_i .

Secondly, each connected circle pair is analyzed separately and more likely circle is preserved in the list. Since these types of connected circles are caused by the shadow effects, and intensity of the shadowed pixels are lower the intensity distribution within these circles is analyzed. Let I_1 , I_2 and I_3 are the mean intensities of the regions defined as P_1 , P_2 and P_3 , respectively in Figure 64.

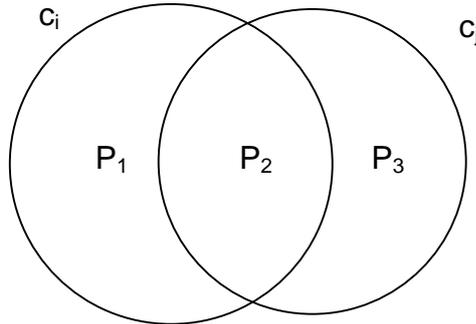


Figure 64 Two possible connected circle and defined region where mean intensities evaluated.

If I_1 is high (i.e. higher than 180), the connected region is probably on the POL. Then I_2 and I_3 are compared with each other and c_i is preserved if I_1 is 1.2 times larger than I_3 and c_j is preserved if I_3 is 1.2 times higher than I_1 . If they have similar values, on the other, the circle with higher R_{edges} is preserved in the detection list. Similarly, if I_2 is not high, the circle with higher R_{edges} is preserved in the detection list.

5.1.6 Experimental Results

In this thesis, the proposed schema for recognition of POLs is tested using the data set given in APPENDIX C. The performance of the POL recognition schema is evaluated as given in Figure 60 and the results are presented in Table 14. As seen in the table, over 80% percent of the POLs are detected correctly with a precision of 80%. The pixel wise performance given in the table shows that some of the pixels on the POLs are missed

despite that POL is recognized and the detected circle recognized as POLs include non-POL pixels since both precision and recall values decrease. However, since the decreases in these values are on the order of 5%, the recognition results are quite precise.

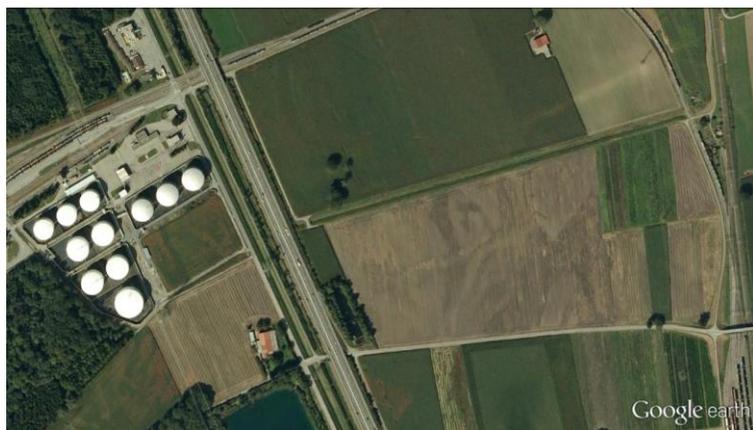
Table 14 POL recognition performance of the proposed schema evaluated using test set.

Circle Wise Performance	Precision	0.82773
	Recall	0.81567
Pixel Wise Performance	Precision	0.77985
	Recall	0.79613

From Figure 65 to Figure 69, some examples of the used images in the test set and the obtained results are presented. The generated industrial masks and the recognized POLs are shown in red color on the images while the edges of these regions are emphasized more to make the results clearer. For each of the image, the performances are given in Table 15.

Table 15 POL recognition performances for the images given from Figure 65 to Figure 69.

		Image 1 (Figure 65)	Image 2 (Figure 66)	Image 3 (Figure 67)	Image 4 (Figure 68)	Image 5 (Figure 69)
Circle Wise Performance	Precision	0.91667	0.89231	0.5	0.69444	1
	Recall	1	0.98305	1	1	0.44828
Pixel Wise Performance	Precision	0.81071	0.81123	0.85654	0.84407	0.93224
	Recall	0.93667	0.94414	0.88726	0.81561	0.3279



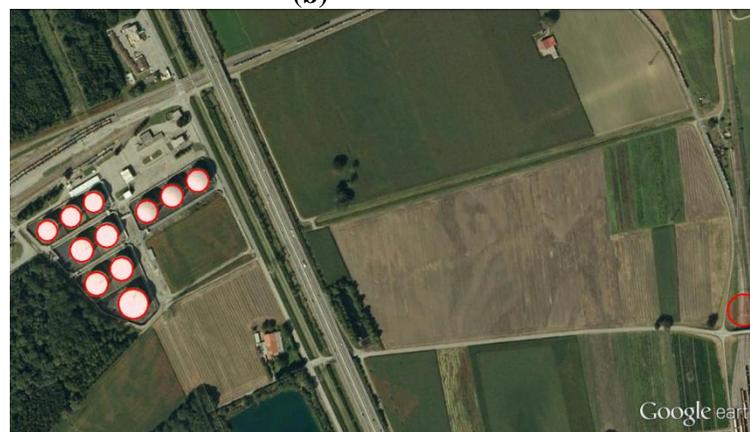
(a)



(b)



(c)

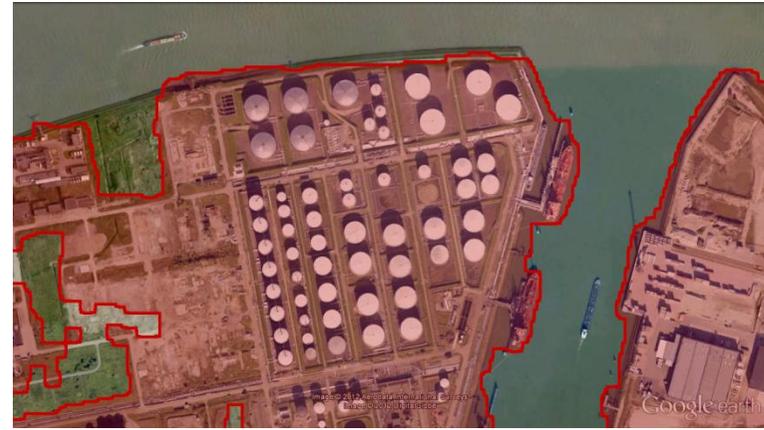


(d)

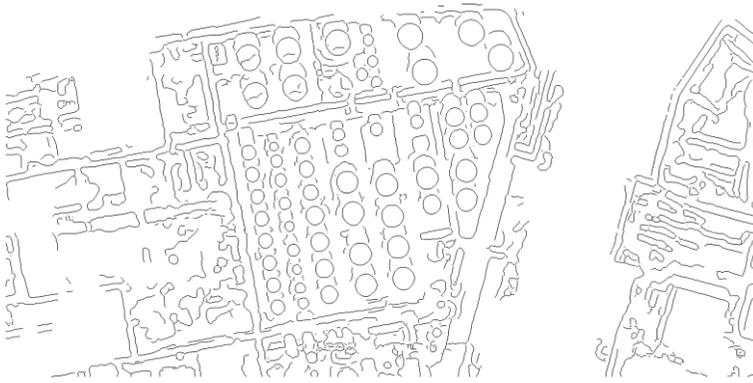
Figure 65 (a) An example image from test set, (b) obtained industrial region map, (c) edge map and (d) recognized POIs on this image.



(a)



(b)



(c)



(d)

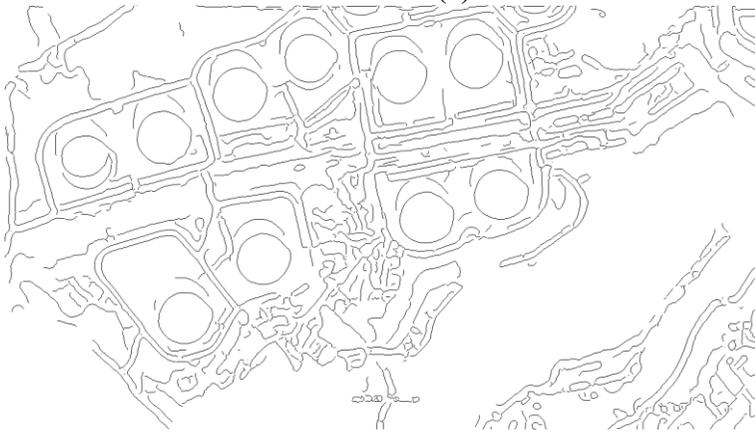
Figure 66 (a) An example image from test set, (b) obtained industrial region map, (c) edge map and (d) recognized POIs on this image.



(a)



(b)



(c)

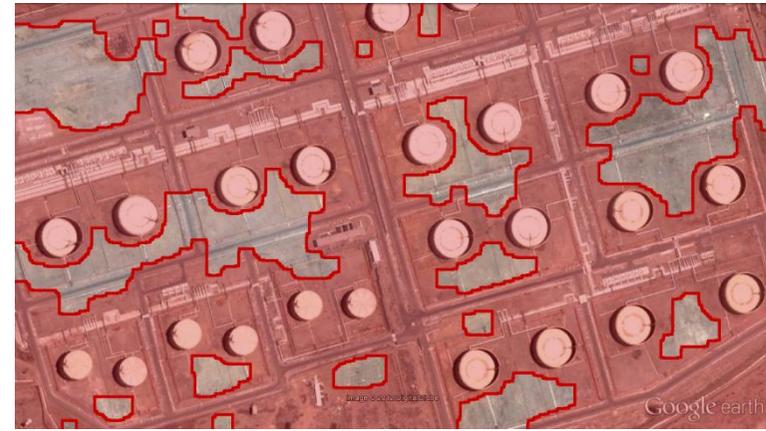


(d)

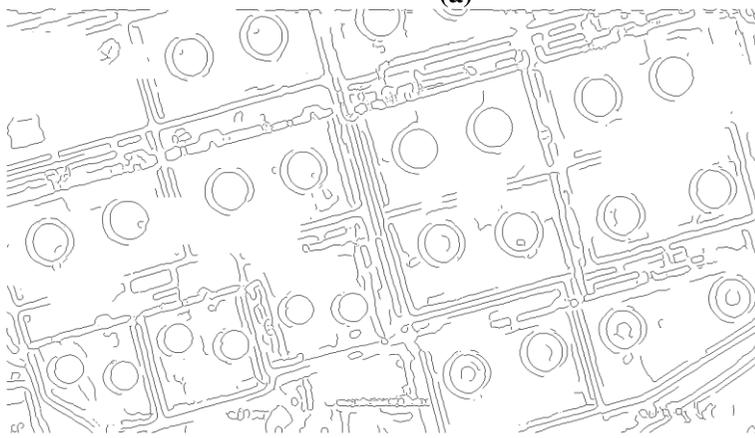
Figure 67 (a) An example image from test set, (b) obtained industrial region map, (c) edge map and (d) recognized POIs on this image.



(a)



(b)

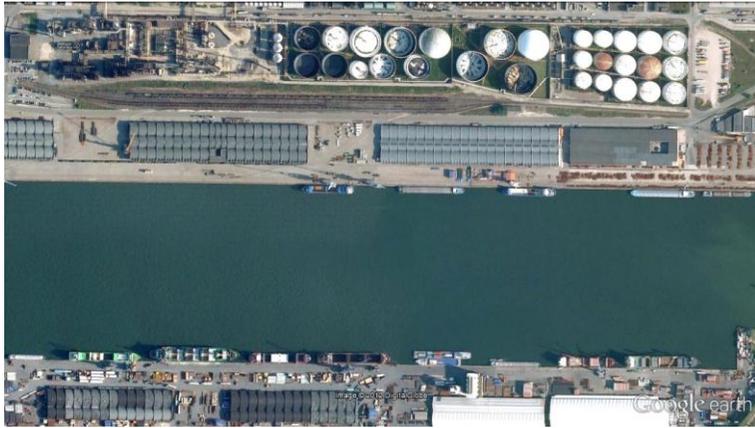


(c)



(d)

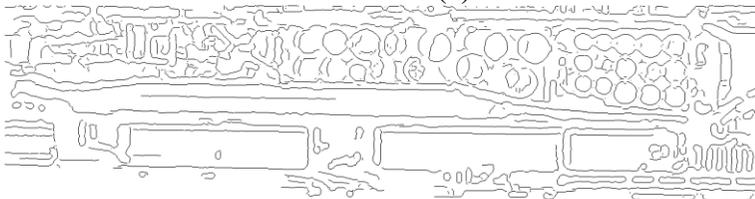
Figure 68 (a) An example image from test set, (b) obtained industrial region map, (c) edge map and (d) recognized POLs on this image.



(a)



(b)



(c)



(d)

Figure 69 (a) An example image from test set, (b) obtained industrial region map, (c) edge map and (d) recognized POIs on this image.

5.1.7 Comparison of Circle Detection Approaches in Terms of POL Detection

In this section, the circle detections approaches given in the previous chapter are compared in terms of their effects on detection of the POLs. For this purpose, the procedure given in Figure 60 is applied but the circle detection approach is changed. For this purpose, the experiments in the previous section for different parameter sets of MHT, MHT CC and RCD CC approaches. MHT CC and RCD CC are MHT and RCD methods which are applied for each connected component on the image separately as explained in CHAPTER 2. Note that, RCD method is not applied directly because of its excessive computation duration mentioned in the previous chapter.

For MHT, the best performance is obtained when r_{min} , r_{max} and r_{step} are selected as 5, 60 and 1. When MHT is operated on each connected component, the parameters kept same and n_n is set as 0 in order reduce the computation burden caused by the complexity of the images. n_n and n_p are set as 1 and 5 when RCD is operated on each connected components of the image edges and the best selection of other parameters are found as $T_f=1e^2$, $T_{min}=50$, $T_a=7$, $T_d=1$ and $T_r=0.5$. Detection performances of these methods are given in Table 16-Table 18.

Table 16 POL recognition performance of the proposed schema when MHT is used for circle detection.

Circle Wise Performance	Precision	0.84166
	Recall	0.81189
Pixel Wise Performance	Precision	0.89258
	Recall	0.80577

Table 17 POL recognition performance of the proposed schema when MHT CC is used for circle detection.

Circle Wise Performance	Precision	0.78813
	Recall	0.74167
Pixel Wise Performance	Precision	0.88612
	Recall	0.61933

Table 18 POL recognition performance of the proposed schema when RCD CC is used for circle detection.

Circle Wise Performance	Precision	0.72789
	Recall	0.82676
Pixel Wise Performance	Precision	0.848
	Recall	0.75098

As seen from the tables above, detection performance of MHT and CDDI is very similar to each other. When MHT and RCD are operated on the connected components of the edges the robustness of the approaches degrades. For those cases, the circle-wise precision values are lower than CDDI and MHT while pixel wise precision rates seem high. This is caused by the problems in detection of small POLs. The main problem of operating these methods in connected components is high miss rates for small POLs and false detection of small POLs. Moreover, pixel-wise recall rates are lower for these cases which show the estimated parameters of the circles are also worse than the conventional MHT and CDDI approaches.

Despite MHT has slightly better performances than CDDI, CDDI is much faster (about 2.5 times) than MHT. As seen from Table 19, CDDI is faster than all other approaches and has great advantage in terms of computation duration so can be selected for detection of POLs especially if the detection is examined in circle-wise.

Table 19 Average computation duration of circle detection methods for one image in seconds.

MHT	81.3699
MHT CC	108.5345
RCD CC	66.0486
CDDI	33.2189

For the images between Figure 65 and Figure 69, the results of the procedure for POL detection given in Figure 60 using MHT, MHT CC and RCD CC are given between Figure 70-Figure 74. These results also show the similarities between the outputs of the circle detection approaches. Besides, in and the POL-wise recall and precision rates evaluated for each of the images in the data set are presented using all the circle detection approaches mentioned above. As seen from those figures, there is at least one case for which one of the

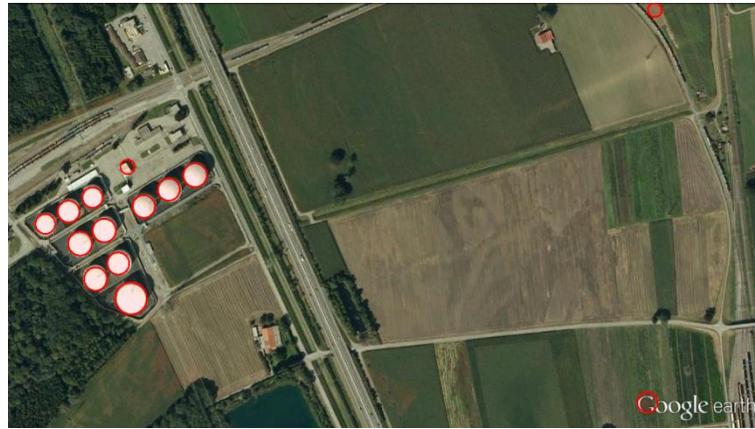
circle detection approaches performs best. Then, the selection of circle detection method may vary depending on the environmental situations.



(a)



(b)



(c)

Figure 70 Recognized POIs on the image in Figure 65 using (a) MHT, (b) MHT CC and (c) RCD CC.



(a)



(b)



(c)

Figure 71 Recognized POLs on the image in Figure 66 using (a) MHT, (b) MHT CC and (c) RCD CC.



(a)

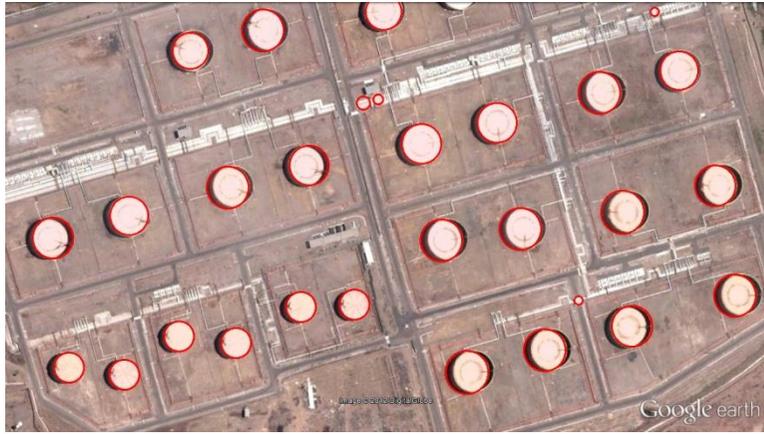


(b)



(c)

Figure 72 Recognized POIs on the image in Figure 67 using (a) MHT, (b) MHT CC and (c) RCD CC.



(a)

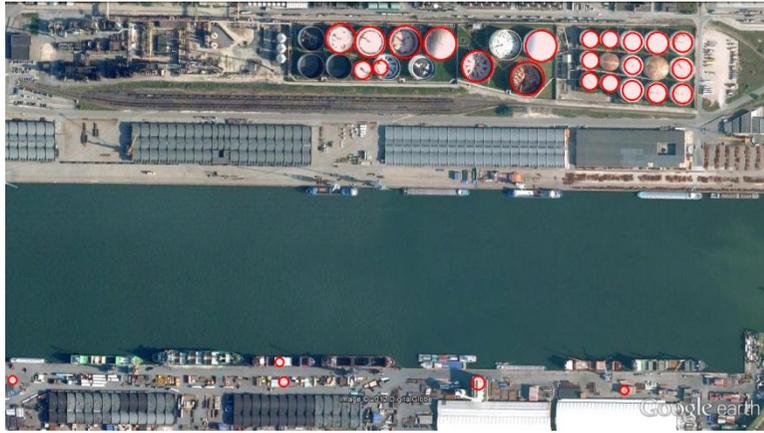


(b)



(c)

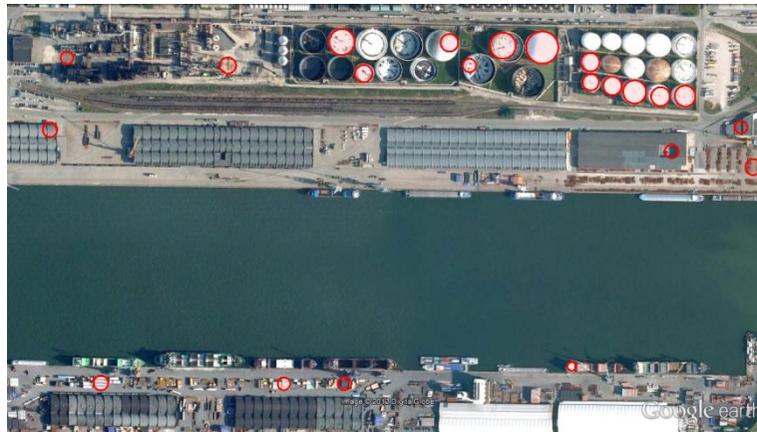
Figure 73 Recognized POIs on the image in Figure 68 using (a) MHT, (b) MHT CC and (c) RCD CC.



(a)



(b)



(c)

Figure 74 Recognized POLs on the image in Figure 68 using (a) MHT, (b) MHT CC and (c) RCD CC

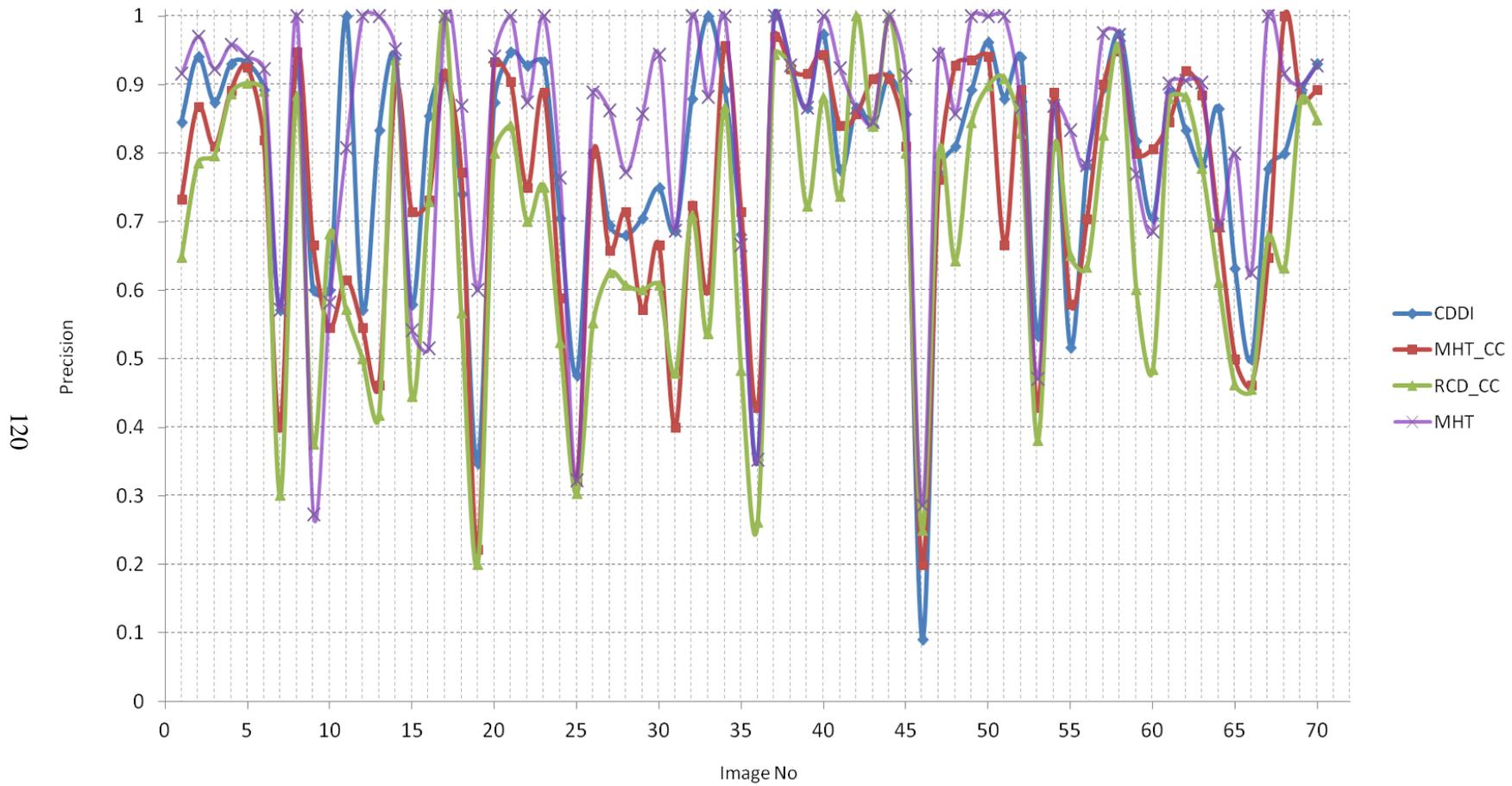


Figure 75 POL-wise precision values for each of the image in the data set when CDDI, MHT, MHT CC and RCD CC are used for detection of the circles.

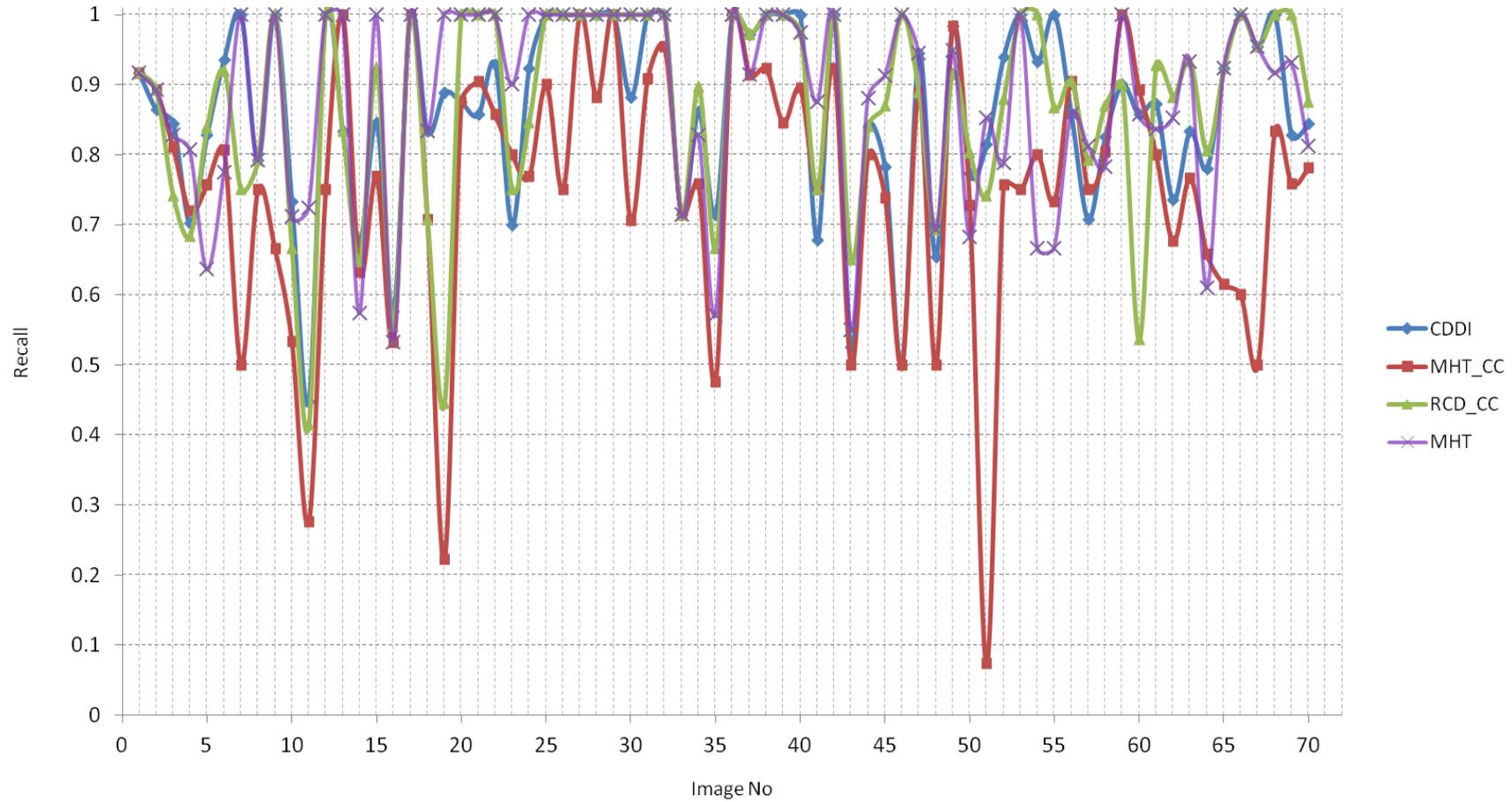


Figure 76 POL-wise recall values for each of the image in the data set when CDDI, MHT, MHT CC and RCD CC are used for detection of the circles.

5.1.8 Discussion

In this section an approach for recognition of POLs from high resolution satellite images is proposed (see Figure 60). This method includes all the steps given in Figure 1 and the details of each step are explained in the previous parts.

The performed experiments show that the proposed approach produce successful results and the size and diversity of the dataset used implies good fidelity of these results.

Originally CDDI is utilized for detection of the circles on the images, which is the most crucial step of the proposed POL recognition approach. The provided speed up and performance stability of CDDI is shown in the previous chapter and the experiments are also repeated while using MHT and RCD methods for detection of the circles through POL recognition. This chapter also shows that how practical CDDI can be for remote sensing applications due to its performance and low computational requirements.

Note that, these experiments are performed on a data set that includes limited type of POLs. On the other hand, there may exist different types of POLs because of its purpose or geographical features. For instance in Figure 77(a) there are brown POLs where in this study the POLs are assumed as white. As a result, the proposed approach missed half of the POLs in the scene (see Figure 77(b)). However, it is possible obtain another schema for detection of the other types of the POLs. A final step may be composing these methods to cover different types of POLs but the performance of the approach may be lower than the one presented in this study.

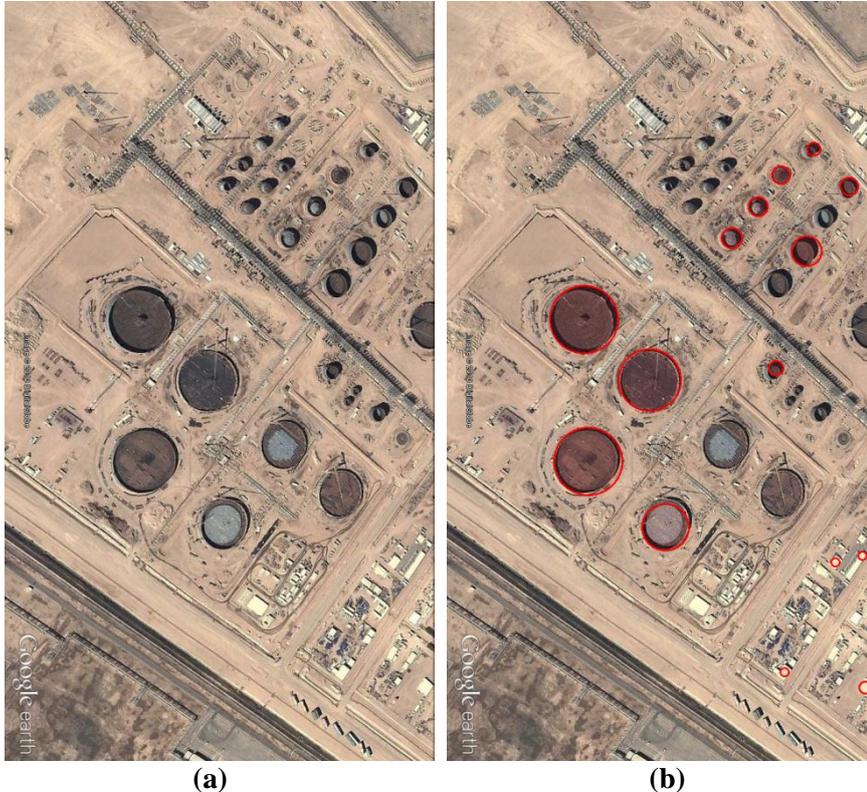


Figure 77 (a) An image containing dark colored POLs and (b) the detected POLs by the proposed algorithm using CDDI.

5.2 Detection of Rectangular Buildings in the Industrial Regions

In this section, an analysis is given for the detectability of the rectangular buildings in the industrial regions by use of the proposed rectangle detection algorithm, RDDI. For this purpose, the detected POLs in the previous section are removed from the pre-evaluated industry mask and RDDI is executed for the resultant masked image.

The analyses show that there are discontinuities on the edges of rectangular buildings in these industrial zones due to some reasons. First of all, since some of those buildings are in dark color, some edges become undetectable due to the existence shadows in two sides of the building. Similarly, there are asphalt or concrete roads around these building which lead undetected edge segments. Vegetation and vehicles (like trucks, crabs, etc.) around these buildings are another problem and they disturb the shape of the buildings. Finally, existence of buildings attached to each other may yield with another shape if the boundaries between them cannot be detected precisely by the edge detectors. These structures are also hardly detectable as a rectangle.

In Figure 78 and Figure 79, two examples for rectangle detection in the industrial regions are given. In these experiments, the parameters of RDDI are selected as $\alpha=0.65$, $\beta=0.65$, $\lambda=0.65$. Since the edges of rectangles are divided into many parts, n_n and n_p parameters are set as 6 and 11, respectively. As seen from those images, some of the buildings are detected but a better segmentation approach, rather than simple edge detection, for these structures are required since for many of them the edges are missing.

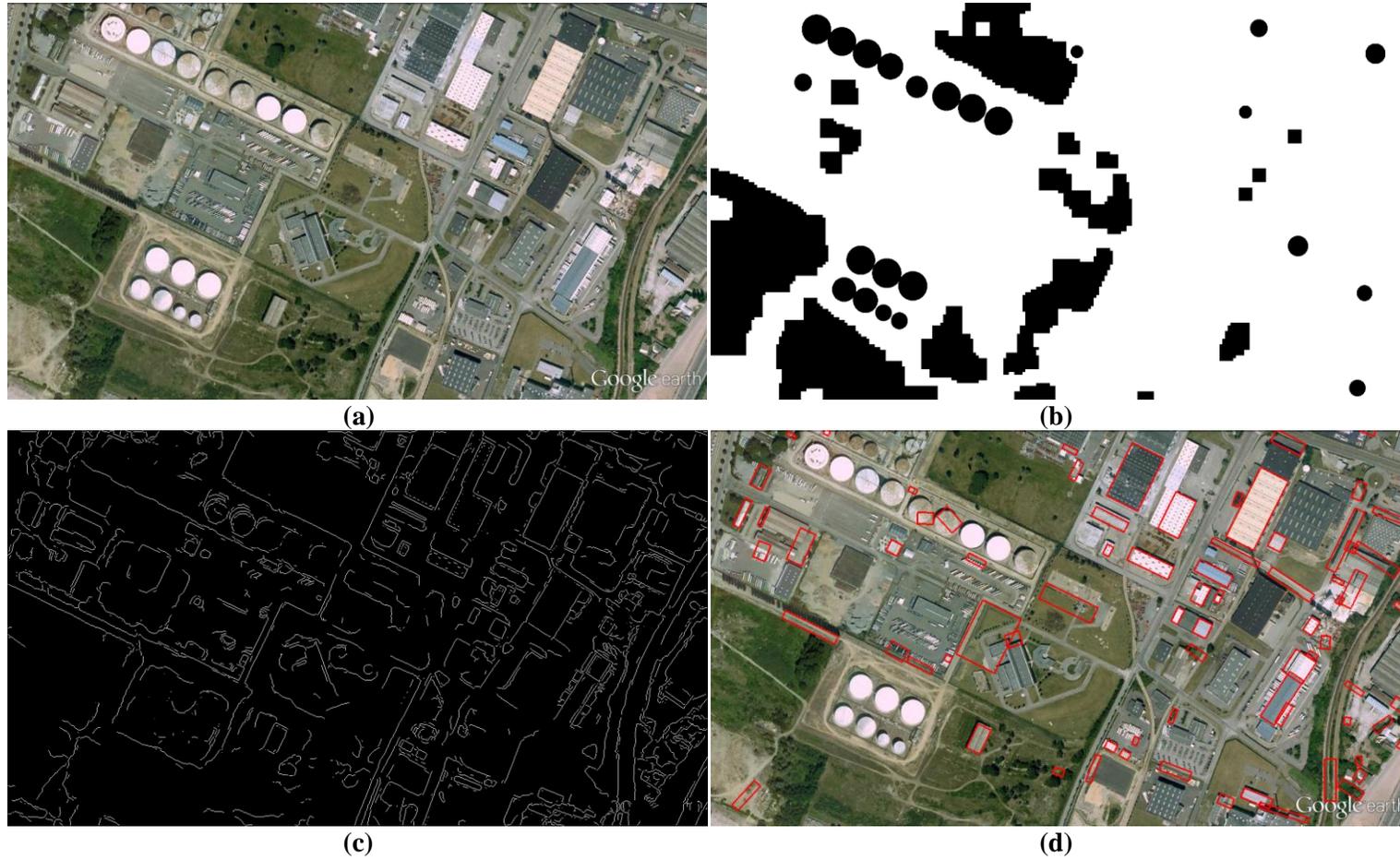


Figure 78 (a) An example image from the data set, (b) mask used for rectangle detection, (c) used edge map and (d) detected rectangles by RDDI.

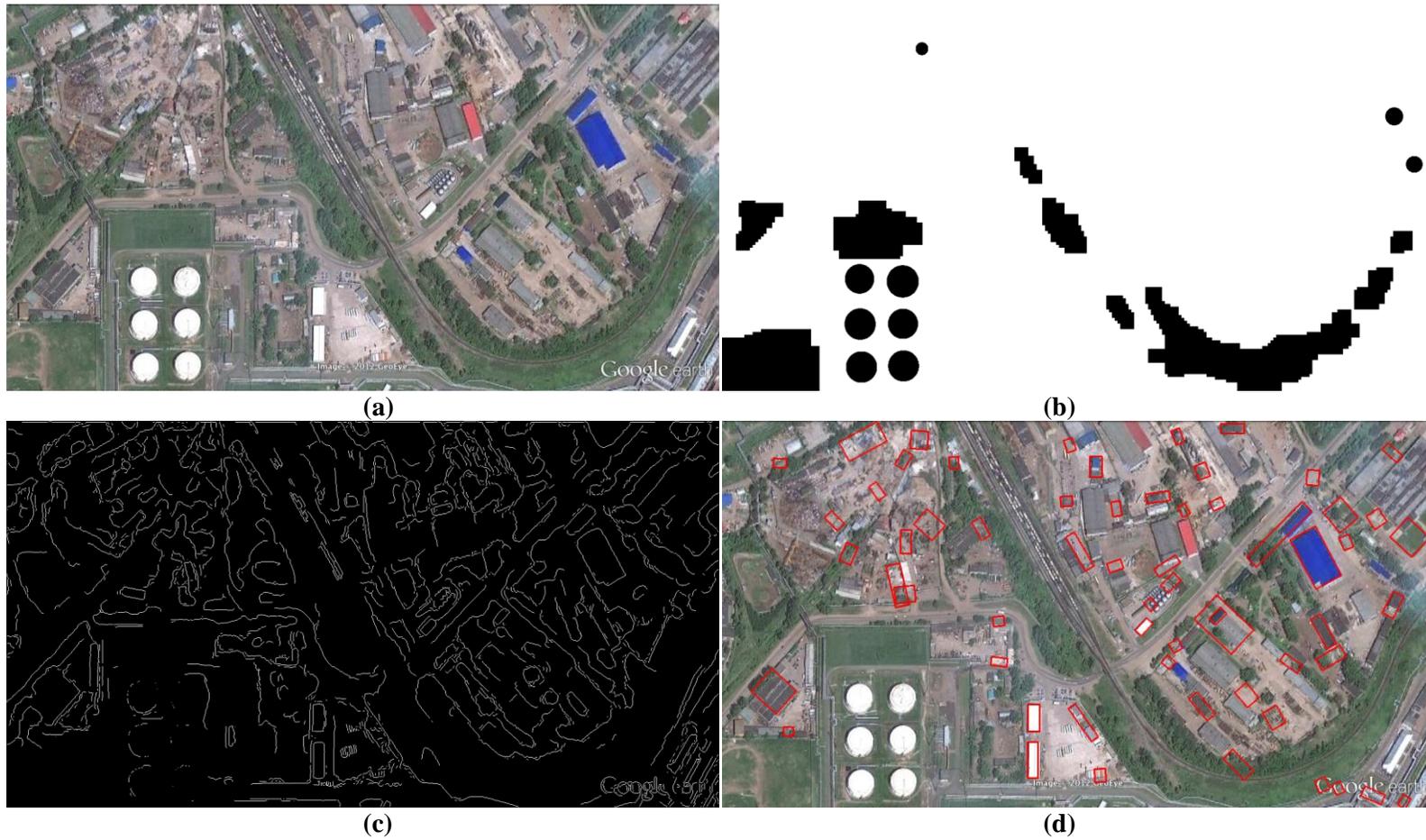


Figure 79 (a) An example image from the data set, (b) mask used for rectangle detection, (c) used edge map and (d) detected rectangles by RDDI.

5.2.1 Discussion

In this section, rectangular structures in the industrial regions are analyzed. By use of RDDI approach, the rectangles are detected in these regions. Despite, the buildings generally look like rectangles, the analysis show that there are very different effects as shadows, plants or other additional structures that disturbs their shapes a lot. Hence, it is not possible to detect all the buildings in the industrial zones only by use of RDDI approach. However, it is still possible to improve the performance of RDDI using another segmentation approach rather than simple edge detection.

CHAPTER 6

CONCLUSION

The main motivation of the thesis is to obtain a fast and accurate geometrical shape detection approach in order to utilize in processing of big data like remote sensing images. For this purpose, an algorithm is proposed which determines specific points on transformed domains by examining the extracted features in these domains and their similarities in between. Radon Transform is utilized to map the image into 1-D spaces and the analysis show that only two different transformations through two different angles are necessary to define a shape. As the dimension of the space is reduced, required computations are decreased remarkably with respect to the existing shape detection methods in the literature.

This method is examined in terms of detection of circular, rectangular and elliptic shapes and the comparisons with the state-of-art approaches in the literature are presented. The performance comparisons are achieved by use of both synthetic and real images where the outcomes are examined in terms of true detections, false detections and computation durations. Finally, the approach is discussed in terms of detection of any shape assuming the closed form representation of the shape is given.

The most detailed experiments are conducted for the circle detection case. The abilities of the proposed method, CDDI, are presented using different synthetic cases. The effects of the utilized edge detection approach on CDDI also examined and Canny edge detector found to be the best choice over the well known edge detectors because of its good local thresholding and its tendency for keeping the edges smooth and connected.

As the next step, CDDI is compared with Modified Hough Transform (MHT) and Randomized Circle Detection (RCD) algorithms. Previous studies in the literature show that Hough Transform based approaches are robust against many of the possible effects and the randomized approaches are quite beneficial to reduce the computation duration. The experiments performed show the detection performance of CDDI is comparable, and even better for some cases, with respect to these MHT and RCD. Moreover, MHT requires a priori knowledge about the radii of the circles in the scene and the parameter choice of RCD is sensitive from one image to another; where the parameters of CDDI do not depend on the scene characteristics. These methods are also compared in terms of their computational loads and it is shown that CDDI requires much less computation than the other methods, especially with respect to the Hough Transform based approaches.

The analyses also show that one of the reasons of the low computation load of CDDI is caused by the individual assessment of each object on the image. Then, MHT and RCD are modified to operate in connected components on the edge map of the images. However, the proposed approach is still faster than the other methods where its detection performance is better than the performance of the modified MHT and RCD algorithms.

Another outcome of this thesis is the observation of the increase in required computation power as the complexity of the images is increased for RCD approach. It is quite clear to generalize this conclusion for any other randomized approach since the probability of picking a few points on the same shape is reduced rapidly as the number of objects on the image increases.

The derivations for the proposed method to detect rectangles, RDDI, and ellipses, EDDI, are also presented in this thesis. The computational and detection performances are also analyzed for both cases and their robustness against noise are shown using synthetic images. Different kinds of images are used for representing its applicability to real world problems. Moreover, RDDI is compared with Windowed Hough Transform (WHT) in terms of detection performance and computational load. As in the circle detection case, the detection performance of RDDI is similar to WHT up to a certain noise level while the proposed method performs faster. The outputs of the real images also show that the proposed method is as successful as the Hough Transform based method.

In the second part of the thesis, a procedure for identification of geometrically shaped objects in the industrial regions using 3-band, 1m resolution satellite images is proposed. The proposed procedure includes almost every step of a general object identification procedure including land cover/use classification; pre-filtering, shape detection and post-processing for false target elimination. For land cover/use classification procedure a state-of-the-art procedure is adopted as this step is widely studied in the literature and modification of the step is out of scope of the thesis. However, different feature sets are examined and a good feature set is selected according to the data set used, after performed experiments. The pre-filtering and post-processing steps are defined according to the physical features of the target objects.

The first target defined is the POLs and the proposed POL identification procedure is another contribution of the thesis to the literature since, to the best of our knowledge, there is not any other procedure for POL identification in the literature. Besides, the proposed shape detection algorithm is utilized for detection of circles, which is the most crucial step of POL detection. The success of the proposed approach is verified by use of a large data set which is obtained using different satellites and different illumination conditions, including different types of POLs with various sizes. By the experiments performed promising results are obtained where the detection probability and the precision is above 80% for the POLs with radius of 7m or larger. These results also show the usability of the proposed shape detection approach in real world applications with a quite well performance. MHT and RCD are also used in this procedure as an alternative for circle detection step and the comparisons show that around 2.5 times speed up is achieved by CDDI while having very similar detection performances. Moreover, CDDI outperforms MHT and RCD in both detection performance and fastness while these methods are performed for each connected component of the image, independently.

The rectangular objects are also examined in the scope of last chapter and the analysis show that some of the buildings in the industrial zones can be detected by use of the proposed rectangle detection approach. On the other hand, due to the deformations on the rectangular shape due to shadows, vegetations or additional parts of the buildings, rectangle detection is not that proper to detect all the buildings on the image.

Future Directions

First of all, the shape detection approach can be extended for other geometrical shapes as discussed in the previous chapters. Besides, an automatic procedure can be developed which uses the profiles obtained by directional integrations and the relations between the critical points to actively learn the shape statistics in order to detect any complicated shape without using its closed form relation. Secondly, the edge detection schema used in the shape detection procedure can be specialized according to the interested shape to obtain more precise results.

Another improvement can be achieved in POL recognition by modifying the land cover/use classification approach. Despite, the adopted approach is quite promising results, a post classification can be made at the boundaries of the classes where most of the errors occur. Besides, utilization of the other spectral bands, i.e. near infrared, which are now provided by many of the satellite image providers, the classification results can be refined.

Lastly, fusion of the data obtained from other sensor modalities with wide band optical sensor data should be taken into account through the detection and recognition process of POLs. Once the registration with the utilized data is achieved, Synthetic Aperture Radar (SAR) images or hyperspectral data reduce the effort for detection and recognition of the POLs, with higher performance, like any other man-made structures on earth.

REFERENCES

- [1] R. A. Schowengerdt, *Remote sensing, models, and methods for image processing*, 2nd ed. San Diego: Academic Press, 1997.
- [2] N. Paparoditis, *et al.*, "Building detection and reconstruction from mid- and high-resolution aerial imagery," *Computer Vision and Image Understanding*, vol. 72, pp. 122-142, NOV 1998.
- [3] H. Mayer, "Automatic object extraction from aerial imagery - A survey focusing on buildings," *Computer Vision and Image Understanding*, vol. 74, pp. 138-149, MAY 1999.
- [4] H. Moon, *et al.*, "Optimal edge-based shape detection," *Ieee Transactions on Image Processing*, vol. 11, pp. 1209-1227, NOV 2002.
- [5] D. Mumford, *et al.*, "Discriminating Figure from Ground - the Role of Edge-Detection and Region Growing," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 84, pp. 7354-7358, Oct 1987.
- [6] P. V. C. Hough, "Method and means for recognizing complex patterns," 1962.
- [7] R. O. Duda and P. E. Hart, "Use of Hough Transformation to Detect Lines and Curves in Pictures," *Communications of the ACM*, vol. 15, pp. 11-&, 1972.
- [8] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, pp. 111-122, 1981.
- [9] E. R. Davies, *Machine vision : Theory, Algorithms, Practicalities*, 3rd ed. Amsterdam ; Boston: Elsevier, 2005.
- [10] H. K. Yuen, *et al.*, "Comparative-Study of Hough Transform Methods for Circle Finding," *Image and Vision Computing*, vol. 8, pp. 71-77, Feb 1990.
- [11] S. H. Chiu and H. H. Liaw, "An effective voting method for circle detection," *Pattern Recognition Letters*, vol. 26, pp. 121-133, JAN 15 2005.
- [12] D. Ioannou, *et al.*, "Circle recognition through a 2D Hough Transform and radius histogramming," *Image and Vision Computing*, vol. 17, pp. 15-26, JAN 1999.
- [13] H. S. Kim and J. H. Kim, "A two-step circle detection algorithm from the intersecting chords," *Pattern Recognition Letters*, vol. 22, pp. 787-798, May 2001.
- [14] R. K. K. Yip, *et al.*, "Line Segment Patterns Hough Transform for Circles Detection Using a 2-Dimensional Array," *Proceedings of the IECON 93 - International Conference on Industrial Electronics, Control, and Instrumentation, Vols 1-3*, pp. 1361-1365 2398, 1993.
- [15] R. K. K. Yip, *et al.*, "Modification of Hough Transform for Circles and Ellipses Detection Using a 2-Dimensional Array," *Pattern Recognition*, vol. 25, pp. 1007-1022, Sep 1992.
- [16] T. D'Orazio, *et al.*, "A new algorithm for ball recognition using circle Hough transform and neural classifier," *Pattern Recognition*, vol. 37, pp. 393-408, MAR 2004.

- [17] Q. Ji and Y. Xie, "Randomised hough transform with error propagation for line and circle detection," *Pattern Analysis and Applications*, vol. 6, pp. 55-64, Apr 2003.
- [18] C. Kimme, *et al.*, "Finding Circles by an Array of Accumulators," *Communications of the ACM*, vol. 18, pp. 120-122, 1975.
- [19] R. A. McLaughlin, "Randomized hough transform: Better ellipse detection.," *1996 IEEE Tencon - Digital Signal Processing Applications Proceedings, Vols 1 and 2*, pp. 409-414, 1996.
- [20] R. A. McLaughlin, "Randomized Hough Transform: Improved ellipse detection with comparison," *Pattern Recognition Letters*, vol. 19, pp. 299-305, Mar 1998.
- [21] C. F. Olson, "Constrained Hough transforms for curve detection," *Computer Vision and Image Understanding*, vol. 73, pp. 329-345, Mar 1999.
- [22] D. Walsh and A. E. Raftery, "Accurate and efficient curve detection in images: the importance sampling Hough transform," *Pattern Recognition*, vol. 35, pp. 1421-1431, Jul 2002.
- [23] L. Xu and E. Oja, "Randomized Hough Transform (RHT) - Basic Mechanisms, Algorithms, and Computational Complexities," *CVGIP-Image Understanding*, vol. 57, pp. 131-154, Mar 1993.
- [24] L. Xu, *et al.*, "A New Curve Detection Method - Randomized Hough Transform (Rht)," *Pattern Recognition Letters*, vol. 11, pp. 331-338, May 1990.
- [25] T. J. Atherton and D. J. Kerbyson, "Size invariant circle detection," *Image and Vision Computing*, vol. 17, pp. 795-803, SEP 1999.
- [26] A. A. Rad, *et al.*, "Fast Circle Detection Using Gradient Pair Vectors," in *VIIIth Digital Image Computing: Tech. and App.*, Sydney, 2003.
- [27] Y. M. Li and W. C. K. Wong, "An efficient and robust method to detect object centre," *Pattern Recognition*, vol. 30, pp. 659-671, May 1997.
- [28] C. T. Ho and L. H. Chen, "A Fast Ellipse/Circle Detector Using Geometric Symmetry," *Pattern Recognition*, vol. 28, pp. 117-124, Jan 1995.
- [29] T. C. Chen and K. L. Chung, "An efficient randomized algorithm for detecting circles," *Computer Vision and Image Understanding*, vol. 83, pp. 172-191, Aug 2001.
- [30] K. L. Chung and Y. H. Huang, "Speed up the computation of randomized algorithms for detecting lines, circles, and ellipses using novel tuning- and LUT-based voting platform," *Applied Mathematics and Computation*, vol. 190, pp. 132-149, Jul 1 2007.
- [31] R. J. Lee, *et al.*, "Robust and efficient automated detection of tooling defects in polished stone," *Computers in Industry*, vol. 56, pp. 787-801, Dec 2005.
- [32] K. L. Chung, *et al.*, "Efficient sampling strategy and refinement strategy for randomized circle detection," *Pattern Recognition*, vol. 45, pp. 252-263, Jan 2012.
- [33] T. Kawaguchi and R. Nagata, "Ellipse detection using a genetic algorithm," *Fourteenth International Conference on Pattern Recognition, Vols 1 and 2*, pp. 141-145, 1998.
- [34] V. Ayala-Ramirez, *et al.*, "Circle detection on images using genetic algorithms," *Pattern Recognition Letters*, vol. 27, pp. 652-657, Apr 15 2006.

- [35] E. Cuevas, *et al.*, "Circle Detection by Harmony Search Optimization," *Journal of Intelligent & Robotic Systems*, vol. 66, pp. 359-376, May 2012.
- [36] E. Cuevas, *et al.*, "Circle detection using discrete differential evolution optimization," *Pattern Analysis and Applications*, vol. 14, pp. 93-107, Feb 2011.
- [37] E. Cuevas, *et al.*, "Multi-circle detection on images using artificial bee colony (ABC) optimization," *Soft Computing*, vol. 16, pp. 281-296, Feb 2012.
- [38] C. Akinlar and C. Topal, "EDCircles: A Real-Time Circle Detector with a False Detection Control," *Pattern Recognition*, vol. 46, pp. 725-740, Mar 2013.
- [39] I. Frosio and N. Borghese, "Real-time accurate circle fitting with occlusions," *Pattern Recognition*, vol. 41, pp. 1041-1055, Mar 2008.
- [40] C. Akinlar and C. Topal, "Edpf: A Real-Time Parameter-Free Edge Segment Detector with a False Detection Control," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 26, Feb 2012.
- [41] H. H. Peng and R. Rao, "A novel circle detection method using Radon Transform," *Image Processing: Machine Vision Applications*, vol. 6813, 2008.
- [42] J. Canny, "A Computational Approach to Edge-Detection," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-698, Nov 1986.
- [43] V. Y. Mariano, *et al.*, "Performance evaluation of object detection algorithms," *16th International Conference on Pattern Recognition, Vol III, Proceedings*, pp. 965-969, 2002.
- [44] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Sixth Int. Conference on Computer Vision*, pp. 839-846, 1998.
- [45] D. Lagunovsky and S. Ablameyko, "Rectangle-shaped object detection in aerial images," *Visual Communications and Image Processing '95, Pts 1-3*, vol. 2501, pp. 1566-1574, 1995.
- [46] D. Lagunovsky and S. Ablameyko, "Straight-line-based primitive extraction in grey-scale object recognition," *Pattern Recognition Letters*, vol. 20, pp. 1005-1014, Oct 1999.
- [47] C. Lin and R. Nevatia, "Building detection and description from a single intensity image," *Computer Vision and Image Understanding*, vol. 72, pp. 101-121, NOV 1998.
- [48] W. B. Tao, *et al.*, "A new approach to extract rectangle building from aerial urban images," *2002 6th International Conference on Signal Processing Proceedings, Vols I and II*, pp. 143-146, 2002.
- [49] Y. X. Zhu, *et al.*, "Automatic particle detection through efficient Hough transforms," *Ieee Transactions on Medical Imaging*, vol. 22, pp. 1053-1062, Sep 2003.
- [50] Y. X. Zhu, *et al.*, "Fast detection of generic biological particles in CRYO-EM images through efficient hough transforms," *2002 Ieee International Symposium on Biomedical Imaging, Proceedings*, pp. 205-208, 2002.
- [51] Y. X. Zhu, *et al.*, "Automated identification of filaments in cryoelectron microscopy images," *Journal of Structural Biology*, vol. 135, pp. 302-312, Sep 2001.

- [52] C. R. Jung and R. Schramm, "Rectangle detection based on a windowed Hough Transform," *Xvii Brazilian Symposium on Computer Graphics and Image Processing, Proceedings*, pp. 113-120 405, 2004.
- [53] W. Zhe, *et al.*, "A Rectangle Detection Method for Real-Time Extraction of Large Panel Edge," in *Image and Graphics (ICIG), 2011 Sixth International Conference on*, 2011, pp. 382-387.
- [54] Y. Liu, *et al.*, "An MRF model-based approach shape objects in to the detection of rectangular color images," *Signal Processing*, vol. 87, pp. 2649-2658, Nov 2007.
- [55] L. G. Shapiro and G. C. Stockman, *Computer vision*. Upper Saddle River, NJ: Prentice Hall, 2001.
- [56] D. Chaudhuri and A. Samal, "A simple method for fitting of bounding rectangle to closed regions," *Pattern Recognition*, vol. 40, pp. 1981-1989, Jul 2007.
- [57] A. Pietrowcew, "Face detection in colour images using fuzzy Hough transform," *Opto-Electronics Review*, vol. 11, pp. 247-251, Sep 2003.
- [58] P. H. Li, *et al.*, "Robust and accurate iris segmentation in very noisy iris images," *Image and Vision Computing*, vol. 28, pp. 246-253, Feb 2010.
- [59] J. R. Kim, *et al.*, "Automated crater detection, a new tool for Mars cartography and chronology," *Photogrammetric Engineering and Remote Sensing*, vol. 71, pp. 1205-1217, Oct 2005.
- [60] C. R. Zhu and R. S. Wang, "A fast automatic extraction algorithm of elliptic object groups from remote sensing images," *Pattern Recognition Letters*, vol. 25, pp. 1471-1478, Oct 1 2004.
- [61] A. S. Aguado, *et al.*, "Ellipse detection via gradient direction in the Hough transform," in *Image Processing and its Applications, 1995., Fifth International Conference on*, 1995, pp. 375-378.
- [62] K. C. Chen, *et al.*, "Quantization-free parameter space reduction in ellipse detection," *Expert Systems with Applications*, vol. 38, pp. 7622-7632, Jun 2011.
- [63] Z. G. Cheng and Y. C. Liu, "Efficient technique for ellipse detection using Restricted Randomized Hough Transform," *Itcc 2004: International Conference on Information Technology: Coding and Computing, Vol 2, Proceedings*, pp. 714-718, 2004.
- [64] Y. H. Xie and Q. Ji, "A new efficient ellipse detection method," *16th International Conference on Pattern Recognition, Vol Ii, Proceedings*, pp. 957-960, 2002.
- [65] A. Y. S. Chia, *et al.*, "Ellipse detection with Hough Transform in one dimensional parametric space," *2007 Ieee International Conference on Image Processing, Vols 1-7*, pp. 2585-2588, 2007.
- [66] Y. W. Lei and K. C. Wong, "Ellipse detection based on symmetry," *Pattern Recognition Letters*, vol. 20, pp. 41-47, Jan 1999.
- [67] D. K. Prasad, *et al.*, "Edge curvature and convexity based ellipse detection method," *Pattern Recognition*, vol. 45, pp. 3204-3221, Sep 2012.
- [68] Q. Ji and R. M. Haralick, "A statistically efficient method for ellipse detection," in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on*, 1999, pp. 730-734.
- [69] Z. Y. Liu, *et al.*, "Multisets mixture learning-based ellipse detection," *Pattern Recognition*, vol. 39, pp. 731-735, Apr 2006.

- [70] J. B. Burns, *et al.*, "Extracting Straight-Lines," *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 425-455, Jul 1986.
- [71] I. S. Gradshteiñ, *et al.*, *Table of integrals, series, and products*: Academic press, 2007.
- [72] (26 January 2013). *Ellipse*. Available: <http://en.wikipedia.org/wiki/Ellipse>
- [73] J. A. Richards and X. Jia, *Remote sensing digital image analysis : an introduction*, 4th ed. Berlin: Springer, 2006.
- [74] X. M. Zhang, *et al.*, "A method to evaluate the capability of Landsat-5 TM band 6 data for sub-pixel coal fire detection," *International Journal of Remote Sensing*, vol. 18, pp. 3279-3288, OCT 1997.
- [75] A. Prakash and R. P. Gupta, "Land-use mapping and change detection in a coal mining area - a case study in the Jharia coalfield, India," *International Journal of Remote Sensing*, vol. 19, pp. 391-410, FEB 1998.
- [76] A. Filippidis, *et al.*, "Multisensor data fusion for surface land-mine detection," *Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, vol. 30, pp. 145-150, FEB 2000.
- [77] R. A. Almeida and Y. E. Shimabukuro, "Detecting areas disturbed by gold mining activities through JERS-1 SAR images, Roraima State, Brazilian Amazon," *International Journal of Remote Sensing*, vol. 21, pp. 3357-3362, NOV 2000.
- [78] S. Voigt, *et al.*, "Integrating satellite remote sensing techniques for detection and analysis of uncontrolled coal seam fires in North China," *International Journal of Coal Geology*, vol. 59, pp. 121-136, JUL 12 2004.
- [79] R. S. Gautam, *et al.*, "Application of SVM on satellite images to detect hotspots in Jharia coal field region of India," *Advances in Space Research*, vol. 41, pp. 1784-1792, 2008.
- [80] S. C. Mularz, "Satellite and Airborne Remote Sensing Data for Monitoring of an open-Cats Mine," in *ISPRS Commission IV Symposium on GIS- Between Visions and Applications, 'IAPRS', CIS Between Visions and Applications*, Stuttgart, 1998.
- [81] W. Yanqing, *et al.*, "Detection of circular oil tanks based on the fusion of SAR and optical images," in *Image and Graphics, 2004. Proceedings. Third International Conference on*, 2004, pp. 524-527.
- [82] E. P. Baltsavias, *et al.*, *Automatic extraction of man-made objects from aerial and space images (III)*. Lisse ; Exton, PA: A.A. Balkema, 2001.
- [83] Z. Zhang, *et al.*, "Semi-automatic building extraction from stereo image pairs," *Automatic Extraction of Man-Made Objects from Aerial and Space Images (Iii)*, pp. 115-122 415, 2001.
- [84] T. Yoon, *et al.*, "Semi-automatic road extraction from IKONOS satellite images," *Remote Sensing for Environmental Monitoring, Gis Applications, and Geology*, vol. 4545, pp. 320-328 330, 2002.
- [85] S. Udomhunsakul, *et al.*, "Semi-automatic road extraction from aerial images," *Remote Sensing for Environmental Monitoring, Gis Applications, and Geology Iii*, vol. 5239, pp. 26-32 562, 2004.
- [86] F. Leymarie, *et al.*, "Towards the automation of road networks extraction processes," *Remote Sensing for Geography, Geology, Land Planning, and Cultural Heritage*, vol. 2960, pp. 84-95 270, 1996.

- [87] N. Jiang, *et al.*, "Semi-Automatic Building Extraction from High Resolution Imagery Based on Segmentation," *2008 International Workshop on Earth Observation and Remote Sensing Applications*, pp. 125-129 390, 2008.
- [88] X. Y. Hu, *et al.*, "A robust method for semi-automatic extraction of road centerlines using a piecewise parabolic model and least square template matching," *Photogrammetric Engineering and Remote Sensing*, vol. 70, pp. 1393-1398, DEC 2004.
- [89] E. Gulch and H. Muller, "New applications of semi-automatic building acquisition," *Automatic Extraction of Man-Made Objects from Aerial and Space Images (Iii)*, pp. 103-114 415, 2001.
- [90] A. Gruen and H. H. Li, "Semi-automatic road extraction as a model driven optimization procedure," *Digital Photogrammetry and Remote Sensing '95, Proceedings*, vol. 2646, pp. 160-171 318, 1995.
- [91] I. Couloigner and T. Ranchin, "Semi-automatic extraction of urban road network: Assessment of the quality," *Operational Remote Sensing for Sustainable Development*, pp. 309-314 497, 1999.
- [92] R. Collins, *et al.*, "Automatic Extraction of Buildings and Terrain from Aerial Images," in *International Workshop on Automatic Extraction of Man-Made Objects from Aerial and Space Images*, ed, 1995, pp. 169-178.
- [93] M. Fradkin, *et al.*, "Building detection from multiple aerial images in dense urban areas," *Computer Vision and Image Understanding*, vol. 82, pp. 181-207, JUN 2001.
- [94] M. Gerke, *et al.*, "Building extraction from aerial imagery using a generic scene model and invariant geometric moments," *Ieee/Isprs Joint Workshop on Remote Sensing and Data Fusion over Urban Areas*, pp. 85-89 345, 2001.
- [95] S. Müller and D. Zaum, "Robust Building Detection in Aerial Images," in *CMRT05. IAPRS*, Vienna, Austria, 2005.
- [96] J. Peng, *et al.*, "An improved snake model for building detection from urban aerial images," *Pattern Recognition Letters*, vol. 26, pp. 587-595, APR 2005.
- [97] F. Rottensteiner and J. Jansa, "Automatic extraction of buildings from lidar data and aerial images," in *In ISPRS Commission IV, Symposium 2002 July 9 - 12, 2002, Ottawa*, ed, p. 2002.
- [98] M. Gerke, *et al.*, "Automatic Detection of Buildings and Trees from Aerial Imagery Using Different Levels of Abstraction," ed.
- [99] T. J. Kim and J. P. Muller, "Development of a graph-based approach for building detection," *Image and Vision Computing*, vol. 17, pp. 3-14, JAN 1999.
- [100] M. M. Trivedi and J. C. Bezdek, "Low-Level Segmentation of Aerial Images with Fuzzy Clustering," *Ieee Transactions on Systems Man and Cybernetics*, vol. 16, pp. 589-598, JUL-AUG 1986.
- [101] H. D. Cheng, *et al.*, "Color image segmentation: advances and prospects," *Pattern Recognition*, vol. 34, pp. 2259-2281, DEC 2001.
- [102] S. E. Franklin, *et al.*, "Texture analysis of IKONOS panchromatic data for Douglas-fir forest age class separability in British Columbia," *International Journal of Remote Sensing*, vol. 22, pp. 2627-2632, Sep 10 2001.

- [103] M. Tuceryan and A. K. Jain, "Texture analysis," in *Handbook of pattern recognition and computer vision*, C. H. Chen, *et al.*, Eds., ed: World Scientific Publishing Co., Inc., 1993, pp. 235-276.
- [104] Q. Chen, *et al.*, "Fast and active texture segmentation based on orientation and local variance," *Journal of Visual Communication and Image Representation*, vol. 18, pp. 119-129, APR 2007.
- [105] V. K. Madasu and P. Yarlagadda, "An in Depth Comparison of Four Texture Segmentation Methods," in *Digital Image Computing Techniques and Applications, 9th Biennial Conference of the Australian Pattern Recognition Society on*, 2007, pp. 366-372.
- [106] S. E. Grigorescu, *et al.*, "Comparison of texture features based on Gabor filters," *Image Processing, IEEE Transactions on*, vol. 11, pp. 1160-1167, 2002.
- [107] O. Pichler, *et al.*, "A comparison of texture feature extraction using adaptive Gabor filtering, pyramidal and tree structured wavelet transforms," *Pattern Recognition*, vol. 29, pp. 733-742, May 1996.
- [108] D. A. Clausi, "Comparison and fusion of co-occurrence, Gabor and MRF texture features for classification of SAR sea-ice imagery," *Atmosphere-Ocean*, vol. 39, pp. 183-194, Sep 2001.
- [109] D. G. Vince, *et al.*, "Comparison of texture analysis methods for the characterization of coronary plaques in intravascular ultrasound images," *Computerized Medical Imaging and Graphics*, vol. 24, pp. 221-229, Jul-Aug 2000.
- [110] R. Lumia, *et al.*, "Texture Analysis of Aerial Photographs," *Pattern Recognition*, vol. 16, pp. 39-46, 1983.
- [111] H. H. Nguyen and P. Cohen, "A Fuzzy Clustering Approach to Texture Segmentation," *1990 Ieee International Conference on Systems, Man, and Cybernetics*, pp. 200-205 929, 1990.
- [112] T. Caelli and D. Reye, "On the Classification of Image Regions by Color, Texture and Shape," *Pattern Recognition*, vol. 26, pp. 461-470, APR 1993.
- [113] M. P. Dubuisson-Jolly and A. Gupta, "Color and texture fusion: application to aerial image segmentation and GIS updating," *Image and Vision Computing*, vol. 18, pp. 823-832, JUL 2000.
- [114] A. Lorette, *et al.*, "Texture analysis through a Markovian modelling and fuzzy classification: Application to urban area extraction from satellite images," *International Journal of Computer Vision*, vol. 36, pp. 221-236, FEB 2000.
- [115] S. W. Myint, "Fractal approaches in texture analysis and classification of remotely sensed data: comparisons with spatial autocorrelation techniques and simple descriptive statistics," *International Journal of Remote Sensing*, vol. 24, pp. 1925-1947, May 10 2003.
- [116] C. A. Coburn and A. C. B. Roberts, "A multiscale texture analysis procedure for improved forest stand classification," *International Journal of Remote Sensing*, vol. 25, pp. 4287-4308, Oct 2004.
- [117] S. K. Ge, *et al.*, "Texture analysis for mapping *Tamarix parviflora* using aerial photographs along the Cache Creek, California," *Environmental Monitoring and Assessment*, vol. 114, pp. 65-83, Mar 2006.
- [118] Y. Xu and E. C. Uberbacher, "2D image segmentation using minimum spanning trees," *Image and Vision Computing*, vol. 15, pp. 47-57, JAN 1997.

- [119] R. O. Duda, *et al.*, *Pattern classification*, 2nd ed. New York: Wiley, 2001.
- [120] F. Xu, *et al.*, "Aerial images segmentation based on SVM," *2003 International Conference on Machine Learning and Cybernetics, Vols 1-5, Proceedings*, pp. 2207-2211 3268, 2003.
- [121] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *Ieee Transactions on Geoscience and Remote Sensing*, vol. 42, pp. 1778-1790, AUG 2004.
- [122] C. Huang, *et al.*, "An assessment of support vector machines for land cover classification," *International Journal of Remote Sensing*, vol. 23, pp. 725-749, FEB 2002.
- [123] P. Mitra, *et al.*, "Segmentation of multispectral remote sensing images using active support vector machines," *Pattern Recognition Letters*, vol. 25, pp. 1067-1074, JUL 2 2004.
- [124] R. M. Haralick, *et al.*, "Textural Features for Image Classification," *Ieee Transactions on Systems Man and Cybernetics*, vol. SMC3, pp. 610-621, 1973.
- [125] M. Hauta-Kasari, *et al.*, "Generalized co-occurrence matrix for multispectral texture analysis," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, 1996, pp. 785-789 vol.2.
- [126] R. Jain, *et al.*, *Machine vision*. New York: McGraw-Hill, 1995.
- [127] C. Chang and C. Lin, *LIBSVM: a library for support vector machines*, 2001.
- [128] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," *Ieee Transactions on Neural Networks*, vol. 13, pp. 415-425, MAR 2002.
- [129] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, p. 27, 2011.
- [130] P. H. Wang, "Pattern-Recognition with Fuzzy Objective Function Algorithms - Bezdek,Jc," *Siam Review*, vol. 25, pp. 442-442, 1983.

APPENDIX A

EVALUATION OF PROFILE FUNCTIONS FOR ROTATED ELLIPSES

The equation of an ellipse was given in Equation (105). The resultant horizontal and vertical profiles were given in 4.1 when the orientation of the ellipse, θ , is 0° . In this part, generalized versions of these profiles $g_1(x)$ and $g_2(y)$ will be derived valid for all values of θ .

When Equation (105) is rewritten Equation (163) is obtained and Equation (164) can be derived from this equation.

$$\frac{[(x-x_c)\cos\theta-(y-y_c)\sin\theta]^2}{a^2} + \frac{[(x-x_c)\sin\theta-(y-y_c)\cos\theta]^2}{b^2} = 1 \quad (163)$$

$$\begin{aligned} (y-y_c)^2 \left[\frac{\sin^2\theta}{a^2} + \frac{\cos^2\theta}{b^2} \right] + (y-y_c)(x-x_c) \left[\frac{\sin 2\theta}{b^2} - \frac{\sin 2\theta}{a^2} \right] \\ + (x-x_c)^2 \left[\frac{\cos^2\theta}{a^2} + \frac{\sin^2\theta}{b^2} \right] - 1 = 0 \end{aligned} \quad (164)$$

Once the following variables are defined, Equation (164) can be rewritten in the form of Equation (166).

$$\begin{aligned} k_1 &= \left[\frac{\sin^2\theta}{a^2} + \frac{\cos^2\theta}{b^2} \right] \\ k_2 &= (x-x_c) \left[\frac{\sin 2\theta}{b^2} - \frac{\sin 2\theta}{a^2} \right] \\ k_3 &= (x-x_c)^2 \left[\frac{\cos^2\theta}{a^2} + \frac{\sin^2\theta}{b^2} \right] - 1 \end{aligned} \quad (165)$$

$$k_1(y-y_c)^2 + k_2(y-y_c) + k_3 = 0 \quad (166)$$

Then, y is found as:

$$y = y_c - k_2 \pm \frac{\sqrt{\Delta}}{2k_1} \quad (167)$$

$$\text{where } \Delta = k_2^2 - 4k_1k_3$$

by which the derivative of y w.r.t. x is found as:

$$\frac{dy}{dx} = -\frac{dk_2}{dx} \pm \frac{\frac{k_1}{\sqrt{\Delta}} \frac{d\Delta}{dx} - 2\sqrt{\Delta} \frac{dk_1}{dx}}{(2k_1)^2} \quad (168)$$

Now, let simplify Δ in order to simplify the further operations as given below:

$$\Delta = (x-x_c)^2 \left(\frac{\sin 2\theta}{b^2} - \frac{\sin 2\theta}{a^2} \right)^2 - 4 \left(\frac{\sin^2 \theta}{a^2} + \frac{\cos^2 \theta}{b^2} \right) \left[(x-x_c)^2 \left(\frac{\cos^2 \theta}{a^2} + \frac{\sin^2 \theta}{b^2} \right) - 1 \right] \quad (169)$$

$$\Delta = (x-x_c)^2 \left[-\frac{2\sin^2 2\theta}{a^2 b^2} - \frac{4}{a^2 b^2} (\sin^4 \theta + \cos^4 \theta) \right] + 4 \left(\frac{\sin^2 \theta}{a^2} + \frac{\cos^2 \theta}{b^2} \right) \quad (170)$$

$$\Delta = (x-x_c)^2 \left[-\frac{(2\sin^2 \theta + 2\cos^2 \theta)^2}{a^2 b^2} \right] + 4 \left(\frac{b^2 \sin^2 \theta + a^2 \cos^2 \theta}{a^2 b^2} \right) \quad (171)$$

$$\Delta = -\frac{4}{a^2 b^2} (x-x_c)^2 + 4 \left(\frac{b^2 \sin^2 \theta + a^2 \cos^2 \theta}{a^2 b^2} \right) = -\frac{4}{a^2 b^2} \left((x-x_c)^2 - b^2 \sin^2 \theta - a^2 \cos^2 \theta \right) \quad (172)$$

When the derivatives in Equation (168) are evaluated:

$$\begin{aligned} \frac{d\Delta}{dx} &= -\frac{8(x-x_c)}{a^2 b^2} \\ \frac{dk_1}{dx} &= 0 \\ \frac{dk_2}{dx} &= \frac{\sin 2\theta}{a^2 b^2} (a^2 - b^2) \end{aligned} \quad (173)$$

Then, Equation (168) can be rewritten as:

$$\frac{dy}{dx} = -\frac{dk_2}{dx} \pm \frac{1}{2k_1 \sqrt{\Delta}} \frac{d\Delta}{dx} \quad (174)$$

which is equal to:

$$\frac{dy}{dx} = \frac{\sin 2\theta}{a^2 b^2} (b^2 - a^2) \mp \frac{2ab(x-x_c)}{(a^2 \cos^2 \theta + b^2 \sin^2 \theta) \sqrt{a^2 \cos^2 \theta + b^2 \sin^2 \theta - (x-x_c)^2}} \quad (175)$$

Note that, the derivative in the equation above has two different values, since ellipse has a second order equation, which makes the shape closed, hence there are two y (say y_{1k} and y_{2k}) values corresponding to each x_k values. As a result, the slopes of the curve at (x_k, y_{1k})

and (x_k, y_{2k}) are different from each other as given in Equation (175). Then $g_1(x)$ is found as below by use of Equation (18):

$$g_1(x) = \sqrt{1 + \left(\frac{dy}{dx} \Big|_{y=y_1} \right)^2} + \sqrt{1 + \left(\frac{dy}{dx} \Big|_{y=y_2} \right)^2} \quad (176)$$

To sum up, the vertical profile of an ellipse is found as:

$$g_1(x) = \sqrt{1 + (T_{1x} + T_{2x})^2} + \sqrt{1 + (T_{1x} - T_{2x})^2} \quad (177)$$

where,

$$T_{1x} = \frac{\sin 2\theta}{a^2 b^2} (a^2 - b^2) \quad (178)$$

$$T_{2x} = \frac{2ab(x - x_c)}{(a^2 \cos^2 \theta + b^2 \sin^2 \theta) \sqrt{a^2 \cos^2 \theta + b^2 \sin^2 \theta - (x - x_c)^2}} \quad (179)$$

By a similar approach, $g_2(y)$ can be found as in Equation (181) after evaluating $\frac{dx}{dy}$ as:

$$\frac{dx}{dy} = \frac{\sin 2\theta}{a^2 b^2} (b^2 - a^2) \mp \frac{2ab(y - y_c)}{(a^2 \sin^2 \theta + b^2 \cos^2 \theta) \sqrt{a^2 \sin^2 \theta + b^2 \cos^2 \theta - (y - y_c)^2}} \quad (180)$$

$$g_2(y) = \sqrt{1 + (T_{1y} + T_{2y})^2} + \sqrt{1 + (T_{1y} - T_{2y})^2} \quad (181)$$

where,

$$T_{1y} = \frac{\sin 2\theta}{a^2 b^2} (a^2 - b^2) \quad (182)$$

$$T_{2y} = \frac{2ab(y - y_c)}{(a^2 \sin^2 \theta + b^2 \cos^2 \theta) \sqrt{a^2 \sin^2 \theta + b^2 \cos^2 \theta - (y - y_c)^2}} \quad (183)$$

APPENDIX B

FUZZY C-MEANS CLUSTERING

Fuzzy C-Means (FCM) clustering is a modification of the c-means clustering approach. In both methods, the aim is to find c mean vectors which indicate the centers of the clusters, [119]. In c-means approach, the assumption is that the distance (i.e. Mahalanobis distance) between an element and the corresponding cluster center is small. Hence, the probability of this element being in that cluster is almost 1 where the probability of this element in another cluster is almost 0. This statement can be expressed as follows if w_i is defined as being in the i^{th} cluster and c_i is the center of the i^{th} cluster:

$$P(w_i | x_j) = u_{ij} = \begin{cases} 1, & \text{if } |x_j - c_i| = \min \{x_j - \bar{c}\} \\ 0, & \text{otherwise} \end{cases} \quad (184)$$

where u_{ij} given in the above expression is called the membership of the j^{th} sample to the i^{th} cluster. Hence, in the c-means clustering method each sample is a member of only one of the c clusters under examination.

On the other hand, in FCM clustering approach the probabilities given in Equation (184) may be nonzero for more than one cluster. In other words, each sample may be a member of multiple clusters which yields a more generalized approach. In this case the following criterion must be satisfied:

$$\sum_i u_{ij} = 1 \quad (185)$$

The membership values are calculated according to the distance of the sample to the center of the clusters. As a result, the FCM algorithm attempts to cluster measurement vectors by searching for local minima of the following generalized within group sum of squared (WGSS) error objective function [130], according to the Equation (184):

$$J = \sum_{i=1}^c \sum_{j=1}^n (u_{ij})^m \|x_j - \mu_i\|_A^2 \quad (186)$$

$\|\cdot\|_A$ operation is any inner product norm and Euclidian norm is used in this project. Using the Lagrange multiplier method the results for membership values and cluster centers are found as:

$$u_{ij} = \left(\sum_{k=1}^c \left(\frac{\|x_j - \mu_i\|}{\|x_j - \mu_k\|} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (187)$$

$$\mu_i = \frac{\sum_{j=1}^n x_j (u_{ij})^m}{\sum_{j=1}^n (u_{ij})^m} \quad (188)$$

As seen from Equations (187) and (188) the solution can only be an iterative algorithm which is summarized as:

- Initialize u_{ij} 's, i.e. u_{ij}^0
- Calculate fuzzy cluster centers, μ_i^p using u_{ij}^{p-1}
- Update u_{ij} 's to form u_{ij}^p using μ_i^p
- Calculate the difference in μ_i^p and μ_i^{p-1} ; u_{ij}^p and u_{ij}^{p-1} to check convergence
- Repeat the procedure until convergence

APPENDIX C

DATA SET USED IN POL DETECTION

In the following figures, the images captured from Google Earth™ to evaluate the performance of POL recognition procedure and corresponding ground truths are presented. In this data set, there are 70 images at 1m of resolution, having 1991 POLs from 17 different industrial zones of 12 different countries with various radii between 7 and 55 meters and all images are in 1280x717 of size.

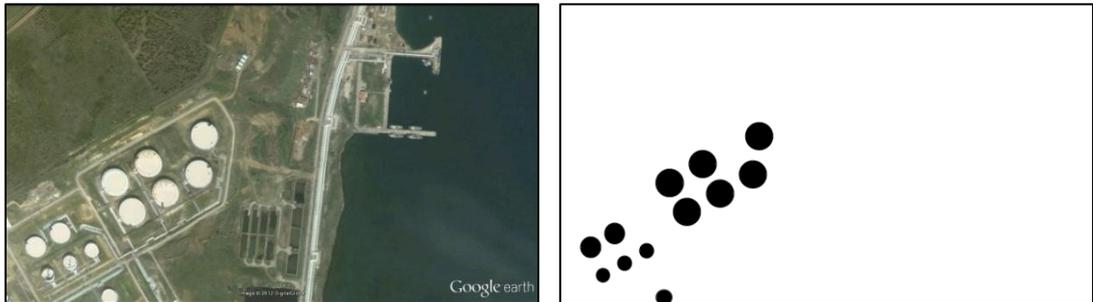


Figure C. 1



Figure C. 2

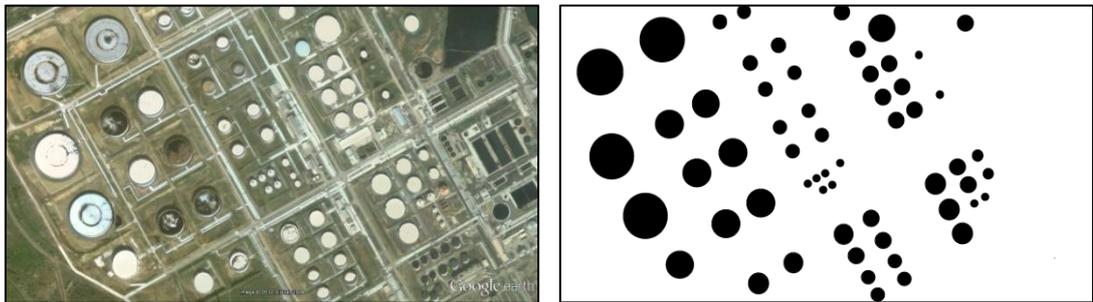


Figure C. 3

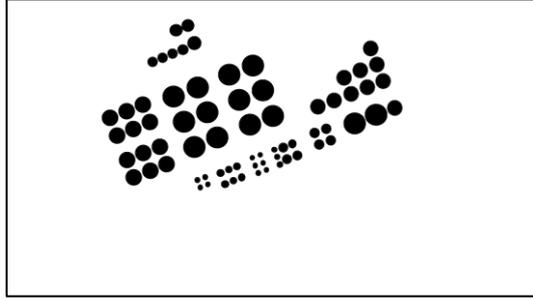


Figure C. 4

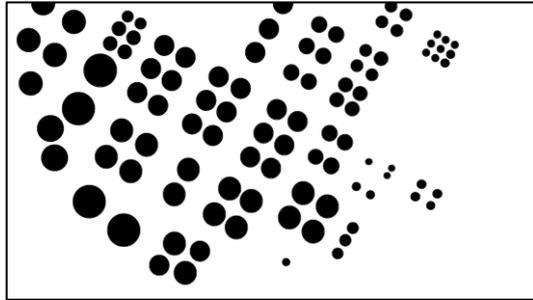


Figure C. 5

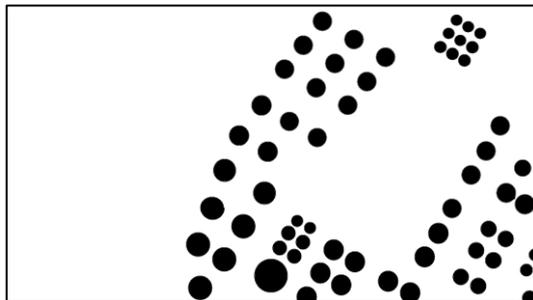


Figure C. 6

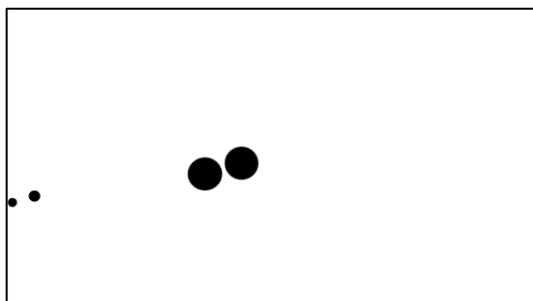


Figure C. 7

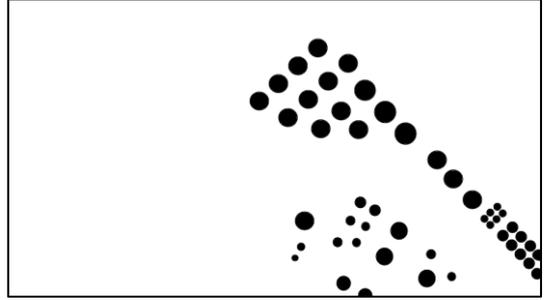


Figure C. 8

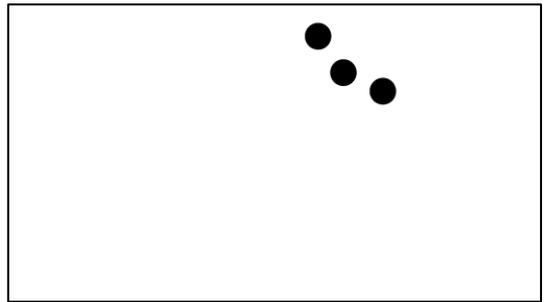


Figure C. 9

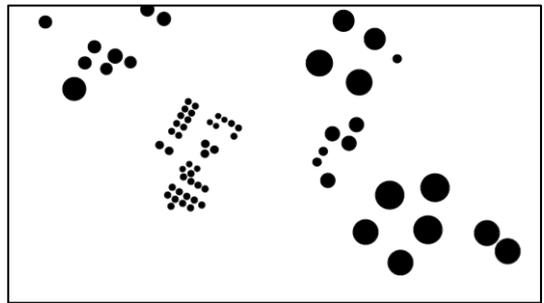


Figure C. 10

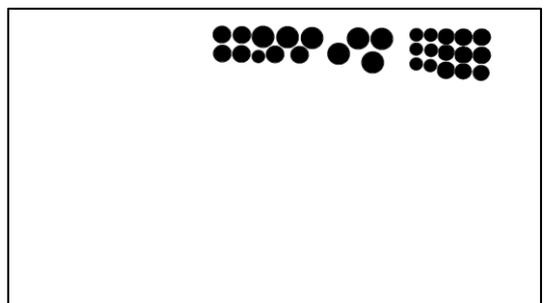


Figure C. 11

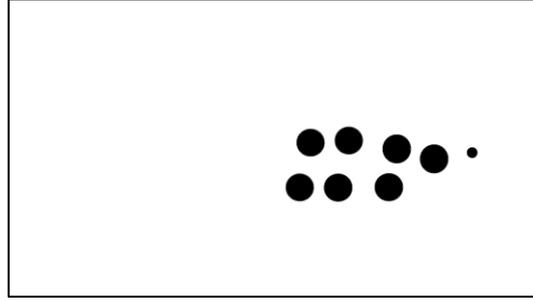


Figure C. 12

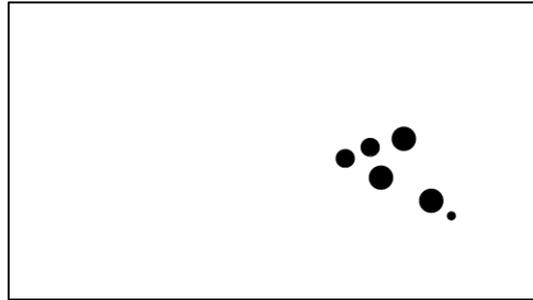


Figure C. 13

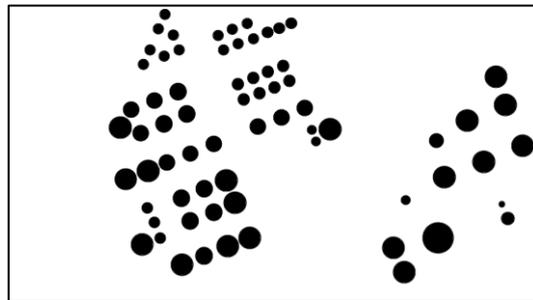


Figure C. 14

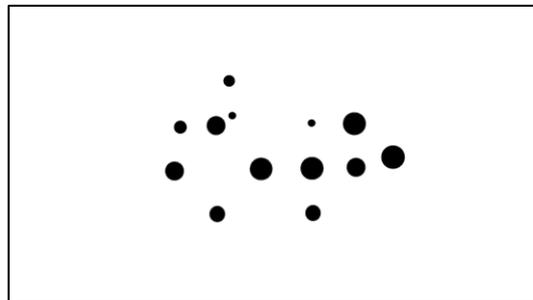


Figure C. 15

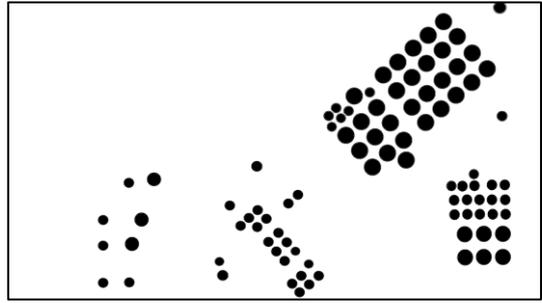


Figure C. 16

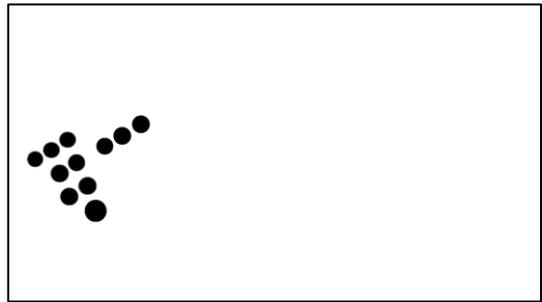


Figure C. 17

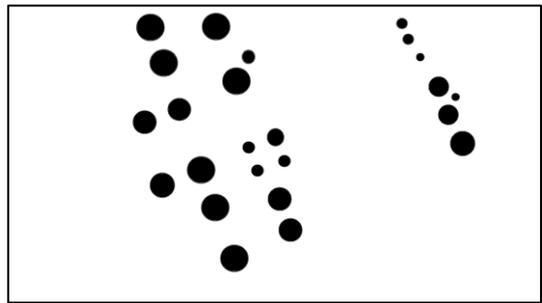


Figure C. 18

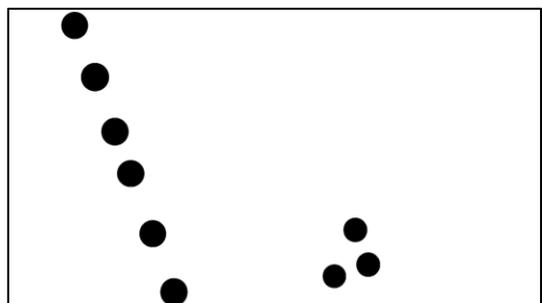


Figure C. 19

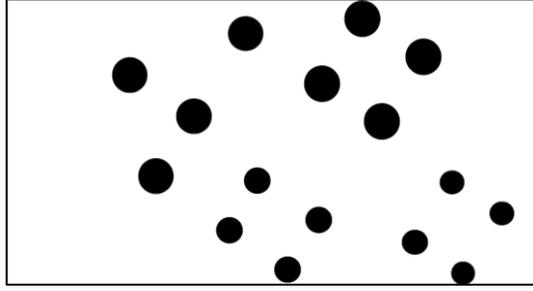


Figure C. 20

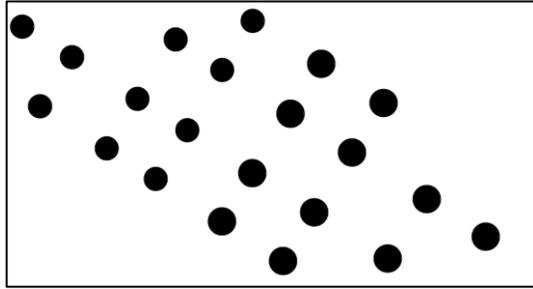


Figure C. 21

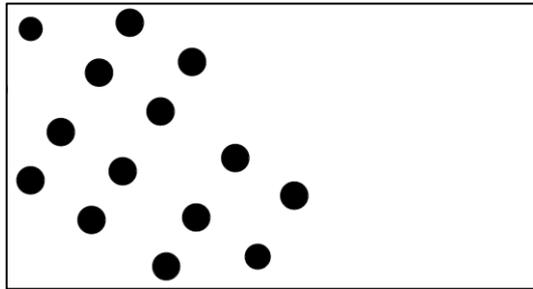


Figure C. 22

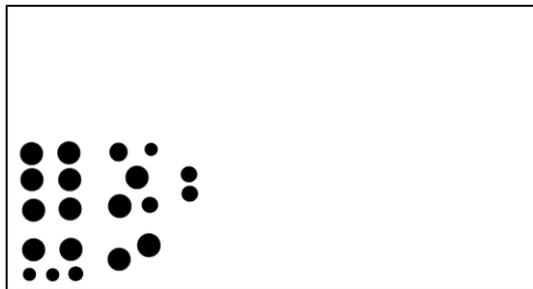


Figure C. 23

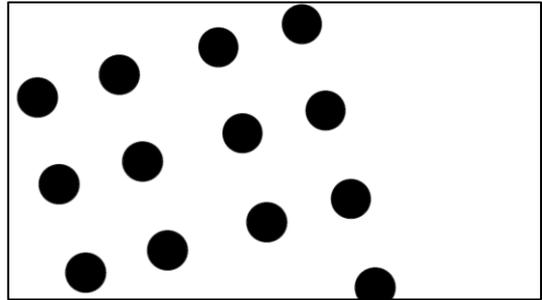


Figure C. 24

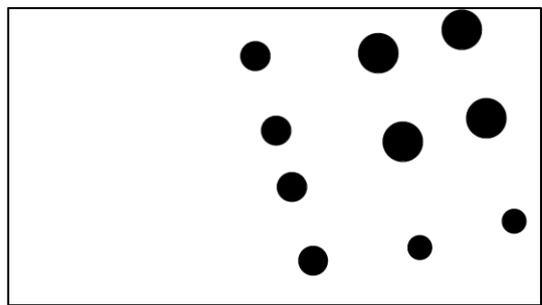


Figure C. 25

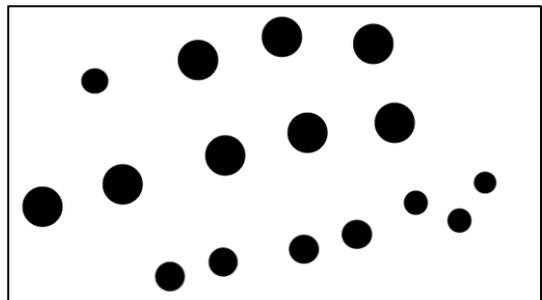


Figure C. 26

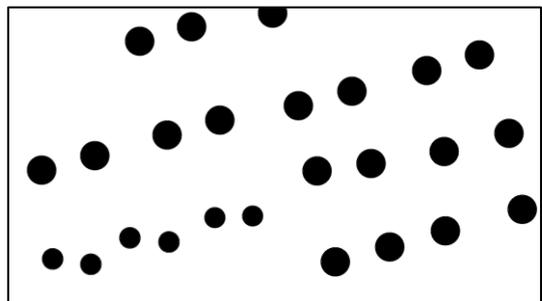


Figure C. 27

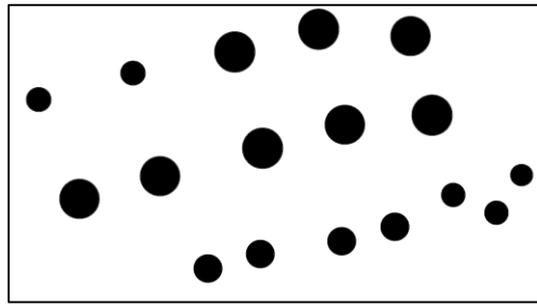


Figure C. 28

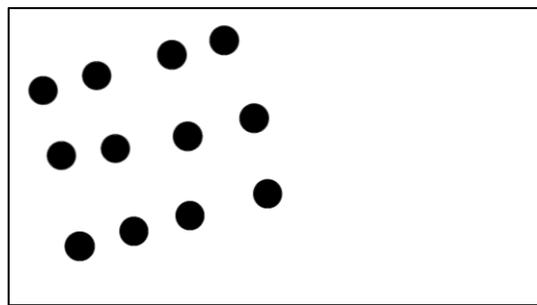


Figure C. 29

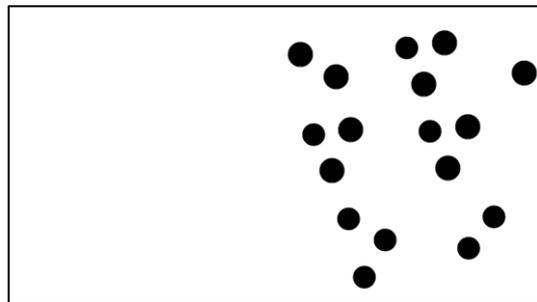


Figure C. 30

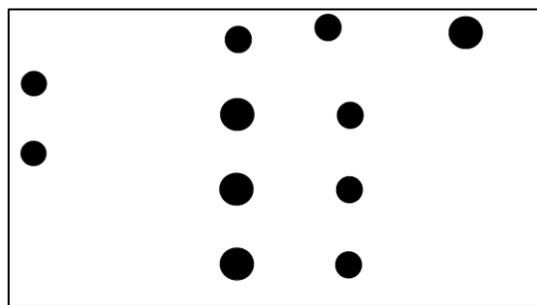


Figure C. 31

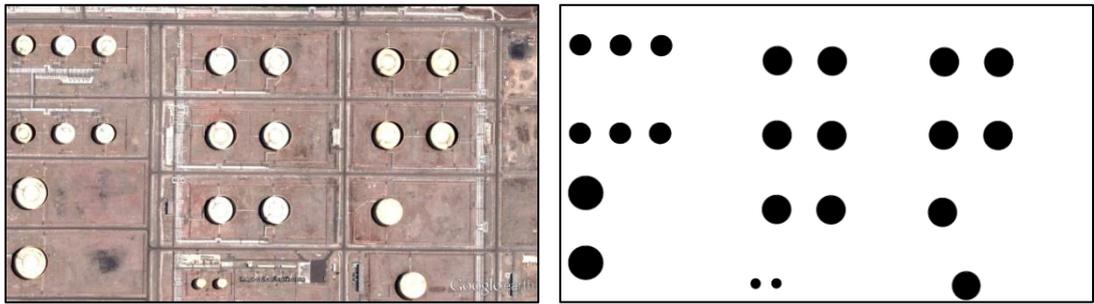


Figure C. 32

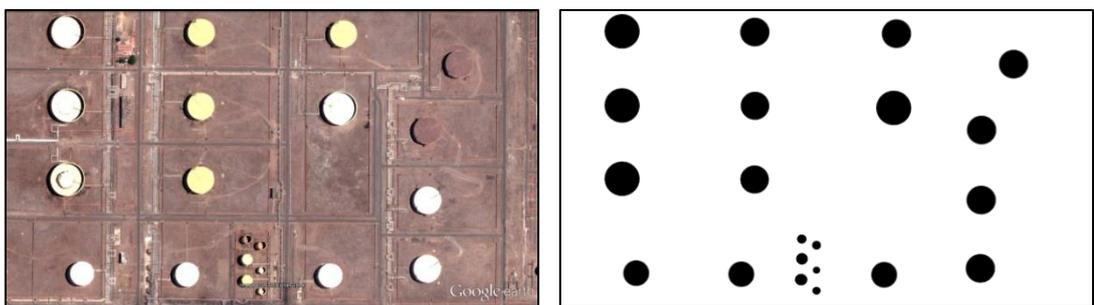


Figure C. 33



Figure C. 34

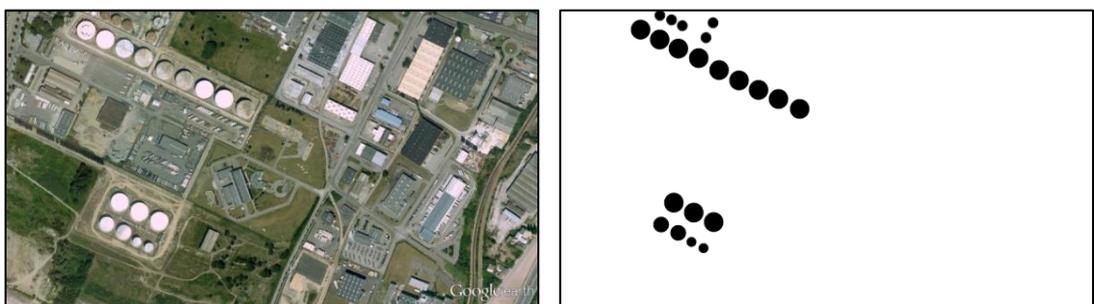


Figure C. 35

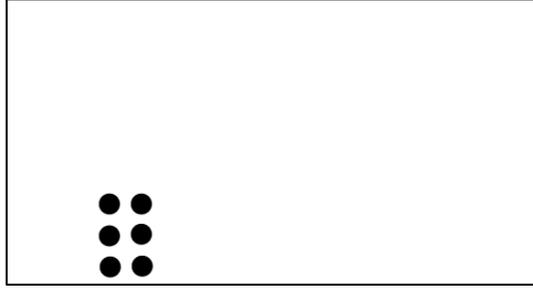


Figure C. 36

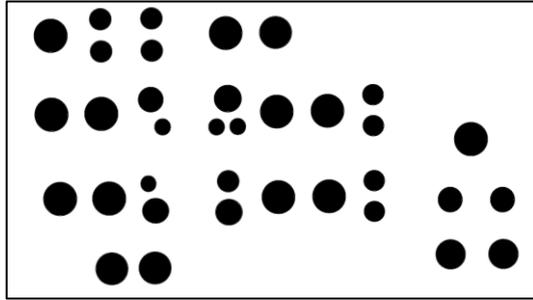


Figure C. 37

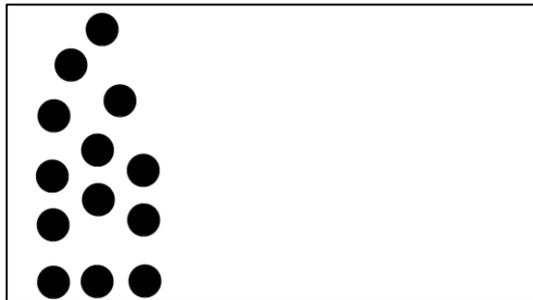


Figure C. 38

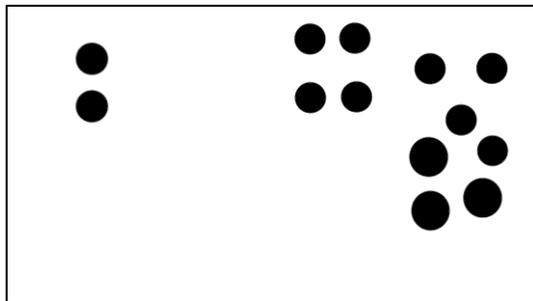


Figure C. 39

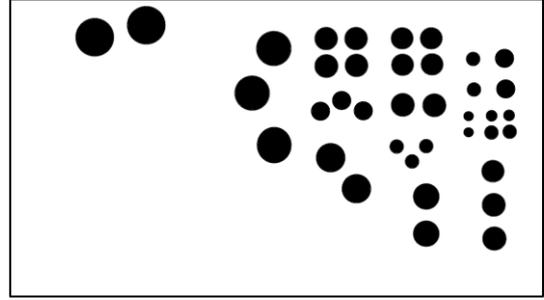


Figure C. 40

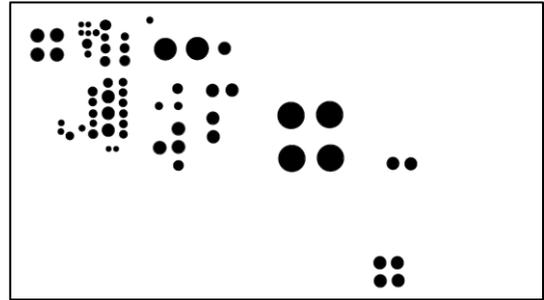


Figure C. 41

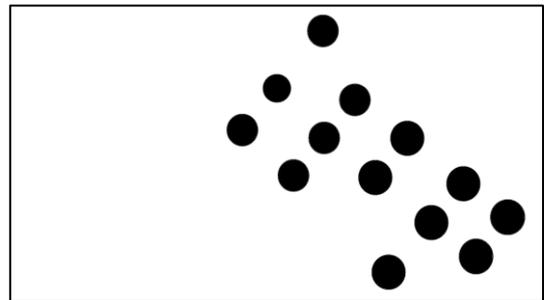


Figure C. 42

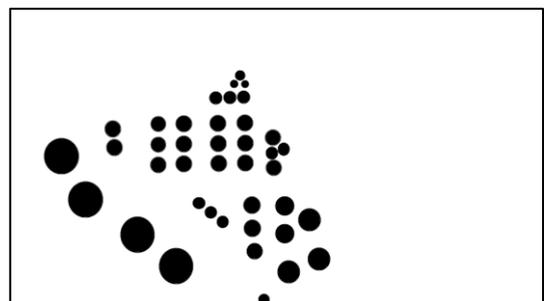


Figure C. 43

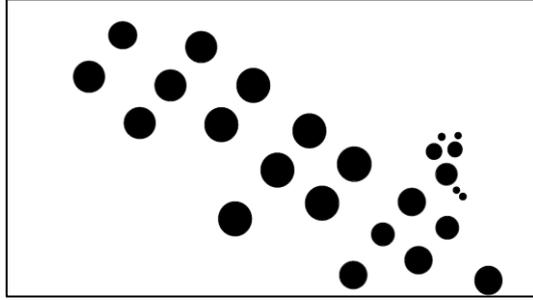


Figure C. 44

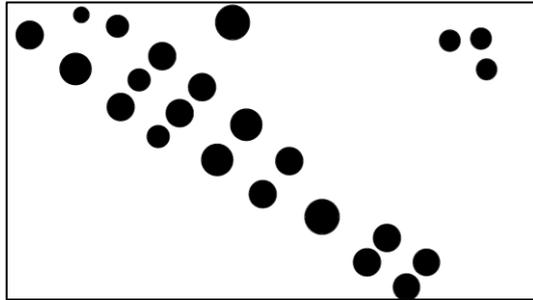


Figure C. 45

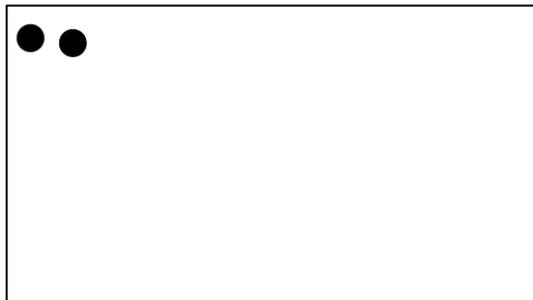


Figure C. 46

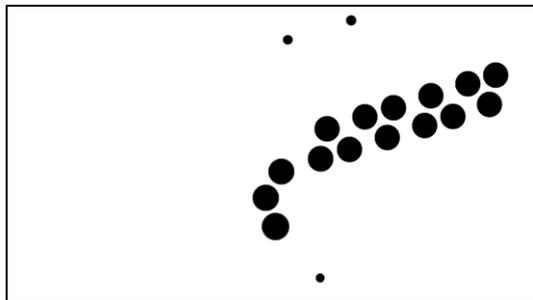


Figure C. 47

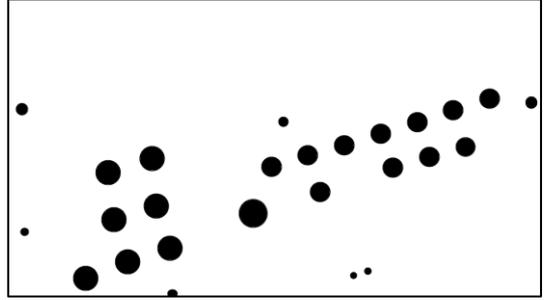


Figure C. 48

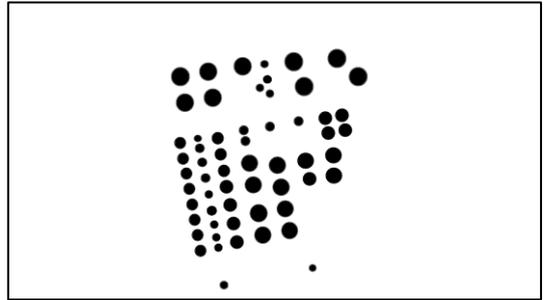


Figure C. 49

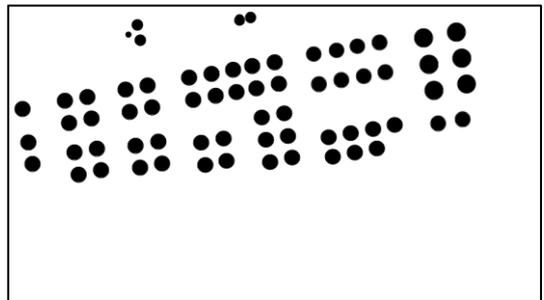


Figure C. 50

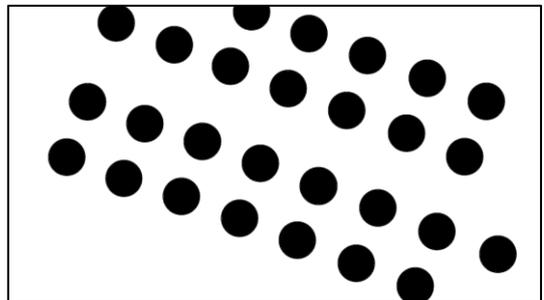


Figure C. 51

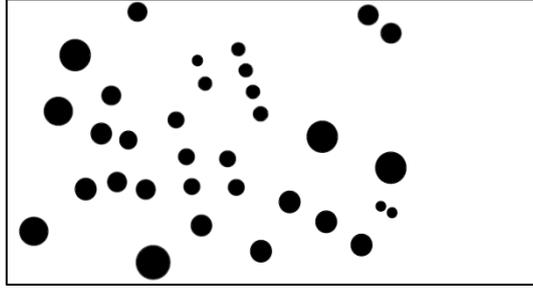


Figure C. 52

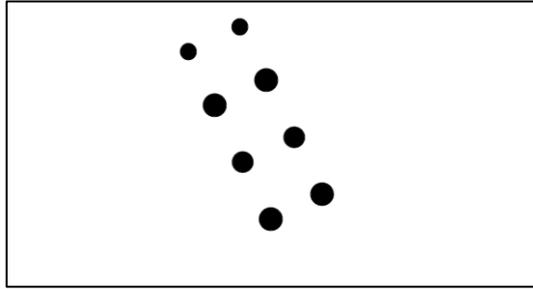


Figure C. 53

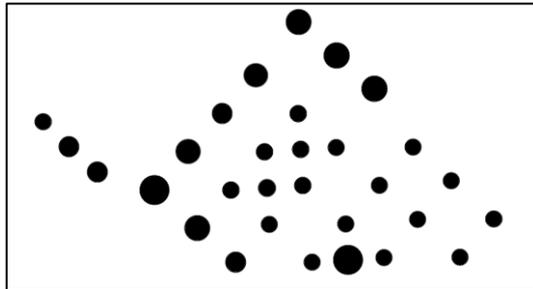


Figure C. 54

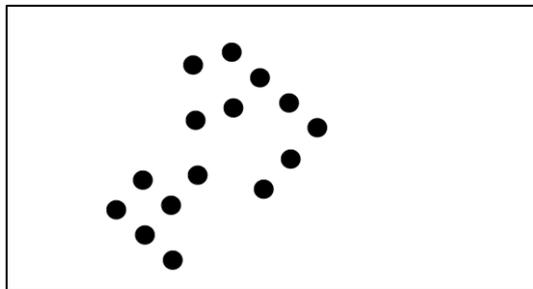


Figure C. 55

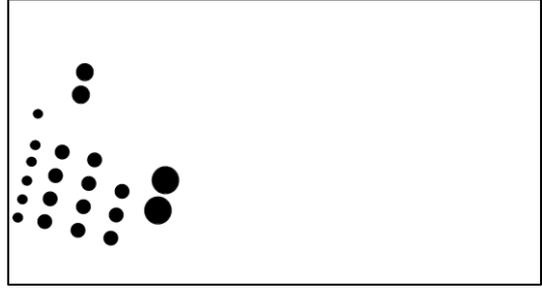


Figure C. 56

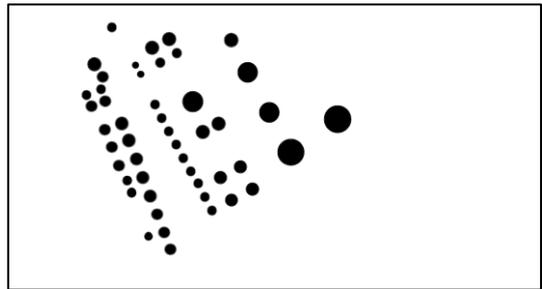


Figure C. 57

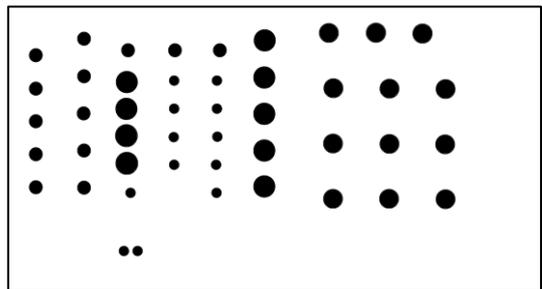


Figure C. 58

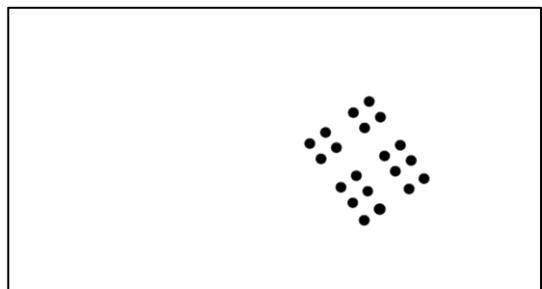


Figure C. 59

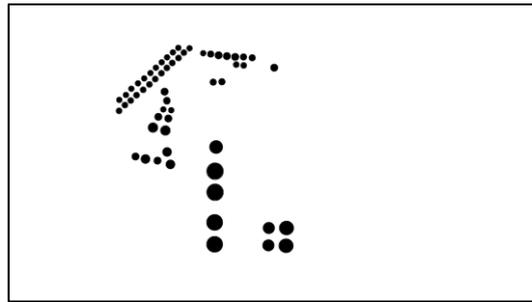


Figure C. 60

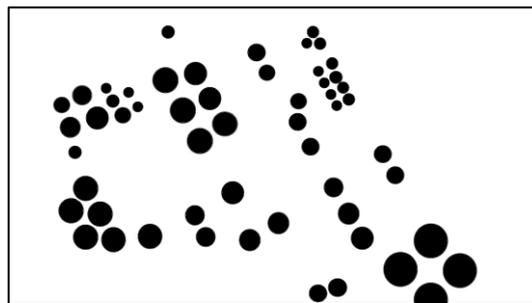


Figure C. 61

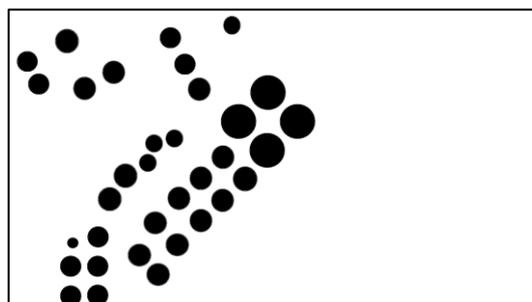


Figure C. 62

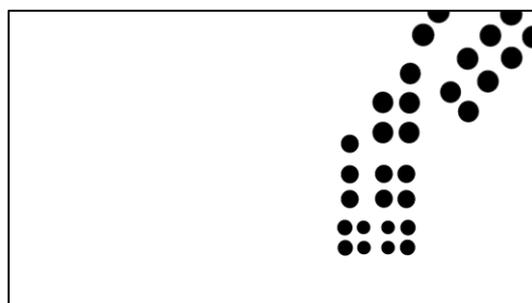


Figure C. 63

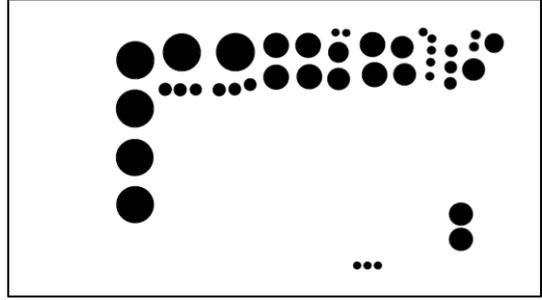


Figure C. 64

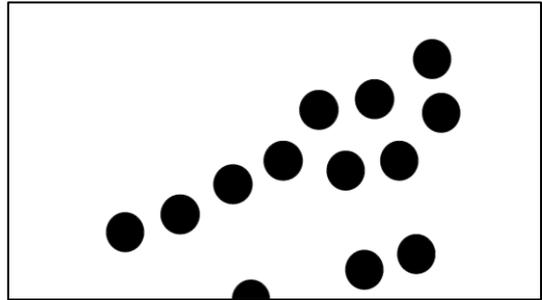


Figure C. 65

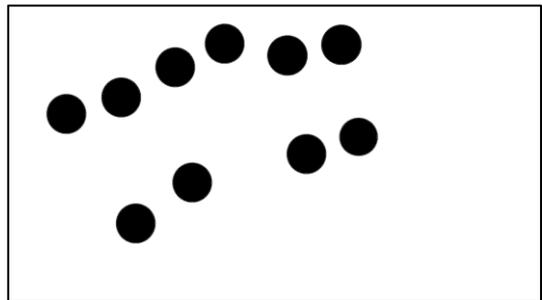


Figure C. 66

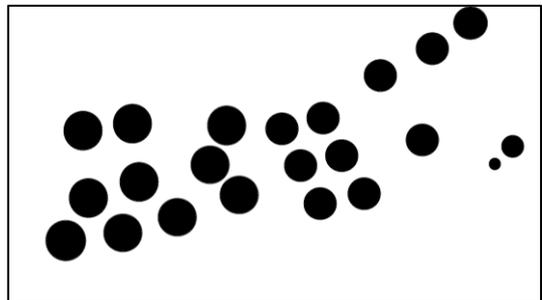


Figure C. 67

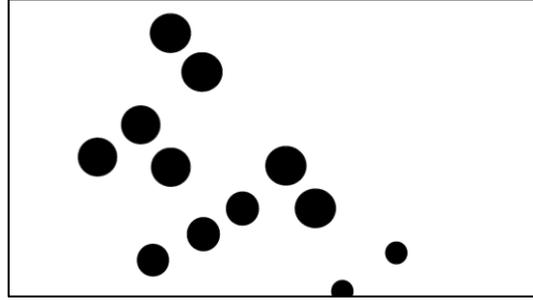


Figure C. 68

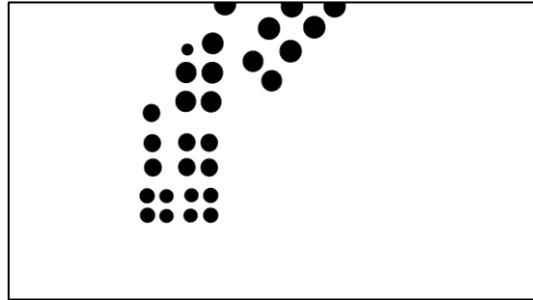


Figure C. 69

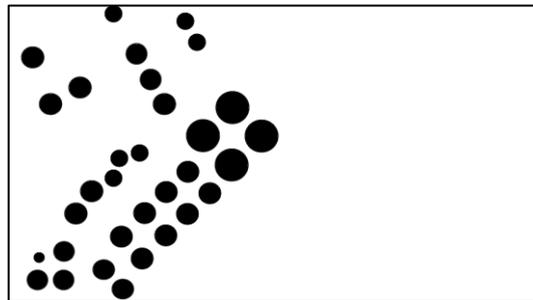


Figure C. 70

VITA

O. Erman OKMAN was born in Ankara, Turkey, in 1982. He received the B.S. and M.S degrees in Electrical and Electronics Engineering, from Middle East Technical University, Ankara, Turkey in 2004 and 2006, respectively. From 2004 to 2006, he was a research assistant at the same department and participated in European Union FP7 funded projects, iClass and 3D-TV. He commenced his doctorate study at Electrical and Electronics Engineering at Middle East Technical University and started to work as a researcher at TÜBİTAK (The Scientific and Technological Research Council of Turkey) UEKAE ILTAREN Research Institute. Since 2010 he has been working as a Senior System Engineer at SDT (Space and Defence Technologies) Inc. His research interests include remote sensing image analysis, image enhancement, data hiding, radar signal processing and statistical signal processing. His publications are as follows:

Refereed Journal Papers:

O. E. Okman, G. Bozdagi Akar, "A fast circle detection method using radon transform", under review in *Computer Vision and Image Understanding*.

A. O. Karalı, O. E. Okman, T. Aytaç, 2011, Adaptive image enhancement based on clustering of wavelet coefficients for infrared sea surveillance systems, *Infrared Physics and Technology*, Vol. 54, Issue 5, pp.382-394, 2011.

A. O. Karalı, O. E. Okman, T. Aytaç, "Adaptive enhancement of sea-surface targets in infrared images based on local frequency cues", *Journal of Optical Society of America A*, Vol. 27, Issue 3, pp. 509-517, February 2010.

O. E. Okman, G. Bozdagi Akar, "Quantization index modulation based image watermarking using digital holography", *Journal of Optical Society of America A*, Vol. 24, Issue 1, pp. 243-252, January 2007.

International Conference Papers:

O. E. Okman, G. Bozdagi Akar, "A Circle Detection Approach Based on Radon Transform", *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* in Vancouver, Canada, May 2013,

E. Akyilmaz, O.E. Okman, F. Nar, M. Cetin "Automatic and semi-automatic extraction of curvilinear features from SAR images", *SPIE Remote Sensing Europe* in Edinburgh, Scotland, September 2012.

O. E. Okman, F. Nar, C. Demirkesen, M. Cetin, "Feature SAR preserved despeckling and its parallel implementation with application to railway detection", *9th European Conference on Synthetic Aperture Radar (EUSAR)* in Nuremberg Germany, April 2012.

F. Nar, C. Demirkesen, O. E. Okman, M. Cetin, "Region Based Target detection approach for Synthetic Aperture Radar images and its parallel implementation", *SPIE Security & Defense Conference* in Baltimore USA, April 2012.

O. E. Okman, G. Bozdagi Akar, "Quantization index modulation based watermarking using holography", *Optics and Photonics*, in San Diego, California, USA, August 2006.

National Conference Papers:

- E. Akyılmaz , C. Demirkesen, O. E. Okman, F. Nar, M. Cetin, “SAR Görüntülerinde Etkileşimli Gemi Bölütleme”, 21. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Girne, KKTC, April 2013.
- C. Demirkesen, O. E. Okman, F. Nar, M. Cetin, “Sentetik Açıklıklı Radar Görüntülerinde Alan Tabanlı Hedef Tespiti ve Paralel Gerçekleştirilmesi”, Savunma Teknolojileri Kongresi (SAVTEK), Ankara, Turkey, June 2012,.
- N. Kılıçoğlu, S. Şahin, O. E. Okman, Ö. E. İnan, “Sinyal Ayırıştırma Algoritması”, Savunma Teknolojileri Kongresi (SAVTEK), Ankara, Turkey, June 2012.
- A. O. Karalı, S. Çakır, O. E. Okman, T. Aytaç. “İstatistiksel yeğlilik tümleştirme ve uyarlamalı bulanık maskeleyme kullanarak imge tümleştirme”, 20. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Muğla, Turkey, April 2012.
- A. O. Karalı, O. E. Okman, T. Aytaç. “Kızılötesi imgelerde suüstü platformlarının dalgacık katsayılarının toplanmasına dayalı uyarlamalı pekiştirilmesi,” 19. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Antalya, Turkey, April 2011.
- A. O. Karalı, O. E. Okman, T. Aytaç. “Deniz yüzeyi hedefleri içeren kızılötesi imgelerin uyarlamalı pekiştirilmesi”, 18. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Diyarbakır, Turkey, April 2010.
- O. E. Okman, G. Bozdagi Akar, “Hologram Kullanarak Sayısal Imge Damgalama”, 14. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı, Antalya, Turkey, April 2006.