ROTATION, SCALE AND TRANSLATION INVARIANT AUTOMATIC TARGET RECOGNITION USING TEMPLATE MATCHING FOR SATELLITE IMAGERY

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

BY

ALP ERTÜRK

IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2010

Approval of the thesis:

ROTATION, SCALE AND TRANSLATION INVARIANT AUTOMATIC TARGET RECOGNITION USING TEMPLATE MATCHING FOR SATELLITE IMAGERY

submitted by **ALP ERTÜRK** in partial fulfillment of the requirements for the degree of **Master of Science 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. İsmet Erkmen ____________________
Head of Department, **Electrical and Electronics Engineering**

Assoc. Prof. Dr. Tolga Çiloğlu ____________________
Supervisor, **Electrical and Electronics Engineering Dept., METU**

Assoc. Prof. Dr. Aydın Alatan ____________________
Co- Supervisor, **Electrical and Electronics Engineering Dept., METU**

Examining Committee Members:

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

Assoc. Prof. Dr. Tolga Çiloğlu ____________________
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. A. Aydın Alatan ____________________
Electrical and Electronics Engineering Dept., METU

Asst. Prof. Dr. Ilkay Ulusoy ____________________
Electrical and Electronics Engineering Dept., METU

Assoc. Prof. Dr. H. Şebnem Düzgün ____________________
Mining Engineering Dept., METU

Date: ____________________

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 : Alp Ertürk

Signature :

ABSTRACT

ROTATION, SCALE AND TRANSLATION INVARIANT AUTOMATIC TARGET RECOGNITION USING TEMPLATE MATCHING FOR SATELLITE IMAGERY

Ertürk, Alp
M.Sc., Deparment of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Tolga Çiloğlu
Co-Supervisor: Assoc. Prof. Dr. A. Aydın Alatan

January 2010, 150 pages

In this thesis, rotation, scale and translation (RST) invariant automatic target recognition (ATR) for satellite imagery is presented. Template matching is used to realize the target recognition. However, unlike most of the studies of template matching in the literature, RST invariance is required in our problem, since most of the time we will have only a small number of templates of each target, while the targets to be recognized in the scenes will have various orientations, scaling and translations. RST invariance is studied in detail and implemented with some of the competing methods in the literature, such as Fourier-Mellin transform and bipectrum combined with log-polar mapping. Phase correlation and normalized cross-correlation are used as similarity metrics. Encountered drawbacks were overcome with additional operations and modifications of the algorithms. ATR using reconstruction of the target image with respect to the template, based on bispectrum, log-polar mapping and phase correlation outperformed the other methods and successful recognition was realized for various target types, especially for targets on relatively simpler backgrounds, i.e. containing little or no other objects.

Keywords: Automatic target recognition, template matching, log-polar mapping, Fourier-Mellin Transform, bispectrum

ÖZ

UYDU GÖRÜNTÜLERİ İÇİN ŞABLON EŞLEME KULLANILARAK DÖNME, ÖLÇEKLEME VE ÖTELEME DEĞİŞMEZLİKLİ OTOMATİK HEDEF TANIMA

Ertürk, Alp
Yüksek Lisans, Elektrik Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Tolga Çiloğlu
Ortak Tez Yöneticisi: Doç. Dr. A. Aydın Alatan

Ocak 2010, 150 sayfa

Bu tezde, uydu görüntüleri için dönme, ölçekleme ve öteleme değişmezlikli bir otomatik hedef tanıma algoritması sunulmuştur. Hedef tanımayı gerçekleştirmek için şablon eşleme kullanmaktadır. Ancak, literatürdeki şablon eşleme ile ilgili çoğu çalışmanın aksine, problemimizde, her hedefin sadece bir veya az sayıda şablonuna sahip olacağımız için ve sahnelerdeki tanınacak hedefler farklı yönelim, ölçek ve ötelemeye sahip olacaklarından dolayı, dönme, ölçekleme ve öteleme değişmezliği gerekmektedir. Değişmezlik detaylı bir biçimde çalışılmış ve gerçekleştirmek için Fourier-Mellin dönüşümü ve log-polar mapping ile birleştirilen bispectrum gibi literatürdeki başarılı yöntemlerin bazıları kullanılmıştır. Benzerlik ölçevleri olarak faz korelasyonu ve normalize edilmiş çapraz-korelasyon kullanılmıştır. Karşılaşılan zayıflıklar ek işlemler ve yöntemlerin uyarlanması ile aşılmıştır. Bispectrum, log-polar mapping ve faz korelasyonuna dayanan, hedef imgenin şablona göre geri oluşturulması ile otomatik hedef tanıma yöntemi diğer yöntemlerden daha iyi sonuç vermiştir ve birçok hedef türü için, özellikle arkaplan göreceli basitken, yani başka objeleri az içeriyor veya hiç içermiyorken, başarılı tanıma gerçekleştirilmiştir.

Anahtar Kelimeler: Otomatik hedef tanıma, şablon eşleme, log-polar mapping, Fourier-Mellin dönüşümü, bispectrum

To My Family and My Other Half

ACKNOWLEDGMENTS

This thesis was conducted in the Signal Processing Research Laboratory in Electrical and Electronics Engineering Department of Middle East Technical University.

I am thankful for my advisors and supervisors Assoc. Prof. Dr. A.Aydın Alatan and Assoc. Prof. Dr. Tolga Çiloğlu, for their assistance, guidance and insight throughout this study.

My greatest thanks go to my family and Burcu. I am grateful for your unending support and encouragement; without you, none of this would have been possible.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

DEDICATION viii

ACKNOWLEDGMENTS... ix

TABLE OF CONTENTS x

LIST OFTABLES xvi

LIST OF FIGURES xvii

LIST OF ABBREVIATIONS xxv

CHAPTERS

1. INTRODUCTION.. 1

1.1 Motivation... 1

1.2 Scope of the Thesis 2

1.3 Outline of the Thesis 2

2. TEMPLATE MATCHING 3

2.1 Introduction 3

2.2 Literature Survey 3

2.3 Similarity Metrics 5

2.3.1 Sum of Absolute Differences 6

2.3.2 Sum of Squared Differences 6

2.3.3 Cross-Correlation 6

2.3.4 Normalized Cross-Correlation 7

2.3.5 Phase Correlation 9

3. ROTATION, SCALE AND TRANSLATION INVARIANCE 15

3.1 Introduction 15

3.2 Literature Survey …………………………………….…………………..... 16

3.3 Moment-Based Approaches ……………………………..……….......... 17

3.3.1 Hu Moments …………………………..…………...…..…. 18

3.3.2 Zernike Moments ………….....………….…………...…..….. 19

3.3.2.1 Introduction to Zernike Moments …….………...…. 19

3.3.2.2 Zernike Moments ……………..............................…. 19

3.3.2.3 Rotation Invariance of Zernike Moments …..……... 20

3.3.2.4 Scale and Translation Invariance for Zernike Moments ……….. 20

3.3.3 Angular Radial Transform …………..…..………...…..…... 23

3.3.4 Why Not Use Moments? ……………………...…..…....... 24

3.4 Fourier Transform and Its Applications in RST Invariant Algorithms …………………………….……...………..………... 24

3.5 Log-Polar Mapping ……………………………………..…………….. 27

3.6 Fourier-Mellin Transform …………………….………...…………... 29

3.7 Bispectrum ……………………….……………………….…..…….. 32

4. THE METHODS ………………………………………….................... 35

4.1 Introduction ……………………………………………………………. 35

4.2 ATR Using Template Matching in Fourier-Mellin Domain ……..….… 35

4.2.1 Step-by-step Description of the Method ………………..……. 35

4.2.2 Drawbacks of the Method …………………………………… 36

4.2.2.1 Loss of Phase Information ……………………..……… 36

4.2.2.2 Very Small Dynamic Range ……………………..…. 37

4.3 ATR Using Template Matching Based on Reconstruction by Fourier Mellin Transform and Phase Correlation ……………………..……….... 37

4.3.1 Step-by-step Description of the Method ……….…….…... 37

4.3.2 Drawbacks of the Method ……………………………….... 39

4.3.2.1 Multiple Peaks in Phase Correlation …...…..……..... 39

4.3.2.2 Loss of Phase Information ……………………..……… 40

4.4 ATR Using Template Matching Based on Reconstruction by Bispectrum, Log-Polar Mapping and Phase Correlation ……………………………….... 41

4.4.1 Step-by-step Description of the Method …………………..…. 41

4.4.2 Drawbacks of the Method 42

4.4.2.1 Multiple Peaks in Phase Correlation 42

4.4.3 Slice Matters 42

4.5 Common Points of Consideration 43

4.5.1 Overall Drawbacks of the Approach 43

4.5.1.1 Background 43

4.5.1.1.1 Windowing 44

4.5.1.2 Illumination 44

4.5.2 Important Concepts to Pay Attention To 44

4.5.2.1 Centering the DC Component in FT 44

4.5.2.2 The Discord between Circular and Square Operations 45

4.5.2.3 Interpolation in LPM 46

4.6 Exhaustive Approach 46

4.7 Eigen Approach 47

5. EXPERIMENTAL RESULTS 49

5.1 Introduction to Experiments 49

5.2 Test Data 50

5.3 Experiments on Variations and Parameters 50

5.3.1 Overlap vs. Non-overlap 50

5.3.1.1 Non-overlapping Blocks 51

5.3.1.2 Overlapping Blocks with ½ Overlap Ratio 53

5.3.1.3 Overlapping Blocks with ¾ Overlap Ratio 55

5.3.1.4 Result of the Experiments on Overlapping 57

5.3.2 Windowing Analysis 57

5.3.2.1 Without Windowing 59

5.3.2.2 Gaussian Windowing 61

5.3.2.3 Hamming Windowing 63

5.3.2.4 Hanning Windowing 65

5.3.2.5 Result of the Experiments on Windowing 67

5.3.3 Using Artificial Background for the Template 67

5.3.3.1 Template with Natural Background 68

5.3.3.2 Template with Artificial Background ……………..... 69

5.3.3.3 Result of the Experiments on using Artificial Background for the Template …………..…………………. 70

5.3.4 Slice Analysis ………………………………………..…. 70

5.3.4.1 Single Slices ……………………………………….. 71

5.3.4.2 Multiple Slices …………………......………………. 81

5.3.4.3 Result of the Experiments on Slice Analysis ....…. 84

5.4 Performance Tests …..………………….…………………..………. 84

5.4.1 Effect of Bispectrum Slice Constant on Recall vs. Precision ……..……………………………………………………..… 84

5.4.2 First Performance Comparison Test ………………...…….. 87

5.4.2.1 Test Data for the First Test ………………...….…. 87

5.4.2.2 Recall vs. Precision Graphs for the First Test …..…. 89

5.4.2.3 Results of the First Performance Comparison Test ... 92

5.4.3 Second Performance Comparison Test ………..………….. 92

5.4.3.1 Test Data for the Second Test……………….....…. 92

5.4.2.2 Recall vs. Precision Graphs for the Second Test 101

5.4.2.3 Results of the Second Performance Comparison Test ………………..……………………..……………… 103

5.4.4 Third Performance Comparison Test …………...……..… 103

5.4.4.1 Test Data for the Third Test ……………………..... 103

5.4.4.2 Recall vs. Precision Graphs for the Third Test …..... 107

5.4.4.3 Results of the Third Performance Comparison Test ……………………………………...…...……………. 109

5.4.5 Performance Test on Using Extra Templates ……..………. 109

5.4.5.1 Test Data for the Performance Test on Using Extra Templates ……………………...……………………. 109

5.4.4.2 Recall vs. Precision Graphs for the Performance Test on Using Extra Templates ……………….…………..… 110

5.4.4.3 Results of the Performance Test on Using Extra Templates ………………………………………………. 112

5.4.6 Performance Test on using Different Numbers of Rotated Templates ………………..…………………………………….... 112

5.4.6.1 Test Data for the Performance Test on using Different Numbers of Rotated Templates ………………….……. 112

5.4.6.2 Recall vs. Precision Graphs for the Performance Test on using Different Numbers of Rotated Templates ……... 112

5.4.6.3 Results for the Performance Test on using Different Numbers of Rotated Templates ………………...……….... 114

5.4.7 Performance Test on Airports …………….……………...… 114

5.4.7.1 Test Data for the Performance Test on Airports …. 114

5.4.7.2 Recall vs. Precision Graphs for the Performance Test on Airports ……..…………………………………………… 118

5.4.7.3 Results for the Performance Test on Airports …..... 120

5.4.8 Performance Test on Helipads …………….………………….. 120

5.4.8.1 Test Data for the Performance Test on Helipads …. 120

5.4.8.2 Recall vs. Precision Graphs for the Performance Test on Helipads …..……………………………………...…. 125

5.4.8.3 Results for the Performance Test on Helipads ...…. 127

5.4.9 Performance Test on Buildings ……………..………………. 127

5.4.9.1 Test Data for the Performance Test on Buildings ………………………………..……..……………….. 127

5.4.9.2 Recall vs. Precision Graphs for the Performance Test on Buildings ……………………………..………….. 128

5.4.9.3 Results for the Performance Test on Buildings …... 130

5.4.10 Performance Test on Cars ……………………...……. 130

5.4.10.1 Test Data for the Performance Test on Cars ……. 130

5.4.10.2 Recall vs. Precision Graphs for the Performance Test on Cars ………………………..……………………….. 131

5.4.10.3 Results for the Performance Test on Cars ……..... 133

5.4.11 Overall Analysis of the Performance Tests ……………….. 133

5.5 Computation Time Analysis ………………………………………. 136

5.6 Template Selection and Extraction ……………………..…..……. 136

6. CONCLUSIONS …………………………………………………….………..… 139

6.1 Summary ………………………………………………………………..…. 139

6.2 Conclusions ……...…………………………………………………… 140

6.3 Future Studies ……………………………………………………..….. 142

REFERENCES ……………………………………………….…….……… 144

LIST OF TABLES

TABLES

Table 5.1: Precision rates of the methods at 100% recall rate ………………..….. 133

Table 5.2: Recall rates of the methods at 100% precision rate ………………..… 134

Table 5.3: The effect of using extra templates on recall rates at 100% precision rate ……………………………………………………………………………..… 135

Table 5.4: The effect of using extra templates on precision rates at 100% recall rate …………………………………………………………………………….. 135

Table 5.5: Computation times of the methods ……………………...…….……... 136

LIST OF FIGURES

FIGURES

Figure 2.1: Template image for NCC ………………..…………………………..……... 7

Figure 2.2: Test image for NCC, the matched template location is marked …......….. 8

Figure 2.3: Normalized cross-correlation result for the images in Figures 2.1 and 2.2 …………..…………………………………………………………………………..…... 8

Figure 2.4: Lena image and the circularly translated Lena image ……………...…. 10

Figure 2.5: Phase correlation result for Figure 2.4 …………………….....………..... 10

Figure 2.6: Template image for PC ...………………………..…………………..…..... 11

Figure 2.7: Test image for PC, the matched template location is marked ………... 11

Figure 2.8: Phase correlation result for Figures 2.6 and 2.7 ……………………..… 12

Figure 2.9: Lena image and the Lena image with increased average value and fix gain …………………………………………………………………….……….... 13

Figure 2.10: Phase correlation result for Figure 2.9 …...………………………….. 13

Figure 3.1: Some of the orthogonal Zernike moment basis functions …………….. 22

Figure 3.2: Real parts of ART basis functions (n=3, m=12) …………………...…. 23

Figure 3.3: An image and its circularly translated version ……………………….. 25

Figure 3.4: Fourier transform magnitudes of the images in Figure 3.3 …....…...….. 25

Figure 3.5: An image and its scaled version …………………………………...….. 25

Figure 3.6: Fourier transform magnitudes of the images in Figure 3.5 …....…...….. 26

Figure 3.7: An image and its rotated version ………………………………….…... 26

Figure 3.8: Fourier transform magnitudes of the images in Figure 3.7 ………….... 26

Figure 3.9: Log-Polar Mapping in Coordinates ……………………………….…... 27

Figure 3.10: An image and its log-polar mapping ……………………………….... 28

Figure 3.11: Scaling property of log-polar mapping ……………………….……... 29

Figure 3.12: Rotation property of log-polar mapping 29
Figure 3.13: Fourier-Mellin transform process 29
Figure 4.1: Flowchart of ATR Using Template Matching in Fourier-Mellin Domain 36
Figure 4.2: Flowchart of ATR using template matching based on reconstruction by Fourier-Mellin transform and phase correlation 38
Figure 4.3: Flowchart of ATR using template matching based on reconstruction by bispectrum, log-polar mapping and phase correlation 42
Figure 4.4: How the “*fftshift*” works 45
Figure 4.5: Inner and outer circles for LPM 45
Figure 5.1: Template image for the first overlapping test 51
Figure 5.2: Template image for the second overlapping test 51
Figure 5.3: Test image for the first template with the non-overlapping sub-images shown 51
Figure 5.4: Final NCC results for Figure 5.3 52
Figure 5.5: Test image for the second template with the non-overlapping sub-images shown 52
Figure 5.6: Final NCC results for Figure 5.5 53
Figure 5.7: Test image for the first template with the ½ overlapping sub-images shown 53
Figure 5.8: Final NCC results for Figure 5.7 54
Figure 5.9: Test image for the second template with the ½ overlapping sub-images shown 54
Figure 5.10: Final NCC results for Figure 5.9 55
Figure 5.11: Test image for the first template with the ¾ overlapping sub-images shown 55
Figure 5.12: Final NCC results for Figure 5.11 56
Figure 5.13: Test image for the second template with the ¾ overlapping sub-images shown 56
Figure 5.14: Final NCC results for Figure 5.13 57
Figure 5.15: Gaussian window 58
Figure 5.16: Hamming window 58

Figure 5.17: Hanning window ………………………….……………………... 58

Figure 5.18: Template image for the first windowing test ……….……………….... 59

Figure 5.19: Template image for the second windowing test ……………………... 59

Figure 5.20: Results for the first windowing test, without windowing (threshold=0.7) ……………………….………………………………………………………..…. 59

Figure 5.21: Final NCC results for the first windowing test without windowing 60

Figure 5.22: Results for the second windowing test, without windowing (threshold=0.5) ……………………………………………………………….…. 60

Figure 5.23: Final NCC results for the second windowing test without windowing …………………………………………………………………………………... 61

Figure 5.24: Results for the first windowing test, with Gaussian windowing (threshold=0.7) ………………………………………………………………..…. 61

Figure 5.25: Final NCC results for the first windowing test with Gaussian windowing …………………………………………………………………………………... 62

Figure 5.26: Results for the second windowing test, with Gaussian windowing (threshold=0.5) ……………………………………………………………….…. 62

Figure 5.27: Final NCC results for the second windowing test with Gaussian windowing ……………………………………………………………………..…. 63

Figure 5.28: Results for the first windowing test, with Hamming windowing (threshold=0.7) ………………………………………………………………….. 63

Figure 5.29: Final NCC results for the first windowing test with Hamming windowing …………………………………………………………………….... 64

Figure 5.30: Results for the second windowing test, with Hamming windowing (threshold=0.5) ………………………………………………………………….. 64

Figure 5.31: Final NCC results for the second windowing test with Hamming windowing ……………………………………………………………………..…. 65

Figure 5.32: Results for the first windowing test, with Hanning windowing (threshold=0.7) ………………………………………………………………….. 65

Figure 5.33: Final NCC results for the first windowing test with Hanning windowing.…………………………………………………………………..…. 66

Figure 5.34: Results for the second windowing test, with Hanning windowing (threshold=0.5) ……………………………………………………….………………. 66
Figure 5.35: Final NCC results for the second windowing test with Hanning windowing ……………………………………………………………………….…..... 67
Figure 5.36: Template image with natural background ………………………...…. 68
Figure 5.37: Template image with artificial background ……………………....…. 68
Figure 5.38: Results for the template image with natural background (using Hanning windowing) ……………………………………………………………………..... 68
Figure 5.39: Final NCC results for the template image with natural background (using Hanning windowing) ……………………………………………………. 69
Figure 5.40: Results for the template image with artificial background (using Hanning windowing) ……………………………………………………………. 69
Figure 5.41: Final NCC results for the template image with artificial background (using Hanning windowing) ……………………………………………………. 70
Figure 5.42: Template image for the slice analysis test ………………………...…. 71
Figure 5.43: Results using bispectrum slice constant = 1, threshold=0.4 (Size is reduced to fit in page) ……………………..………………………………….…. 71
Figure 5.44: NCC results using bispectrum slice constant = 1 …….……………..... 72
Figure 5.45: Results using bispectrum slice constant = 2, threshold=0.4 (Size is reduced to fit in page) ……………………………………………….………….…. 72
Figure 5.46: NCC results using bispectrum slice constant = 2 ………..………….... 73
Figure 5.47: Results using bispectrum slice constant = 3, threshold=0.4 (Size is reduced to fit in page) ……………………………………………………….…. 73
Figure 5.48: NCC results using bispectrum slice constant = 3 ……………….....…. 74
Figure 5.49: Results using bispectrum slice constant = 4, threshold=0.4 (Size is reduced to fit in page) ……………………………………..…………………….. 74
Figure 5.50: NCC results using bispectrum slice constant = 4 …………………..…. 75
Figure 5.51: Results using bispectrum slice constant = 5, threshold=0.4 (Size is reduced to fit in page) ………………………………….……………………….. 75
Figure 5.52: NCC results using bispectrum slice constant = 5 ……….……...….…. 76
Figure 5.53: Results using bispectrum slice constant = 6, threshold=0.4 (Size is reduced to fit in page) ……………………..……………………………………….. 76

Figure 5.54: NCC results using bispectrum slice constant = 6 …….…………….... 77

Figure 5.55: Results using bispectrum slice constant = 7, threshold=0.4 (Size is reduced to fit in page) …………………….……………………………….…….… 77

Figure 5.56: NCC results using bispectrum slice constant = 7 …………………..... 78

Figure 5.57: Results using bispectrum slice constant = 8, threshold=0.4 (Size is reduced to fit in page) ……………………………….……………………….. 78

Figure 5.58: NCC results using bispectrum slice constant = 8 ……………….….... 79

Figure 5.59: Results using bispectrum slice constant = 9, threshold=0.4 (Size is reduced to fit in page) ……………………..………………………………..… 79

Figure 5.60: NCC results using bispectrum slice constant = 9 ………………….... 80

Figure 5.61: Results using bispectrum slice constant = 10, threshold=0.4 (Size is reduced to fit in page) ……………………………………………………….….. 80

Figure 5.62: NCC results using bispectrum slice constant = 10 ………………..… 81

Figure 5.63: Results using the first five bispectrum slice constants, threshold=0.4 (Size is reduced to fit in page) ………………………………..………..….. 82

Figure 5.64: NCC results using the first five bispectrum slice constants ……......... 82

Figure 5.65: Results using the first ten bispectrum slice constants, threshold=0.4 (Size is reduced to fit in page) ………………………………………...…… 83

Figure 5.66: NCC results using the first ten bispectrum slice constants ………..… 83

Figure 5.67: Template image for the effect of bispectrum slice constant on recall vs. precision ……………………………………………………………………...… 84

Figure 5.68: Test image for the effect of bispectrum slice constant on recall vs. precision ....… 85

Figure 5.69: Recall vs. precision graph to observe the effect of bispectrum slice constant ……………………………………………………………….……..... 86

Figure 5.70: Template image for the first performance comparison test …….…..... 87

Figure 5.71: First test image for the first performance comparison test (size is reduced to fit in page) …………………………………………………..……….. 87

Figure 5.72: Second test image for the first performance comparison test ….….… 88

Figure 5.73: Third test image for the first performance comparison test …….…… 89

Figure 5.74: Results for the first performance comparison test ………………….... 90

Figure 5.75: Performance graphs for different bispectrum slice constants of the last method for the first performance comparison test …………………………………. 91

Figure 5.76: First template image for the second performance comparison test ….. 92

Figure 5.77: Second template image for the second performance comparison test .. 92

Figure 5.78: Third template image for the second performance comparison test … 92

Figure 5.79: First test image for the second performance comparison test (size reduced to fit in page) ………………………………………………………... 93

Figure 5.80: Second test image for the second performance comparison test …..… 93

Figure 5.81: Third test image for the second performance comparison test ……..... 94

Figure 5.82: Fourth test image for the second performance comparison test (size reduced to fit in page) ………………………………………………………... 94

Figure 5.83: Fifth test image for the second performance comparison test (size reduced to fit in page) …………………………………………………….….. 95

Figure 5.84: Sixth test image for the second performance comparison test (size reduced to fit in page) ………………………………………………………... 95

Figure 5.85: Seventh test image for the second performance comparison test ……. 96

Figure 5.86: Eighth test image for the second performance comparison test ……... 96

Figure 5.87: Ninth test image for the second performance comparison test ……… 97

Figure 5.88: Tenth test image for the second performance comparison test …….... 97

Figure 5.89: Eleventh test image for the second performance comparison test …... 98

Figure 5.90: Twelfth test image for the second performance comparison test …..... 99

Figure 5.91: Thirteenth test image for the second performance comparison test ... 100

Figure 5.92: Fourteenth test image for the second performance comparison test .. 101

Figure 5.93: Results for the second performance comparison test ………………. 102

Figure 5.94: Template image for the third performance comparison test ……..…. 103

Figure 5.95: First test image for the third performance comparison test ……….... 104

Figure 5.96: Second test image for the third performance comparison test …....... 104

Figure 5.97: Third test image for the third performance comparison test (size reduced to fit page) ……………………………………………………..………..... 105

Figure 5.98: Fourth test image for the third performance comparison test (size reduced to fit page) ………………………………………………………………. 105

Figure 5.99: Fifth test image for the third performance comparison test (size reduced to fit page) ……………………………………………………….................….. 106

Figure 5.100: Sixth test image for the third performance comparison test (size reduced to fit page) …..……………………………………………………..…..... 106

Figure 5.101: Seventh test image for the third performance comparison test ….... 107

Figure 5.102: Results for the third performance comparison test ……………..…. 108

Figure 5.103: Second template image for the performance comparison test for multiple templates ……………………………………………………………..…. 110

Figure 5.104: Results for the performance test on using extra templates …...…… 111

Figure 5.105: Results for the performance test on rotation ……………………… 113

Figure 5.106: Template image for the performance test on airports ……….…….. 114

Figure 5.107: First test image for the performance test on airports (size is reduced) ……………………………………………………………………..………… 115

Figure 5.108: Second test image for the performance test on airports (size is reduced) ……………………………………………………………………………..… 115

Figure 5.109: Third test image for the performance test on airports (size is reduced) ………………………………………………………………………..……… 116

Figure 5.110: Fourth test image for the performance test on airports (size is reduced) …………………………………………………………………………..…… 116

Figure 5.111: Fifth test image for the performance test on airports (size is reduced) ……………………………………………………………………..…..……… 117

Figure 5.112: Sixth test image for the performance test on airports (size is reduced) ……………………………………………………………………………...…… 117

Figure 5.113: Seventh test image for the performance test on airports (size is reduced) ……………………………………………………………..…………. 118

Figure 5.114: Results for the performance test on airports (size is reduced) ……. 119

Figure 5.115: Template image for the performance test on helipads ……………. 120

Figure 5.116: First test image for the performance test on helipads (size is reduced) …………………………………………………………….……………...….. 121

Figure 5.117: Second test image for the performance test on helipads (size is reduced) ………………………………………………………………………..….. 121

Figure 5.118: Three test image for the performance test on helipads (size is reduced) 122

Figure 5.119: Fourth test image for the performance test on helipads (size is reduced) 122

Figure 5.120: Fifth test image for the performance test on helipads (size is reduced) 123

Figure 5.121: Sixth test image for the performance test on helipads (size is reduced) 123

Figure 5.122: Seventh test image for the performance test on helipads (size is reduced) 124

Figure 5.123: Eighth test image for the performance test on helipads (size is reduced) 124

Figure 5.124: Ninth test image for the performance test on helipads (size is reduced) 125

Figure 5.125: Results for the performance test on helipads 126

Figure 5.126: Template image for the performance test on buildings 127

Figure 5.127: First test image for the performance test on buildings (size is reduced to fit in page) 128

Figure 5.128: Second test image for the performance test on buildings (size is reduced to fit in page) 128

Figure 5.129: Results for the performance test on buildings 129

Figure 5.130: Template image for the performance test on cars 130

Figure 5.131: First test image for the performance test on cars 130

Figure 5.132: Second test image for the performance test on cars 131

Figure 5.133: Third test image for the performance test on cars 131

Figure 5.134: Fourth test image for the performance test on cars 131

Figure 5.135: Results for the performance test on cars 132

LIST OF ABBREVIATIONS

ART	: Angular Radial Transform
ATR	: Automatic Target Recognition
CC	: Cross-Correlation
DFT	: Discrete Fourier Transform
FMT	: Fourier-Mellin Transform
FT	: Fourier Transform
ISTAR	: Intelligence, Surveillance, Target Acquisition, and Reconnaissance
LPM	: Log-Polar Mapping
NCC	: Normalized Cross-Correlation
PC	: Phase Correlation
ROC	: Receiver Operating Characteristics
RST	: Rotation, Scale and Translation
SAD	: Sum of Absolute Differences
SNR	: Signal-to-Noise Ratio
SSD	: Sum of Squared Differences

CHAPTER 1

INTRODUCTION

1.1. Motivation

“Automatic target recognition (ATR) generally refers to the autonomous target detection and recognition by computer processing of data from a variety of sensors such as forward looking infrared (FLIR), synthetic aperture radar (SAR), inverse synthetic aperture radar (ISAR), laser radar (LADAR), millimeter wave (MMW) radar, multispectral/hyperspectral sensors, low-light television (LLTV), video, etc.” [1]

ATR is a crucial element of intelligence, surveillance, target acquisition and reconnaissance (ISTAR). Nowadays, the ability of a country to defend itself can only be reached via technological superiority in the fields of ISTAR, because the detection of possible targets and target locations allows taking necessary precautions in times of peace and enables fast and efficient measures, such as the use of unmanned weapon systems, in times of war. Since ATR reduces the workload of human operators operating in the fields of ISTAR, the importance of ATR has also increased.

Template matching, which is a well-established and easy to realize method used in many areas such as image retrieval, image registration, medical imaging and face recognition, will be used in this study to realize ATR. In this study, the templates are the representative images for each possible target to be detected and recognized. In many studies using template matching, the template will remain exactly, or very much, the same in the test image to be matched. However, in ATR, the targets will be in various orientations and scales, with possible small changes in details or coloring and distortions. In order to detect such a wide range of target types, a very large number of templates would have to be stored and used, which would greatly increase the computation time.

Of special concern among these problems is the variation of rotation, scale and translation. RST invariance will enable us to overcome this heavy burden. With RST invariance, a small number of templates for each target type will be sufficient to detect and recognize all targets invariant of their orientations or scales.

1.2. Scope of the Thesis

In this thesis, our main goal is to achieve rotation, scale and translation invariant automatic target recognition using template matching.

This thesis is concerned only with automatic target recognition from satellite imagery. Although the same techniques can be used for aerial images obtained from unmanned aerial vehicles, no experiments are conducted on such images in this study. ATR using other type of sensors is completely out of the scope of this study.

To realize the RST invariance, more than one method is used and the results are compared via experiments.

1.3. Outline of the Thesis

The outline of the thesis is summarized as follows:

In Chapter 2, template matching is introduced and a literature survey is presented, followed by the descriptions of the similarity metrics that are most commonly used in template matching.

In Chapter 3, RST invariance is studied in detail, with literature survey and a through look into some of the promising methods.

In Chapter 4, the methods that are used in this study are explained in detail with respective drawbacks and the efforts made to overcome these. Also presented in this chapter are the common drawbacks of the approach and important points of consideration.

In Chapter 5, experimental results are presented and discussed.

Finally, in Chapter 6, conclusions and possible future studies will be presented.

CHAPTER 2

TEMPLATE MATCHING

2.1 Introduction

The template matching technique refers to the comparison of a standard representative pictorial pattern (template) with an image with the purpose of finding occurrences of the reference pattern within the image [2]. The basic template-matching algorithm consists of sliding the template over the search area and, at each position, calculating a *distortion*, or *correlation*, measure estimating the degree of dissimilarity, or similarity, between the template and the image [3]. The mostly used similarity –or dissimilarity- measures include sum of absolute differences (SAD), sum of squared differences (SSD), cross-correlation (CC) and normalized cross-correlation (NCC), which is by far the most widely used metric. These similarity metrics will be explained in section 2.3 in more detail.

2.2 Literature Survey

Template matching is one of the most common techniques used in image processing and pattern recognition. Template matching applications include image retrieval [4], image registration [5], image recognition [6], object detection [7], medical imaging [8] and face recognition [9] and verification [10].

A general problem with template matching is the high computation time it requires. Over the years, many techniques have been developed with the intent of reducing this computation time.

The "coarse-to-fine" strategy [7], [11], [12] is a well-known approach to reduce the search area, and, therefore, the computational cost of template matching. Different resolution versions of the template and the image are generated and the low-resolution versions of the template are compared against low-resolution versions of the image, to find the location of the best match. The neighborhood of the best-match location is searched in the image using increasingly higher resolutions, up to the original resolution image. In some studies, like the one proposed by Rosenfeld and VanderBrug [11], matching between higher-resolution template and input images is applied only when there is high similarity in the coarse matching. However, the reduction in computation achieved by “coarse-to-fine” template matching comes at the price of lower precision, i.e., the location of the best match at low-resolution is not necessarily exactly at the location of the best match at full-resolution. Hence, there is a trade-off between reducing computation and precision. The “coarse-to-fine” strategy works well for an object with significantly low spatial frequency components which are retained in a low resolution image, but it does not work well for cluttered scenes and objects whose details need to be checked in order to distinguish one from another [13]. To implement the “coarse-to-fine” strategy, it is also necessary to decide how many levels of resolution are to be used, which depends on the template and the image.

Another strategy to accelerate template matching is to eliminate the positions that cannot provide a better degree of match with respect to the current best-matching one. Gharavi–Alkhansari combined this strategy with the coarse-to-fine strategy, and proposed a method for estimating a threshold in the coarse search and pruning the candidates in the fine search [12].

Yet another strategy to reduce the computational cost is to approximate the template by a function. Schweitzer *et al.* proposed an efficient template matching algorithm using integral images and approximating the input image with polynomials [14]. In a recent study, Omachi and Omachi proposed a method called algebraic template matching, which approximated the template image with a polynomial, and calculates NCC between this approximation and partial images of the input image of various widths and heights [15].

When there are multiple templates to be matched, and the templates have some connections or similarities between each other, a template hierarchy may be build to reduce the computation time. Such a hierarchy was used in [16] for human detection and segmentation. Hierarchical approach can also be combined with “coarse-to-fine” strategy [17].

Another problem with template matching is that is not robust against rotation or other distortions. Although rotation variance may be useful in some cases, such as in [18], in many cases, rotation invariance is strictly preferred, especially in automatic target recognition. The straightforward way to solve this problem, without rotation invariance, is to use a very large number of templates for every possible rotation variance and distortion, but this is computationally very expensive, and in general, unacceptable. There have been some studies in the literature to reduce the large computation time caused by this problem. Uenohara and Kanade have used Karhunen-Loeve transform to represent such a large number of templates with various rotations and distortions with a small number of eigenvectors [13]. Paglieroni *et al* proposed a method to reduce the number of positions and orientations to be searched, using distance transforms and translational cross-correlation [19]. Dufour *et al* proposed a method using estimation on the template location in the input image and minimizing a likelihood surface to find the parameters of rotation and distortion [7].

As can be seen, although there are various studies in the literature to solve the problem of varying rotations and distortions, these are basically to reduce the computation time, instead of actually solving the problem. To solve this problem rotation, scale and translation (RST) invariance was studied in this thesis, so that a very small number of templates can be used for a large possibility of rotation or scale variations of the corresponding object in the input image. RST invariance will be studied in Chapter 3 in detail.

2.3 Similarity (and Dissimilarity) Metrics

There is a large number of similarity and distortion metrics used in template matching, and it is not in the scope of this thesis to cover them all in this section. Brief explanations of the most commonly used metrics follows, with special emphasis on normalized cross-correlation and phase correlation.

2.3.1 Sum of Absolute Differences (SAD):

Sum of absolute differences is a distortion, i.e. dissimilarity, metric. In SAD, the absolute value of the difference between each pixel in the original block –the template in our study-, and the corresponding pixel in the block being used for comparison are summed to create this metric. SAD is also called the L^1 norm. Given an image I of size WxH and a template T of size MxN (M<W, N<H), the SAD at position x, y is defined as:

$$SAD(x,y) = \sum_{\mathrm{j=0}}^{\mathrm{N-1}} \sum_{\mathrm{i=0}}^{\mathrm{M-1}} |I(x+i, y+j) - T(i,j)| \qquad (1)$$

2.3.2 Sum of Squared Differences (SSD):

Sum of squared differences is also a distortion, i.e. dissimilarity, metric. In SSD, the squared value of the difference between each pixel in the original block –the template in our study-, and the corresponding pixel in the block being used for comparison are summed to create this metric. SSD is also called the L^2 norm.

Given an image I of size WxH and a template T of size MxN, the SSD at position x, y is defined as:

$$SSD(x,y) = \sum_{\mathrm{j=0}}^{\mathrm{N-1}} \sum_{\mathrm{i=0}}^{\mathrm{M-1}} \left(I(x+i, y+j) - T(i,j)\right)^2 \qquad (2)$$

2.3.3 Cross-correlation (CC):

Cross Correlation is similar to SSD, since it is motivated by the squared Euclidian distance, but can be implemented more efficiently. Starting from the squared Euclidian distance equation:

$$d_{f,t}^2(u,v) = \sum_{x,y} [f(x,y) - t(x-u, y-v)]^2 \qquad (3)$$

$$d_{f,t}^2(x,y) = \sum_{x,y} [f^2(x,y) - 2f(x,y)t(x-u, y-v) + t^2(x-u, y-v)] \quad (4)$$

The third term in the parenthesis is constant, and the first term is approximately constant. Hence, we arrive at the cross-correlation formula, which is:

$$CC(x,y) = \sum_{\mathrm{j}=0}^{\mathrm{N}-1} \sum_{\mathrm{i}=0}^{\mathrm{M}-1} I(x+i, y+j) \cdot T(i,j)| \qquad (5)$$

There are some disadvantages of using cross-correlation:

- If the image energy – the first term in the parenthesis - varies with position, cross-correlation can fail.
- The range of CC(x,y) is dependent on the size of the feature.
- Cross-correlation is not invariant to changes in image amplitude such as those caused by changing lighting conditions across the image sequence.

2.3.4 Normalized Cross Correlation (NCC):

Normalized cross-correlation overcomes the problems mentioned in the previous section of cross-correlation [20]. The equation of the normalized cross-correlation (NCC) is obtained as follows:

$$NCC(x,y) = \frac{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (I(x+i, y+j) - \bar{I}) \cdot (T(i,j) - \bar{T})}{\sqrt{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (I(x+i, y+j) - \bar{I})^2} \cdot \sqrt{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} (T(i,j) - \bar{T})^2}} \qquad (6)$$

The image and the template are normalized to unit length in NCC.

Note that while SSD, SAD and cross-correlation assume constant brightness, NCC is invariant to linear brightness and contrast variations due to normalization.

NCC is by far the most widely used similarity metric in template matching, and it is also used in this study. Given below, in Figures 2.1, Figure 2.2 and Figure 2.3, is a simple example of template matching using normalized cross-correlation:

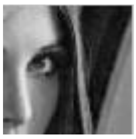


Figure 2.1: Template image for NCC

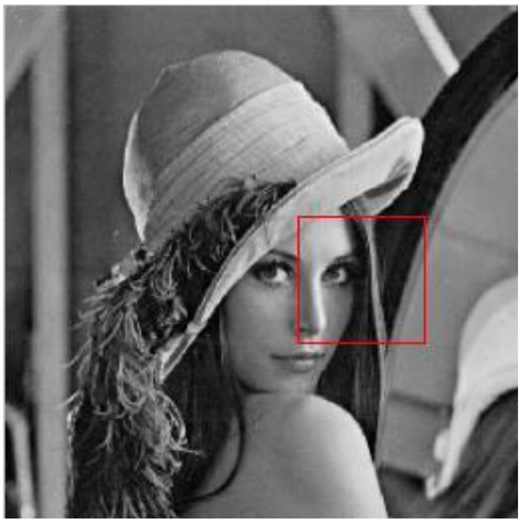


Figure 2.2: Test image for NCC, the matched template location is marked

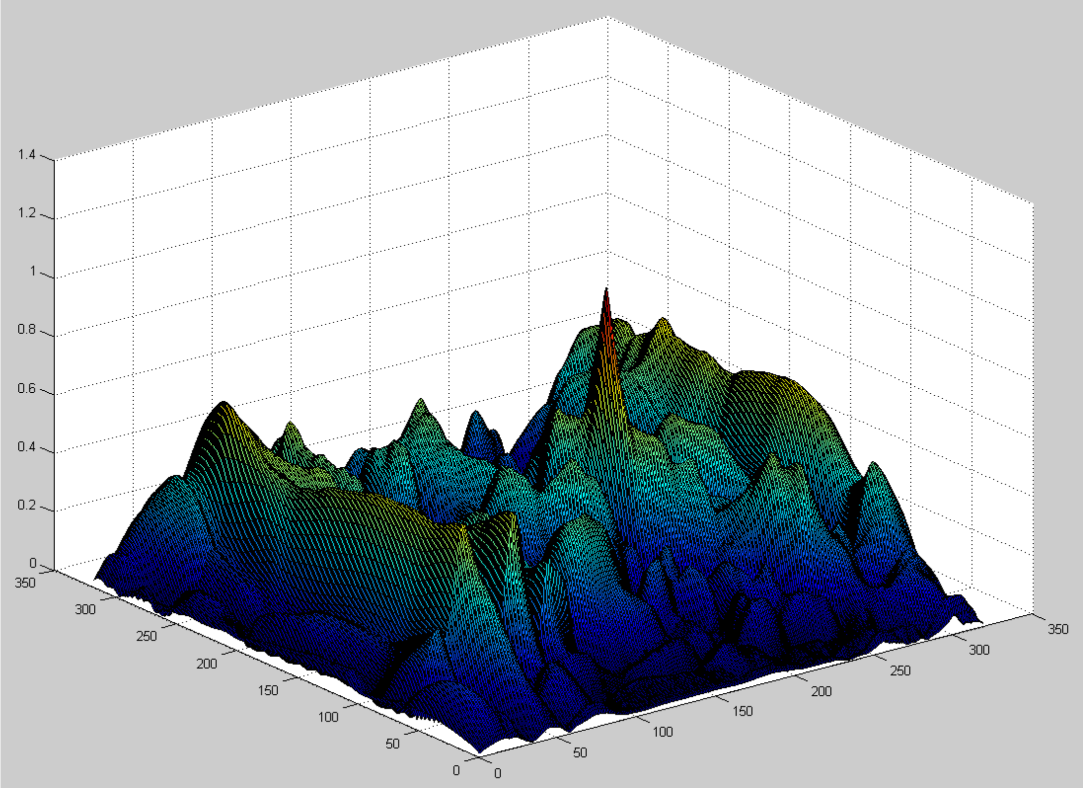


Figure 2.3: Normalized cross-correlation result for the images in Figures 2.1 and 2.2

Since NCC is frequently used in template matching, and a large number of studies of template matching are made on reducing the computation time, it is inevitable that some of these studies try to reduce the computation time of the NCC itself.

In [3], Stefano and Mattoccia applied the principles of successive elimination algorithm and partial distortion elimination, which were proposed for SAD and SSD, to the NCC. Their data-dependent method, called the bounded partial correlation, used a bound for the NCC function to disregard the search positions that are guaranteed not to provide a better match with respect to the current best-match. They have also improved this method in a later study [21]. This approach can also be combined with other methods used to reduce the computation time of template matching, as in [22].

2.3.5 Phase Correlation (PC):

If $x_1[n]$ and $x_2[n]$ represent the two images and $X_1[k]$ and $X_2[k]$ show the corresponding discrete Fourier transforms (DFTs), the phase correlation is defined as:

$$S[n] = F^{-1}\left[\frac{X_1[k]X_2^*[k]}{|X_1[k]X_2^*[k]|}\right] \tag{7}$$

where F^{-1} represents the inverse DFT.

If the two images are the same, the phase correlation result is:

$$S[n] = F^{-1}\left[\frac{X_1[k]X_1^*[k]}{|X_1[k]X_1^*[k]|}\right] = F^{-1}\left[\frac{|X_1[k]|^2}{|X_1[k]|^2}\right] = F^{-1}[1] = \delta[n] \tag{8}$$

Therefore, the phase correlation result of two identical signals has a peak value of unity located at $n = 0$.

If the second image is a spatially shifted version of the first image, i.e.:

$$x_2[n_1, n_2] = x_1[n_1 - t_1, n_2 - t_2] \tag{9}$$

where t_1 and t_2 represents the horizontal and vertical displacements, according to the Fourier shift property, the Fourier transforms will be related as:

$$X_2[u,v] = X_1[u,v]e^{-i(ut_1+vt_2)} \tag{10}$$

Then the phase correlation is obtained as:

$$S[n] = F^{-1}\left[\frac{X_1[k]X_1^*[k]e^{i(ut_1+vt_2)}}{|X_1[k]X_1^*[k]|}\right] = \delta[-t_1,-t_2] \tag{11}$$

Hence, if the second image is a spatially shifted version of the first image, the phase correlation surface is zero everywhere except at location $[-t_1,–t_2]$, in which it is a delta function and represents the displacement between the two images.

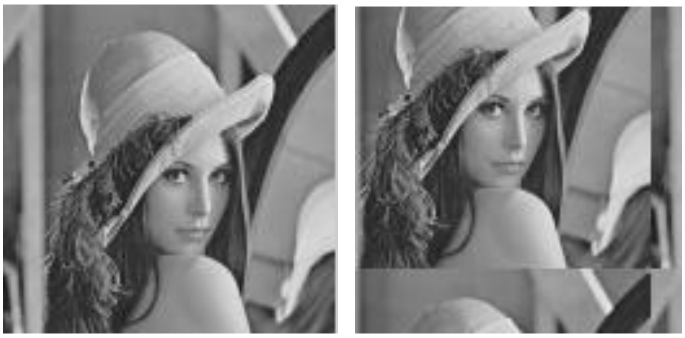


Figure 2.4: Lena image and the circularly translated Lena image

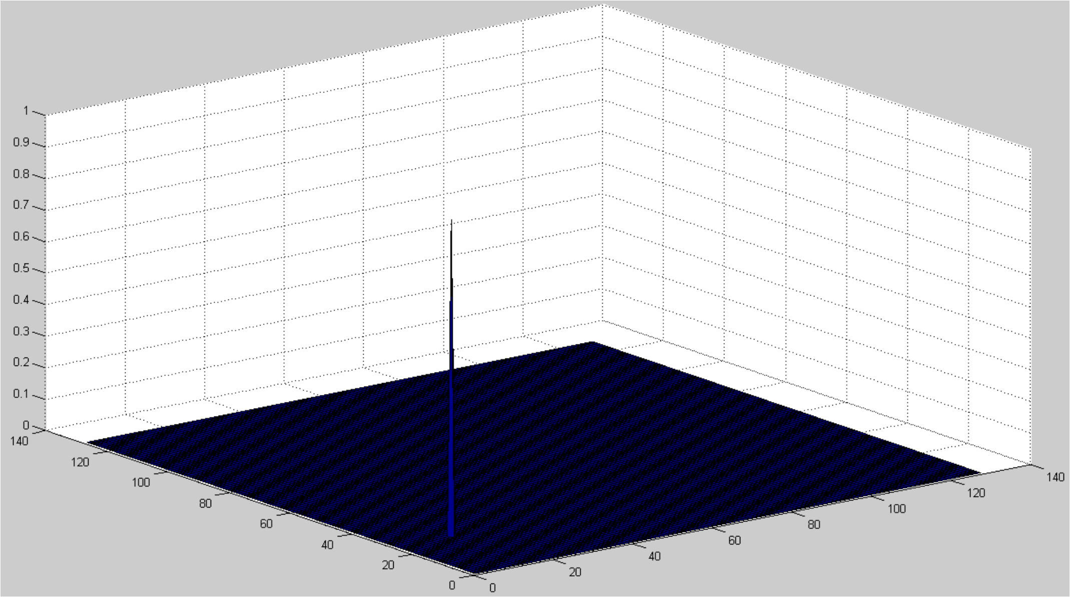


Figure 2.5: Phase correlation result for Figure 2.4

The height of the peak of phase correlation can be used as a good similarity measure for image matching – high peak amplitudes represent good matching, while low peak amplitudes show poor correspondence -, and the location of the peak shows the translational displacement between the two images.

The most remarkable property of phase correlation compared to the classical cross-correlation method is the accuracy by which the peak of the correlation function can be detected: the peak in the phase correlation can be detected much more accurately compared to classical cross correlation because the phase correlation provides a distinct sharp peak at the point of registration whereas the cross-correlation yields several broad peaks and a main peak whose maximum is not always exactly located at the right point [23]. Given below is an example for this property. The example used for template matching using NCC, is repeated here, but is instead matched using phase correlation this time.

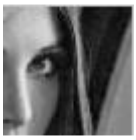


Figure 2.6: Template image for PC

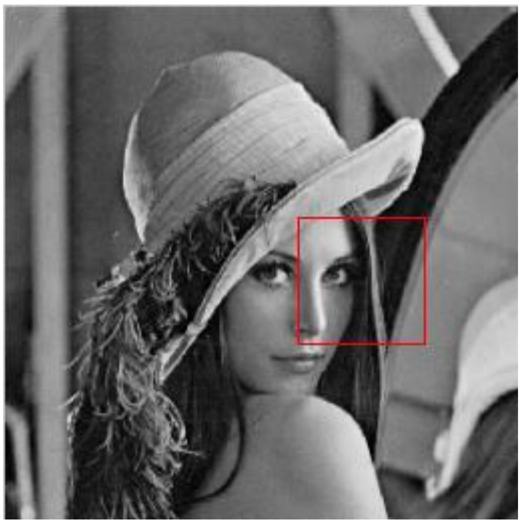


Figure 2.7: Test image for PC, the matched template location is marked

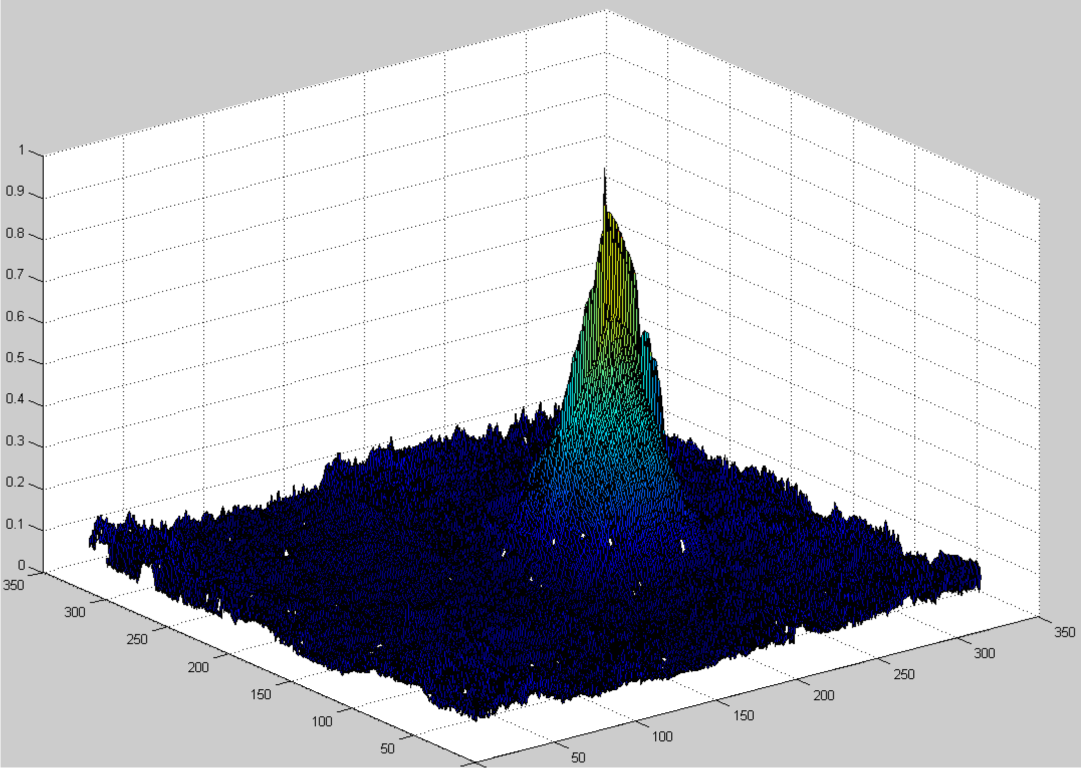


Figure 2.8: Phase correlation result for Figures 2.6 and 2.7

A particularly useful feature of the phase correlation technique is the way performance degrades gracefully as conditions depart from the ideal of pure translation. As conditions depart from ideal translation, the peak in the correlation surface will correspond to the best-fitting translational vector [24].

Another important property is due to the whitening of the signals by normalization, which makes the phase correlation notably robust to those types of noise that are correlated to the signal, e.g., offsets in average value and fixed gain errors [25]. If there is a change in average value and gain so that:

$$x_2[n] = \alpha \cdot x_1[n] + \beta \tag{12}$$

where α and β are constants, then the Fourier transforms will be related by:

$$X_2[k] = \alpha \cdot X_1[k] + \beta \cdot \delta[k] \tag{13}$$

in which case the phase-correlation output is obtained as:

$$S[n] = F^{-1}\left[\frac{X_1[k](\alpha \cdot X_1^*[k] + \beta \cdot \delta[k])}{|X_1[k](\alpha \cdot X_1^*[k] + \beta \cdot \delta[k])|}\right] \tag{14}$$

$$= F^{-1}\left[\frac{\alpha \cdot |X_1[k]|^2 + \beta \cdot \delta[k] \cdot X_1[k]}{|\alpha \cdot |X_1[k]|^2 + \beta \cdot \delta[k] \cdot X_1[k]|}\right] \tag{15}$$

$$= \delta[n] \tag{16}$$

Hence, changes in the average value and gain are automatically canceled by the whitening feature of the phase correlation. Thus, phase correlation is insensitive to changes in image intensity.

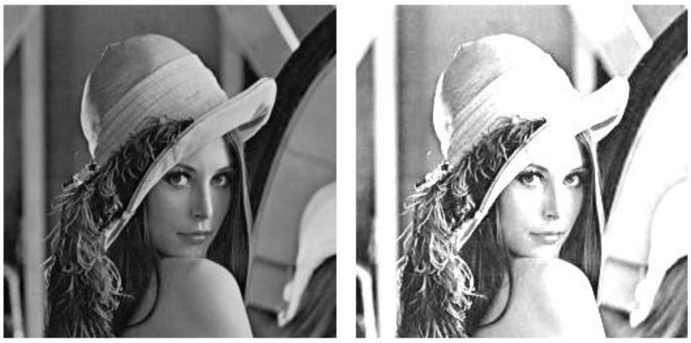


Figure 2.9: Lena image and the Lena image with increased average value and fix gain

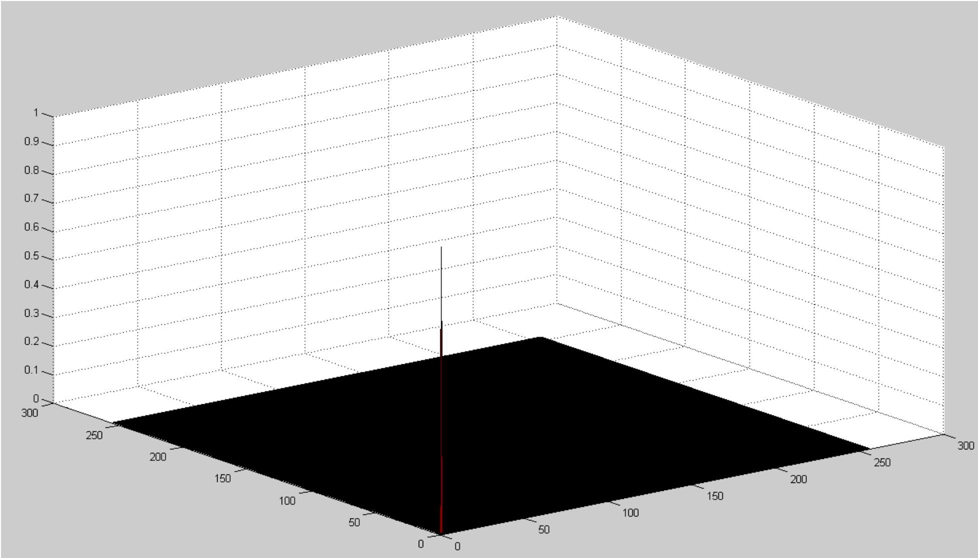


Figure 2.10: Phase correlation result for Figure 2.9

In this study, phase correlation is used as a reliable correlation technique that enables us to detect the translation between the two images. More information on our use of phase correlation will be presented in following chapters, since we will use it - combined with some other techniques that will be explained in the next chapter - to detect the rotation, scale and translation variations between the template and the sub-images.

CHAPTER 3

ROTATION, SCALE AND TRANSLATION INVARIANCE

3.1 Introduction

Rotation, scale and translation (RST) invariance is a technique that is mostly used in watermarking. In order for a watermark to be useful for copyright protection, it must be robust against a variety of possible attacks by pirates. These include robustness against compression such as JPEG compression, geometrical distortion, cropping, row and column removal, addition of noise, filtering, cryptographic and statistical attacks, as well as insertion of other watermarks. While many methods perform well against compression, they lack robustness to geometric distortion [26]. In recent years, watermarking algorithms robust to the geometrical distortions have been the focus of research, because rotation and scaling attacks are considered more challenging than other attacks and even very small geometric distortions can prevent the detection of a watermark [27].

RST invariance is of the utmost importance in this study. Although the problem of the variation of the scale can be easily solved if the altitude that the image is taken is known –since the sizes of the targets to be recognized are already known-, the targets will still be in various orientations. As discussed earlier, using a very large number of templates in every possible orientation for each target is unfeasible in terms of computation time and necessary memory. However, without RST invariance, using only one template in a given orientation is definitely not sufficient to recognize the target in all possible orientations and scales, since correlation results drop drastically with small amount of rotation or scale variation.

Casasent and Psaltis [28] report that the signal-to-noise ratio (SNR) of the correlation peak between two images decreases from 30 dB to 3 dB with either a 2% scale change or a 3.5° rotation. Hence, in this study, we have to solve the RST invariance problem.

3.2 Literature Survey

Invariance against similarity transforms, i.e. RST invariance, has been a focus of research for quite some time.

The first promising results were obtained using moments. Of special concern among these are the Hu moments [29] and the Zernike moments [30]. Angular Radial Transform [31, 32] is moment-based shape descriptor used in MPEG-7, which is similar to Zernike moments, but can be computed faster. The formulations, strengths and weaknesses of these methods will be explained in the next sections in detail. Such features capture global information about the image and do not require closed boundaries as boundary-based methods such as Fourier descriptors do [30]. Moments and functions of moments have been utilized as pattern features in a large number of applications. However they have some weaknesses which prevent them from proving successful results beyond some certain measure [33, 34], these general weaknesses will also be explained later.

Fourier-Mellin transform [28] is a well-performing method for RST invariance. After FMT was used for watermarking by O'Ruanaidh and Pun [35], to provide RST invariance for protection against geometrical attacks, the number of studies on FMT and watermarking have exponentially increased and a very large number of studies on FMT and watermarking can now be found in the literature. Because of the implementation difficulties and image fidelity loss caused by the FMT, some modified algorithms are also commonly used [26]. Most of these algorithms use LPM which is not truly RST invariant. In fact, Fourier–Mellin transform is a log-polar mapping (LPM) following and followed by a Fourier transform, while an inverse Fourier–Mellin transform is an inverse log-polar mapping (ILPM) following and followed by an inverse Fourier transform. Although FMT - and LPM – are mostly used in watermarking [27], [36], [37], they are also used in image registration [38], texture classification [39] and image recognition, including fingerprint [40], iris [41], face [42] and even shoeprint [43] recognition. LPM and FMT will be explained in detail in following sections, since these methods are used to a great extent in this thesis.

Projecting the two dimensional image into one dimension can also yield some properties that can be used to obtain RST invariance. One such technique that is commonly used is the Radon transform, which is a special case of an image projection algorithm. Radon transform represents the image as a collection of projections along various directions. The Radon Transform has been applied to image processing in areas of tomographic reconstruction, image segmentation, determining the orientation of an object, and restoration of images [44]. The Hough transform, which is a special case of the Radon transform, was used in image analysis for edge detection and feature extraction purposes [44]. Radon transform can be used in a variety of ways to obtain RST invariance. Some of the methods that are used with Radon transform for this goal include SVD [44], Fourier transform [45], [46], neural networks [47], LPM [48] and wavelets [49]. Radon transform can also be used by itself for RST invariance after some manipulations [50].

Higher-order spectra, especially the bispectrum, are also used in RST invariance methods. Bispectrum is the Fourier spectrum of the triple correlation of a signal. The higher order spectral features are shown to be as immune to Gaussian noise as features based on moment invariants, but are superior in their immunity to background impulses that cause problems for methods based on accurate calculation of the centroid [51]. Bispectrum behaves similarly to Fourier transform against rotation and scale. However, in contrast to the Fourier transform in which translation invariance is achieved by taking the magnitude and disregarding the phase, Bispectrum is directly translation invariant [52]. This is the basic reason that Bispectrum is used with Radon transform [53] or LPM [54] to obtain RST invariance. Bispectrum is studied in detail in a following section.

3.3 Moment-Based Approaches

Joint moments, of order p and q, of random variables ρ(x,y) are defined in terms of Riemann integrals as:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q \rho(x,y)dxdy \quad (17)$$

The uniqueness theorem states that the double moment sequence $\{m_{pq}\}$ is uniquely determined by ρ(x,y); and conversely, ρ(x,y) is uniquely determined by $\{m_{pq}\}$.

Central moments μ_{pq} are defined as:

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q \rho(x,y)d(x-\bar{x})d(y-\bar{y}) \quad (18)$$

$$\bar{x} = m_{10}/m_{00} \quad \bar{y} = m_{01}/m_{00} \quad (19)$$

The central moments are invariants under translation [29].

To obtain scale invariance, the central moments are normalized:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad \gamma = \frac{(p+q+2)}{2} \quad p+q = 2,3,\ldots \quad (20)$$

3.3.1 Hu Moments

Hu introduced seven nonlinear functions based on normalized central moments which are translation, scale, and rotation invariant [29]. These moment invariants are as follows:

$$\phi_1 = \eta_{20} + \eta_{02} \quad (21)$$

$$\phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \quad (22)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (23)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (24)$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + 3(\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (25)$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (26)$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{30})(\eta_{12} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (27)$$

3.3.2 Zernike Moments

3.3.2.1 Introduction to Zernike Moments

The basis set of regular moments is not orthogonal. Consequently, the recovery of image from these moments is quite difficult and computationally expensive. Moreover, it implies that the information content of the basis have a certain degree of redundancy.

Orthogonal moments including Zernike moments are better than other types of moments in terms of information redundancy and image representation. The orthogonality property enables one to separate out the individual contribution of each order moment to the reconstruction process. The reason for selecting Zernike moments from among the other orthogonal moments is that they possess a useful rotation invariance property. Rotating the image does not change the magnitudes of its Zernike moments. Hence, they could be used as rotation invariant features for image representation.

It should also be noted that Zernike moments are superior to both regular moments and the moment invariants of Hu, in terms of classification accuracy [30].

3.3.2.2 Zernike Moments

Zernike moments are a set of complex polynomials which form a complete orthogonal set over the interior of the unit circle. The form of polynomials is:

$$V_{nm}(x,y) = V_{nm}(\rho,\theta) = R_{nm}(\rho) \cdot exp(jm\theta) \qquad (28)$$

$$R_{nm}(\rho) = \sum_{s=0}^{n-\frac{|m|}{2}} (-1)^s \cdot \frac{(n-s)!}{s! \cdot \left(\frac{n+|m|}{2} - s\right)! \cdot \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s} \qquad (29)$$

Note that the radial polynomials satisfy:

$$R_{n,-m}(\rho) = R_{nm}(\rho) \qquad (30)$$

The Zernike polynomials are orthogonal and satisfy:

$$\int\int_{x^2+y^2\leq 1} [V_{nm}(x,y)]^* V_{pq}(x,y)dxdy = \frac{\pi}{n+1}\delta_{np}\delta_{mq} \tag{31}$$

$$\delta_{ab} = \begin{cases} 1 & a = b \\ 0 & otherwise \end{cases} \tag{32}$$

Zernike moments are obtained by the projection of the image function onto these orthogonal basis functions. The Zernike moment of order n with repetition m for a continuous image function f(x,y) is:

$$A_{nm} = \frac{n+1}{\pi}\int\int_{x^2+y^2\leq 1} f(x,y)V_{nm}^* \, dxdy \tag{33}$$

and in discrete form:

$$A_{nm} = \frac{n+1}{\pi}\sum_{x}\sum_{y} f(x,y)V_{nm}^* (\rho,\theta) \quad , \quad x^2 + y^2 \leq 1 \tag{34}$$

3.3.2.3 Rotation Invariance of Zernike Moments

Consider a rotation of an image by an angle of α, i.e.:

$$f'(\rho,\theta) = f(\rho,\theta - \alpha) \tag{35}$$

After some calculations [27], we arrive at:

$$A'_{nm} = A_{nm}\, exp(-jm\alpha) \tag{36}$$

The equation above shows that Zernike moments merely acquire a phase shift on rotation. Hence, the magnitudes of the Zernike moments of a rotated image remain identical to those before rotation.

3.3.2.4 Scale and Translation Invariance for Zernike Moments

As a disadvantage, the defined features on the Zernike moments are only rotation invariant. To obtain scale and translation invariance, the image is first subjected to a normalization process using its regular moments. The rotation invariant Zernike features are then extracted from the scale and translation normalized image.

Translation invariance is achieved by transforming the original image f (x,y) into another one, so that the origin is moved to the centroid of the original image:

$$f(x+\bar{x}, y+\bar{y}) \tag{37}$$

Scale invariance is obtained by changing the size of the image such that the zeroth order moment m_{00} is equal to a predetermined value β.

Hence for scale and translation invariance, the image f(x,y) should be transformed in to g(x,y), where

$$g(x,y) = f\left(\frac{x}{a}+\bar{x}, \frac{y}{a}+\bar{y}\right) \tag{38}$$

$$a = \sqrt{\beta/m_{00}} \tag{39}$$

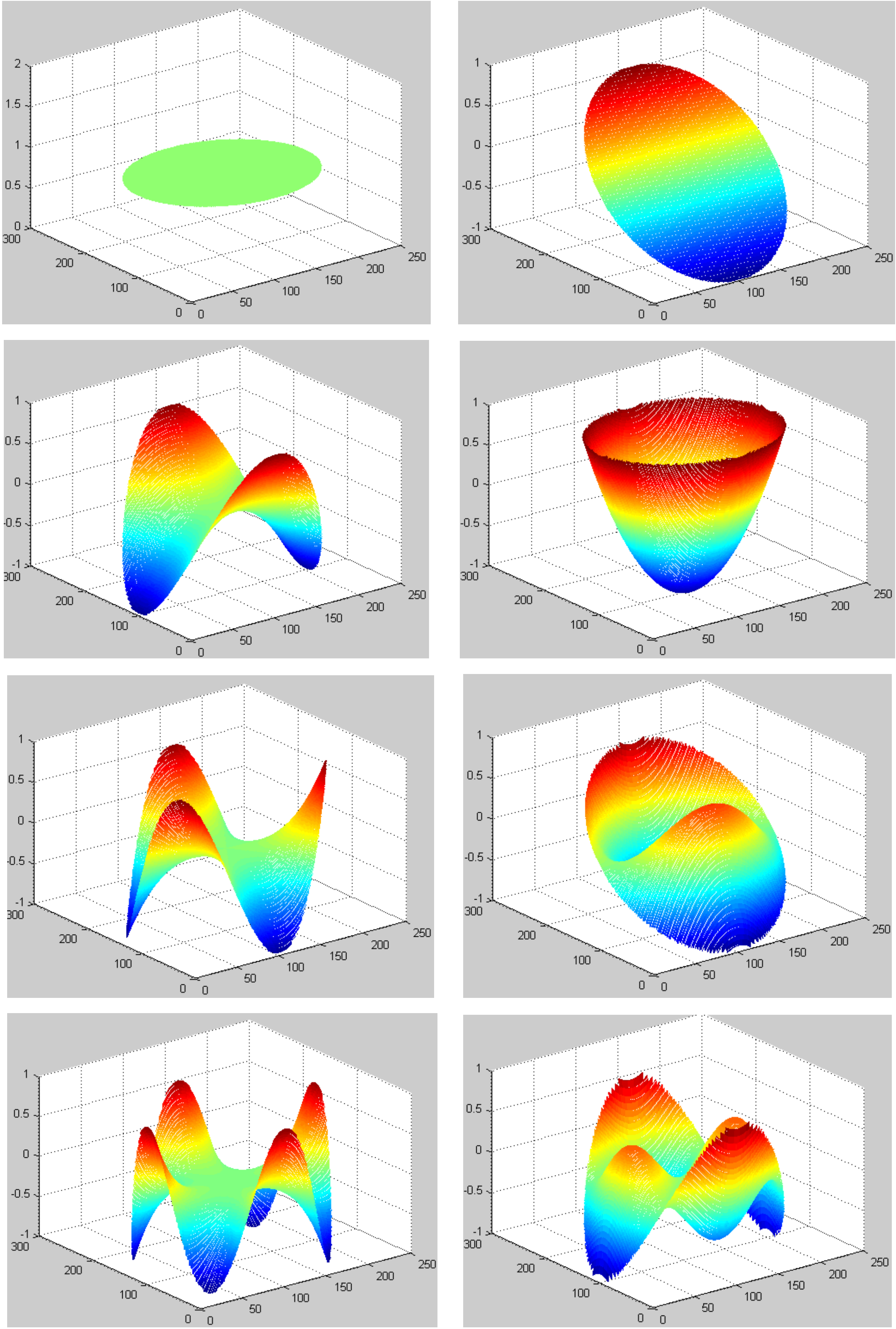


Figure 3.1: Some of the orthogonal Zernike moment basis functions

3.3.2 Angular Radial Transform

Angular Radial Transform is a moment-based method adopted in MPEG-7 as a region based shape descriptor. It is similar to Zernike moments, but can be computed faster. It has compact size, is robust to noise and scaling, invariant to rotation and has the ability to describe complex objects [31].

The ART coefficients, F_{nm} of order *n* and *m*, are defined on a unit disk in polar coordinates as:

$$F_{nm} = \int_0^{2\pi}\int_0^1 V_{nm}(\rho,\theta) f(\rho,\theta) d\rho d\theta \tag{40}$$

where *f(ρ,θ)* is an image function in polar coordinates and $V_{nm}(\rho,\theta)$ is an ART basis function that is separable along the angular and radial directions, i.e.:

$$V_{nm}(\rho,\theta) = A_m(\theta) R_n(\rho) \tag{41}$$

$$A_m(\theta) = \frac{1}{2\pi} exp(jm\theta) \tag{42}$$

$$R_n(\rho) = \begin{Bmatrix} 1 & n = 0 \\ 2\cos(\pi n\rho) & n \neq 0 \end{Bmatrix} \tag{43}$$

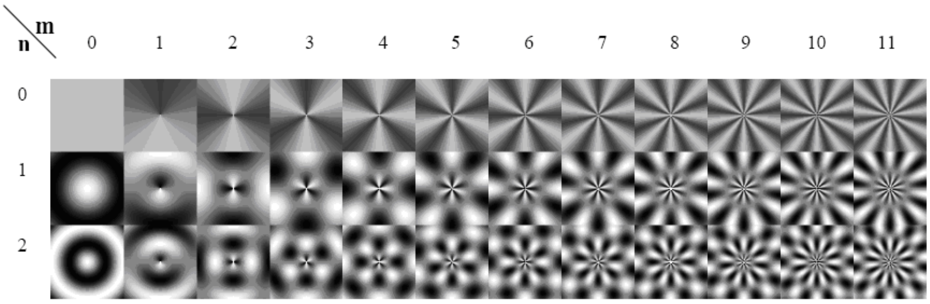


Figure 3.2: Real parts of ART basis functions (n=3, m=12)

Let the image $f^{\alpha}(\rho,\theta)$ be the rotated version of f(ρ,θ) by the angle α around its origin, i.e.,

$$f^{\alpha}(\rho,\theta) = f(\rho,\alpha+\theta) \tag{44}$$

Then, the ART coefficients are obtained as:

$$F_{nm}^{\alpha} = F_{nm}\exp(-jm\alpha) \tag{45}$$

Hence, rotation invariance is obtained by taking the magnitude of the ART coefficients [32].

Just like the Zernike moments, ART doesn't have scale and translation invariance. For scale invariance, ART coefficients are normalized by the magnitude of ART coefficient of order n=0, m=0 [31]. Translation invariance is obtained by centering the transform on the centroid of the object.

3.3.3 Why Not Use Moments?

Although moment-based approaches give promising results for RST invariance and also have the orthogonality property, there are valid reasons for not using such approaches in this study.

First among these is that the moments are very sensitive to noise, which is why moment-based approaches are mostly applied to binary images, or binarized grayscale images using lower order moments [33].

In a study by Abu-Mostafa and Psaltis [34], the recognitive aspects of moment invariants are examined in detail and some very interesting results are obtained. In addition to moment invariants being no longer invariant when noise is present, it is found in the study that moment invariants of Hu suffer from information loss, suppression and redundancy. It was also found that Zernike moment invariants also suffer from information loss, although they don't have information suppression or redundancy.

3.4 Fourier Transform and Its Applications in RST Invariant Algorithms

Although Fourier transform, by itself, is not RST invariant, it has some very useful properties against translation, rotation and scaling, that motivate its usage in a lot of RST invariance techniques.

The discrete Fourier transform (DFT) of f (x_1, x_2) is defined as follows:

$$F(k_1,k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(x_1,x_2) e^{-j2\pi x_1 k_1 / N_1 - j2\pi x_2 k_2 / N_2} \qquad (46)$$

A translation, or a shift in the spatial domain, causes a linear shift in the phase component. Note that both $F(k_1,k_2)$ and $f(x_1,x_2)$ are periodic functions, so the translations are assumed to be circular translation. This leads to the well known result that DFT magnitude is invariant to circular translation, which can be observed from Figure 3.3 and Figure 3.4.

$$f(x_1 + a, x_2 + b) \overset{DFT}{\longleftrightarrow} F(k_1, k_2) exp[-j(ak_1, bk_2)] \tag{47}$$

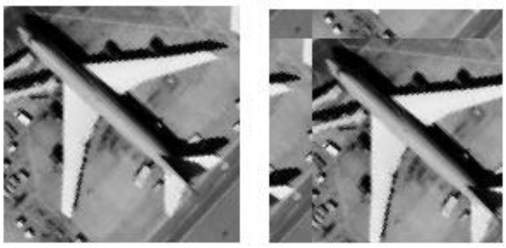


Figure 3.3: An image and its circularly translated version

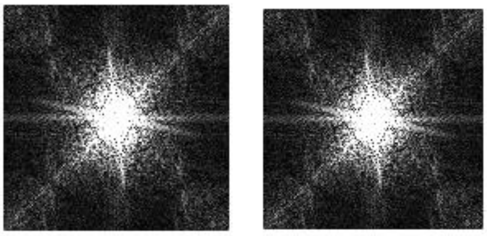


Figure 3.4: Fourier transform magnitudes of the images in figure 3.3

Scaling the image in the spatial domain causes an inverse scaling in the frequency domain, which can be observed from Figure 3.5 and Figure 3.6:

$$f(\rho x_1, \rho x_2) \overset{DFT}{\longleftrightarrow} \frac{1}{\rho} F\left(\frac{k_1}{\rho}, \frac{k_2}{\rho}\right) \tag{48}$$

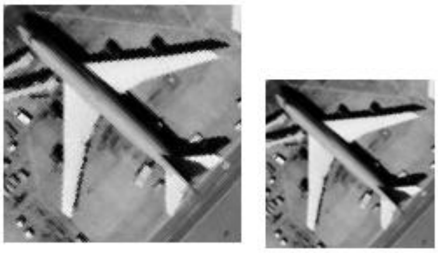


Figure 3.5: An image and its scaled version

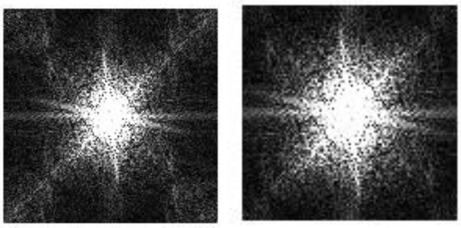


Figure 3.6: Fourier transform magnitudes of the images in figure 3.5

Rotating the image through an angle θ in the spatial domain causes the Fourier representation to be rotated through the same angle:

$$f(x_1 \cos\theta - x_2 \sin\theta, x_1 \sin\theta + x_2 \cos\theta) \\ \overset{DFT}{\longleftrightarrow} F(k_1 \cos\theta - k_2 \sin\theta, k_1 \sin\theta + k_2 \cos\theta) \quad (49)$$

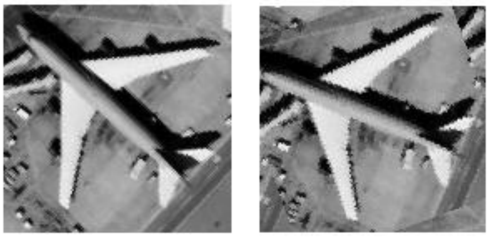


Figure 3.7: An image and its rotated version

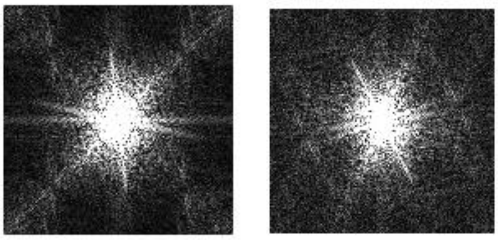


Figure 3.8: Fourier transform magnitudes of the images in figure 3.7

The magnitude of the Fourier transform is commonly used for translation invariance and the properties of Fourier transform against rotation and scale will be used in the next section, in log-polar mapping, to transform rotation and scaling to translation.

3.5 Log-Polar Mapping

Log-polar mapping, by itself, is also not an RST invariance technique; however it has some very important properties that make it very useful in applications that require RST invariance. Before these properties are studied, it is necessary to understand how this mapping works.

Log-polar mapping is a nonlinear transformation that is based on human vision. Human vision has two areas, fovea centralis at the center of visual axis, which has high resolution, and the peripheral, where the resolution decreases according to the distance from the fovea centralis [55]. The log-polar mapping maps the points on the Cartesian plane (x,y) to points in the log-polar plane (ξ,ή) similar to the human vision system.

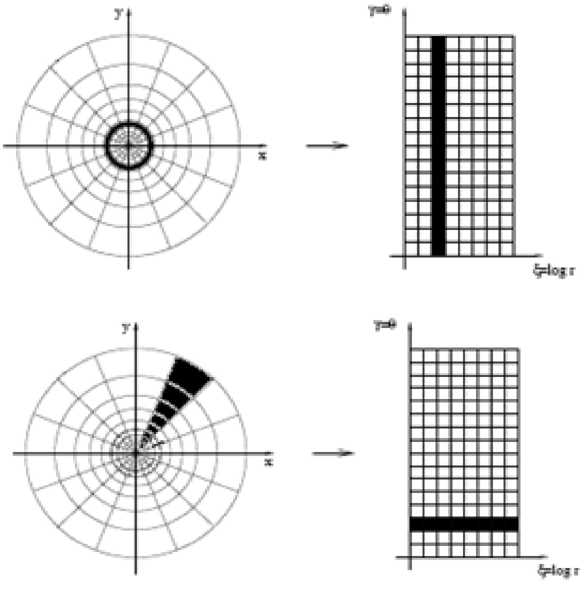


Figure 3.9: Log-Polar Mapping in Coordinates

The coordinate transformation by log-polar mapping is described by the following two equations:

$$\varepsilon = \log\sqrt{x^2 + y^2} \qquad (50)$$

$$\eta = \tan^{-1}\left(\frac{y}{x}\right) \qquad (51)$$

In log-polar domain, one axis gives us the angular value, and the other axis gives us the logarithmic representation of the radial value. This mapping is done as follows: For each (ξ,ή) value, the corresponding (x,y) value is calculated, then the value of the image is obtained at this (x,y) using approximation or interpolation. This value is written to that (ξ,ή) coordinate of the log-polar plane. Given below in Figure 3.10 are an image and the image obtained by taking the log-polar mapping of that image.

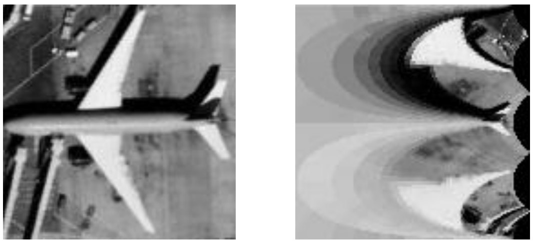


Figure 3.10: An image and its log-polar mapping

The advantage of using Log-Polar over Cartesian coordinate representation is because any rotation and scale in Cartesian coordinate is represented as a shift in the angular direction and a shift in the log-radius direction in Log-Polar coordinate, respectively [56]. In other words, log-polar mapping has the useful property to convert scale and rotation into translation. This can be seen from the equations below:

$$(x, y) \leftrightarrow (\mu, \theta) \tag{52}$$

$$(\rho x, \rho y) \leftrightarrow (\mu + \log \rho\,, \theta) \tag{53}$$

$$(x\cos(\theta+\delta) - y\sin(\theta+\delta)\,, x\sin(\theta+\delta) + y\cos(\theta+\delta)) \leftrightarrow (\mu, \theta+\delta) \tag{54}$$

The above equations demonstrate that image scaling in Cartesian domain results in a translation along the log-radius axis in log-polar domain, and the image rotation in Cartesian domain results in a circular translation along the angle axis in log-polar domain.

However, log-polar mapping is not translation invariant. Hence, although log-polar transform has important properties for scale and rotation invariance, these advantages can be acquired only when the origin of the transformation for both the template and target in the candidate image are matched [56]. In Fourier-Mellin transform, which is explained in the next section, log-polar mapping is used after the magnitude of Fourier transform of the image is taken, to solve this problem.

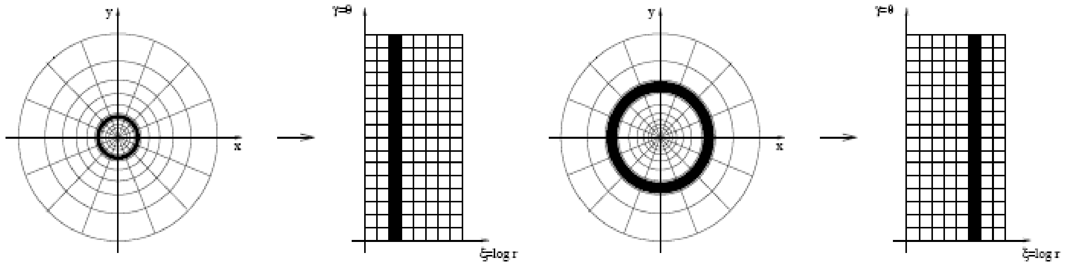


Figure 3.11: Scaling property of log-polar mapping

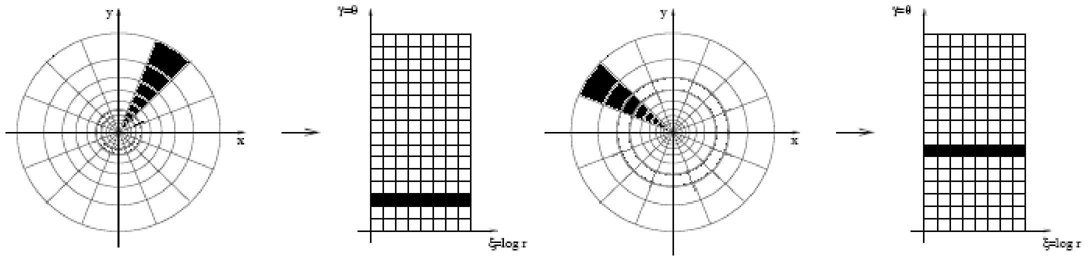


Figure 3.12: Rotation property of log-polar mapping

3.6 Fourier-Mellin Transform

The Fourier-Mellin transform is a useful tool for image recognition because its resulting spectrum is invariant in rotation, translation and scale. The Fourier Transform itself is translation invariant and its conversion to log-polar coordinates converts the scale and rotation differences to vertical and horizontal translations that can be measured. A second DFT, called the Mellin transform gives a transform-space image that is invariant to translation, rotation and scale.

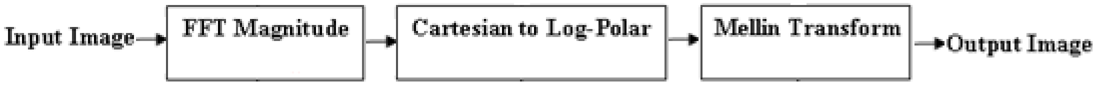


Figure 3.13: Fourier-Mellin transform process

We can write the relationship of an image $i_0(x,y)$, and a rotated, scaled, and translated version of the image, $i_1(x,y)$, as follows:

$$i_1(x,y) = i_0(\sigma(x\cos\alpha + y\sin\alpha) - x_0, \sigma(-x\sin\alpha + y\cos\alpha) - y_0) \quad (55)$$

where the image is scaled by σ, rotated by α, and translated by (x_0,y_0).

The Fourier transforms of $i_0(x,y)$ and $i_1(x,y)$ are $I_0(u,v)$ and $I_1(u,v)$ respectively, and their magnitudes are related by:

$$|I_1(u,v)| = |\sigma|^{-2} \cdot \left|I_0\left(\sigma^{-1}(u\cos\alpha + v\sin\alpha), \sigma^{-1}(-u\sin\alpha + v\cos\alpha)\right)\right| \quad (56)$$

which is seen to be independent of the translation parameters, i.e., is translation invariant.

Rewriting the above equation in log-polar coordinates, we arrive at:

$$|I_1(u,v)| = |\sigma|^{-2} \cdot |I_0(\sigma^{-1}e^{\rho}\cos(\theta - \alpha), \sigma^{-1}e^{\rho}\sin(\theta - \alpha))| \quad (57)$$

or:

$$|I_1(u,v)| = |\sigma|^{-2} \cdot |I_0(\rho - \ln\sigma, \theta - \alpha)| \quad (58)$$

The equation above is independent of the first translation parameters and the rotation and scale parameters have transformed into translation parameters.

Another Fourier transform gives us:

$$F_1\left(w_\rho, w_\theta\right) = |\sigma|^{-2} \cdot e^{-j\left(w_\rho \cdot \ln\sigma + w_\theta \cdot \alpha\right)} \cdot F_0\left(w_\rho, w_\theta\right) \quad (59)$$

The Fourier magnitudes of the two log-polar mappings are related by:

$$\left|F_1\left(w_\rho, w_\theta\right)\right| = |\sigma|^{-2} \cdot \left|F_0\left(w_\rho, w_\theta\right)\right| \quad (60)$$

As can be seen from the equation above, at the end of the Fourier-Mellin transform, we obtain translation and rotation invariance, and scaling only results in a constant scaling of the whole amplitude of the Fourier-Mellin spectrum. This scaling will not present a problem since normalized cross-correlation is used as the matching similarity metric.

Although FMT is a successful method for RST invariance, it has some important weaknesses that should be considered. The first and foremost among these is the loss of phase information since FMT depends on the magnitudes of Fourier transforms. The whole information present in the images is not preserved and the possibility of false detection increases [33]. Also, the cross-correlation of the Fourier-Mellin transforms generally yield a very broad maximum because of the loss of phase information, and hence FMT becomes unreliable both for identification and localization of an object in the image [33].

A possible approach to overcome this problem is not to take the second Fourier transform in FMT and use phase correlation to obtain the translation values in the log-polar mapped image [33], [38], [57]. These translation values correspond to rotation and scale parameters between the two images. These parameters are used to reconstruct the image. Then, the reconstructed image and the reference image are matched using phase correlation to find the optimal translation parameters between the two images.

The method, however, has difficulty in recovering large scales and rotations. This difficulty can be understood by realizing that large rotation and scale differences exacerbate the border effects when computing the Fourier transform. A large translation or scale introduces additional pixel information that can dramatically alter the Fourier coefficients [38].

It should be noted that even in this method, Fourier transform magnitude has to be used before the log-polar mapping for translation invariance while obtaining rotation and scale parameters. Hence, we still lose the phase information of the image. To overcome this, higher-order spectra can be used instead of the Fourier transform for translation invariance. Bispectrum, which is the most frequently used higher-order spectra for translation invariance, will be studied in the next section.

It is important to note that the literature is replete with synthetic examples for the Fourier–Mellin registration method. In particular, a reference image is always matched against a scaled and rotated version of itself. This serves to defer the problem of handling the fine details introduced by an actual optical zoom. Conversely, when the image undergoes minification, translation, or rotation, additional real data seeps into the target image, not just black pixels [38]. Note that artificial black backgrounds can help register two images, because they ensure that the same underlying content is considered [38].

3.7 Bispectrum

As stated earlier, amplitude spectrum does not provide a full description of the image contents, because it lacks the information carried by the phase spectrum [52]. Hence, other translation invariance representations are searched. In practice, the new representation should have its behavior against rotation and scale similar to the amplitude spectrum, for easier estimation of scale and rotation parameters [52]. Fortunately, the amplitude spectrum is not the only spectral representation that is translation-invariant. Certain higher order spectrums also have the same property. These spectrums are defined by:

$$\Psi_n(u_1,u_2,\ldots,u_n) = F^*(s)\prod_{i=1}^{n} F(u_i) \tag{61}$$

where u_i with i = 1, …, n are vectors in the 2-D frequency space, and $s = u_1 + u_2 + \ldots + u_n$.

The first-order spectrum is the power spectrum, which is the squared amplitude spectrum, and is obtained by taking n=1:

$$\Psi_1(u) = F(u) \cdot F^*(u) \tag{62}$$

The second-order spectrum, which is the bispectrum, is obtained by taking n=2:

$$\Psi_2(u_1,u_2) = F(u_1) \cdot F(u_2) \cdot F^*(u_1{+}u_2) \tag{63}$$

The bispectrum is a triple product of Fourier coefficients, and is a complex-valued function of two frequencies, similar to the power spectrum which is a second order product of Fourier coefficients and a function of only one frequency.

Bispectrum retains both the amplitude and the phase information from the Fourier transform of a signal, so the information carried by the phase is not lost. Because of this, bispectrum is commonly used in signal –and image- reconstruction [58], [59], [60].

Bispectrum is translation invariant because linear phase terms are cancelled in the triple product that defines the bispectrum.

Also, bispectrum is zero for Gaussian noise and thus provides high noise immunity [51].

Assuming that F(u) is an N-by-N DFT of an image, the bispectrum becomes a N-by-N-by-N-by-N matrix. It is therefore not practical to evaluate the whole bispectrum. A better solution, that is commonly used, is to take 2-D slices of this 4-D spectrum [52]. These slices are generally defined as:

$$S_k(u) = \Psi_2(u, k \cdot u) \quad , \forall k \in R \tag{64}$$

Although a slice is only a small portion of the whole bispectrum, in the applications found in the literature, it is stated that the reconstruction is still possible and no essential information has been lost [52]. It is also stated that it is possible to obtain slightly better classification results by using multiple slices [54].

The scaling and rotation properties of the bispectrum slices and the whole bispectrum are the same. These properties are basically the same as for the Fourier transform, which indicates that we can directly use the bispectrum slices in the same way as the amplitude spectrum in the Fourier-Mellin transform to estimate the scale and rotation [52]. The scaling and rotation properties of the bispectrum slices are given by:

$$f(\alpha x, \beta y) \overset{B}{\Rightarrow} \frac{1}{|ab|^3} S_k\left(\frac{u}{a}, \frac{v}{b}\right) \tag{65}$$

$$f(r, \theta + \alpha) \overset{B}{\Rightarrow} S_k(w, \phi + \alpha) \tag{66}$$

It should be noted that although the bispectrum provides noise immunity and translation invariance without losing the phase information, bispectrum is sensitive to illumination changes just like the amplitude spectrum. Hence, non-uniform illumination changes may cause significant problems in template matching or image alignment [52].

CHAPTER 4

THE METHODS

4.1 Introduction

Presented in this chapter are the step-by-step descriptions of the methods used in this study. The reasons for using each method are also presented and the strengths and drawbacks of the methods are investigated. Before the methods form into their final versions, experiments and observations of some of these drawbacks have caused us to make some changes on the method, which are also explained accordingly.

4.2 ATR Using Template Matching in Fourier-Mellin Domain:

As explained in section 3.6, Fourier-Mellin transform can be used to obtain RST invariance. In our first method, the template and the sub-images to be compared with the template are transformed into the Fourier-Mellin domain, which is RST invariant, and compared in this domain using normalized cross-correlation.

4.2.1 Step-by-step Description of the Method

In this section, step-by-step description of the method is presented.

1) The input – or test – image is formed into sub-images of the same size as the template image. The blocks can be overlapping or non-overlapping.
2) Fourier transforms of both the template and the sub-images are taken and the phases are disregarded by taking only the magnitude part to obtain translation invariance.
3) Log-polar mappings of the Fourier transform magnitudes are taken to transform scale and rotation into translation.

4) Another Fourier transform follows, also in which the phases are disregarded by taking the magnitudes and which provides scale and rotation invariance.
5) The resulting images in the Fourier-Mellin domain are passed into a similarity metric, which is in our case NCC, and compared with a threshold. The sub-images with correlation results higher than the threshold are the sub-images which contain the target in the template image.

Given below in Figure 4.1 is the flowchart of this method.

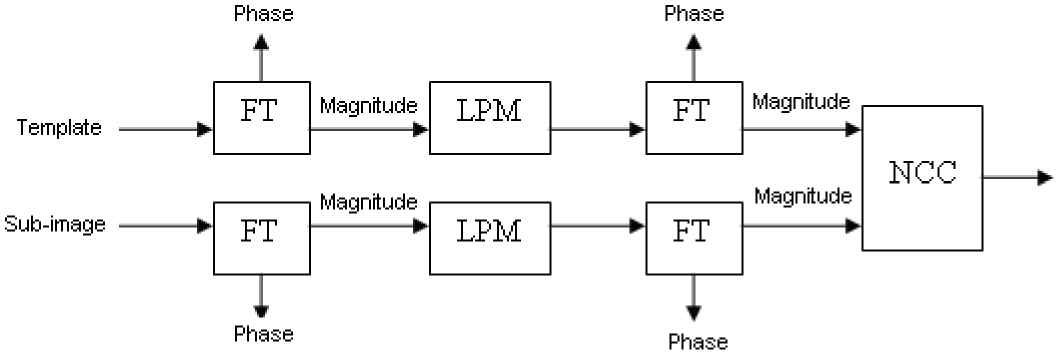


Figure 4.1: Flowchart of ATR using template matching in Fourier-Mellin domain

4.2.2 Drawbacks of the Method

4.2.2.1 Loss of Phase Information

As can be seen from figure 4-1, this method involves taking the Fourier transform magnitude two times in order to obtain RST invariance. This results in the loss of phase information two times. The information carried by the phase of the Fourier transform of the template and the sub-images are lost in the taking of the first Fourier magnitude and the phase information of the log-polar mappings of the Fourier magnitudes are lost in the second taking of the Fourier transform magnitudes.

After the conducted experiments, which are presented in the Chapter 5, it was observed that the phase information is essential and critical, and its loss should be avoided. The loss results in very high correlation results for each sub-image, and hence a very small dynamic range. This can be explained by noting that the loss of the essential information contained in the phase results in transformation into a domain in which, although RST invariance is satisfied, each image becomes very similar.

To partially solve this problem, another method similar to those in some studies in the literature, based on reconstruction, was constructed which gets rid of the need for the second Fourier magnitude. This method is studied in Section 4.3 in detail.

4.2.2.2 Very Small Dynamic Range

As explained in section 4.2.2.1, this weakness is a direct consequence of the loss of phase information caused by taking the Fourier transform magnitudes while transforming into the Fourier-Mellin domain.

4.3 ATR Using Template Matching Based on Reconstruction by Fourier-Mellin Transform and Phase Correlation

In this method, rather than transforming the template and the sub-images to the Fourier-Mellin domain, reconstruction by phase correlation is used to compare the images in image domain.

After the Fourier transform magnitude and the log-polar mapping, the second Fourier transform magnitude is not taken. Rather, the properties of log-polar mapping and phase correlation are used to obtain the rotation and scale parameters between the template and the sub-image to be compared. As was explained in the previous chapter, the Fourier magnitude provides translation invariance while preserving the rotation and scale, and by log-polar mapping rotation and scale parameters are transformed into translation. Then, phase correlation is used to obtain the translation, which corresponds to the rotation and scale parameters, and reconstruct the sub-image so that there is no rotation or scale variation between the template and the sub-image. After this reconstruction, the reconstructed image and template are phase correlated to obtain the translation parameter.

4.3.1 Step-by-step Description of the Method

In this section, step-by-step description of the method is presented. Note that the algorithm of this method is modified in section 4.3.2.2, hence the step-by-step description and the flowchart presented in this section are not the final versions for this method.

1) The input – or test – image is formed into sub-images of the same size as the template image. The blocks can be overlapping or non-overlapping.
2) Fourier transforms of both the template and the sub-images are taken and the phases are disregarded by taking only the magnitude part to obtain translation invariance.
3) Log-polar mappings of the Fourier transform magnitudes are taken to transform scale and rotation into translation.
4) Phase correlation is taken between the LPM of the FT magnitude of the template and the LPM of the FT magnitude of each sub-image to obtain the translations, which correspond to the scale and rotation between the sub-images and the template.
5) Each sub-image is reconstructed using its rotation and scale parameters.
6) The reconstructed sub-images and the template image are phase correlated to obtain the translations between the images.
7) The reconstructed sub-images are again reconstructed using their translation parameters.
8) The reconstructed sub-images are normalized cross-correlated with the template image in the image domain and the results are compared with a threshold. The sub-images with correlation results higher than the threshold are the sub-images which contain the target in the template image.

Given below in Figure 4.2 is the flowchart of this method.

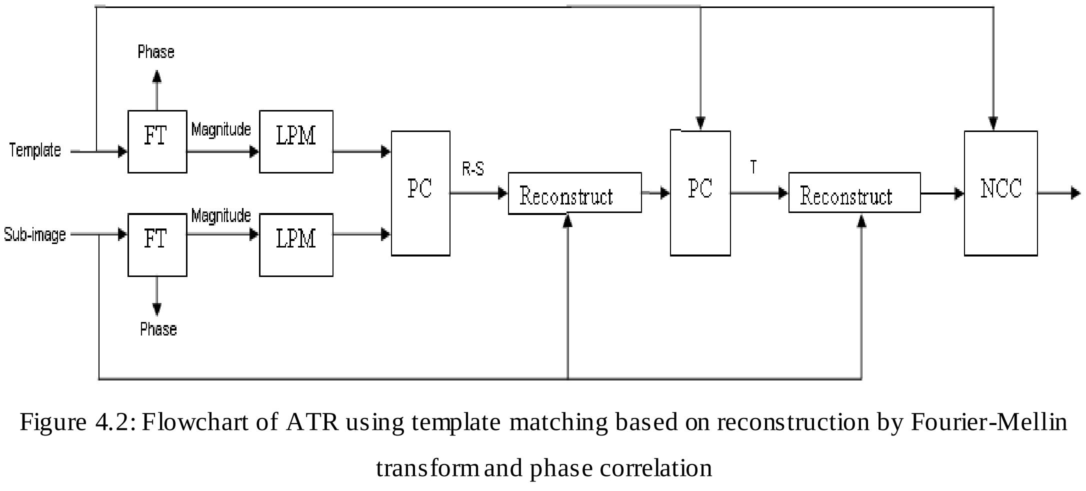


Figure 4.2: Flowchart of ATR using template matching based on reconstruction by Fourier-Mellin transform and phase correlation

4.3.2 Drawbacks of the Method

4.3.2.1 Multiple Peaks in Phase Correlation

As explained before, in the literature, this method is almost always used to obtain the RST parameters between an image and its rotated, scaled and translated versions. However, in our problem, only the target in the images will be the same and the background, and hence an important part of the images, will be different. After the experiments, it was observed that this results in not a clean and single peak for the phase correlations, but rather in a peak cloud, in which the largest peak is not necessarily the correct one to obtain the correct RST parameters. Because of this, some modifications on the method were made.

Multiple peaks were used to obtain rotation and scale parameters, the sub-image was reconstructed to a number of images using each rotation-scale parameter pair, followed again by a phase correlation between each of these images and the template to obtain, again with multiple peaks, the translation parameters. The images are again reconstructed to a larger number of images using the translation parameters. At the end, each image is compared with the template using NCC in image domain, and the image with the largest similarity is the one reconstructed with the correct RST parameters.

The step-by-step description of the revised version of the method is as follows:

1) The input – or test – image is formed into sub-images of the same size as the template image. The blocks can be overlapping or non-overlapping.
2) Fourier transforms of both the template and the sub-images are taken and the phases are disregarded by taking only the magnitude part to obtain translation invariance.
3) Log-polar mappings of the Fourier transform magnitudes are taken to transform scale and rotation into translation.
4) Phase correlation is taken between the LPM of the FT magnitude of the template and the LPM of the FT magnitude of each sub-image to obtain the translations, which correspond to the scale and rotation between the sub-images and the template. However, multiple peaks are obtained for each sub-image, which correspond to multiple R-S pairs.

5) Each sub-image is reconstructed using its each rotation and scale parameter pair.
6) The reconstructed sub-images and the template image are phase correlated to obtain the translations between the images. Again, multiple peaks are used, which correspond to more than one possible translation parameter.
7) The reconstructed sub-images are again reconstructed using each of their translation parameters.
8) The reconstructed sub-images are normalized cross-correlated with the template image in the image domain. For each sub-image the highest correlation result is accepted as corresponding to the sub-image which was reconstructed using true RST parameters.
9) The correlation results are compared with a threshold. The sub-images with correlation results higher than the threshold are the sub-images which contain the target in the template image.

In our experiments, it was observed that to obtain the correct rotation and scale parameters 30 largest peaks was necessary and sufficient, while for the translation parameters, 10 peaks were more than enough in all cases. Note that the near neighborhood of each taken peak is disregarded for the next possible peak location candidates.

4.3.2.2 Loss of Phase Information

Although in this method, we don't use the second FT magnitude of the method in section 4.2, we still have the first FT magnitude for translation invariance. Hence, the phase information of the template and the sub-images are still lost.

It should be noted that, since in this method, instead of using the final similarity metric in Fourier-Mellin domain, we use it in image domain, this loss of phase information does not affect the final correlation results as in the method of section 4.2. However, the loss of phase degrades the results of phase correlation taken after the LPM, and in some cases, the rotation and scale parameters may not be correctly obtained.

Note that in the literature, since this method is almost always used to find the RST parameters between an image and its scaled, rotated and translated version, this loss is not as important as in our case.

To truly solve this problem, we either have to forego translation invariance, or find another transform which behaves against rotation and scale similar to Fourier transform, but is translation invariant without losing the phase information. As explained in the previous chapter, bispectrum satisfies these properties. The method in which bispectrum is used in place of the FT in this method is studied in the next section

4.4 ATR Using Template Matching Based on Reconstruction by Bispectrum, Log-Polar Mapping and Phase Correlation

Bispectrum was used instead of FT in the previous method to prevent the loss of phase information, and hence overcome a weakness. The rest of the method is exactly the same as the previous method.

4.4.1 Step-by-step Description of the Method

In this section, step-by-step description of the method is presented. This method was developed after the previous method of section 4.3 was revised into its final form; hence multiple peaks are used in phase correlation steps.

1) The input – or test – image is formed into sub-images of the same size as the template image. The blocks can be overlapping or non-overlapping.
2) Bispectrum slices of both the template and the sub-images are taken to obtain translation invariance.
3) Log-polar mappings of the bispectra are taken to transform scale and rotation into translation.
4) Phase correlation is taken between the LPM of the bispectrum of the template and the LPM of the bispectrum of each sub-image to obtain the translations, which correspond to the scale and rotation between the sub-images and the template. However, multiple peaks are obtained for each sub-image, which correspond to multiple R-S pairs.
5) Each sub-image is reconstructed using its each rotation and scale parameter pair.
6) The reconstructed sub-images and the template image are phase correlated to obtain the translations between the images. Again, multiple peaks are used, which correspond to more than one possible translation parameter.

7) The reconstructed sub-images are again reconstructed using each of their translation parameters.
8) The reconstructed sub-images are normalized cross-correlated with the template image in the image domain. For each sub-image the highest correlation result is accepted as corresponding to the sub-image which was reconstructed using true RST parameters.
9) The correlation results are compared with a threshold. The sub-images with correlation results higher than the threshold are the sub-images which contain the target in the template image.

Given below in figure 4.3 is the flowchart of this method.

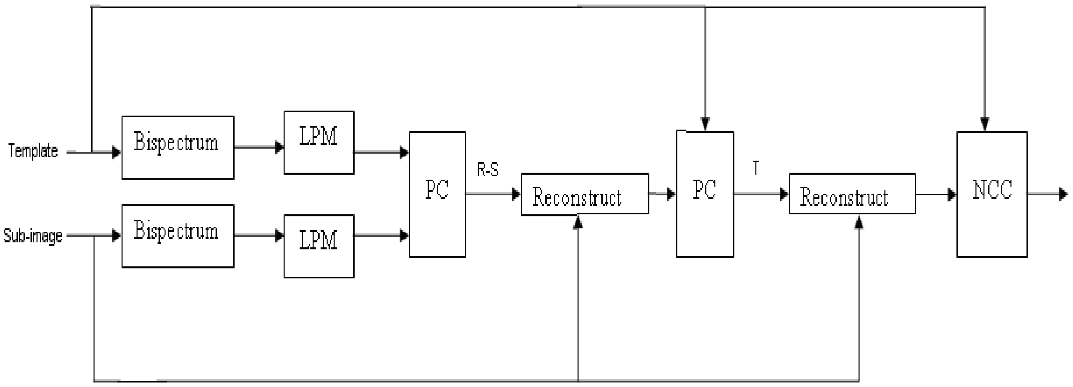


Figure 4.3: Flowchart of ATR using template matching based on reconstruction by bispectrum, log-polar mapping and phase correlation

4.4.2 Drawbacks of the Method

4.4.2.1 Multiple Peaks in Phase Correlation

This problem is still there, and has the same effect as in the previous method, namely the computation time is increased to obtain better results. The same number of peaks is used for this method as in the previous method.

4.4.3 Slice matters

As explained in section 3.7, for an NxN image, the bispectrum is NxNxNxN, which is not only impractical, but also impossible for our template image sizes, to implement. Hence, bispectrum slices are used instead of the whole bispectrum. These slices are obtained as:

$$S_k(u) = \Psi_2(u, k \cdot u) \quad , \forall k \in R \tag{67}$$

In the literature, in most studies, the choice of which slice to use does not affect the results since bispectrum is mostly used as a feature extraction tool. However, in our experiments, it was observed that the choice of the slice affects the results.

In a small number of studies, the whole bispectrum is observed and the slice corresponding to the highest SNR values are taken. However, in 2-D, we cannot obtain the 4-D bispectrum in MATLAB, and hence cannot use this method. Instead, a more common approach is adopted, which involves using more than one bispectrum slice and for each sub-image taking the slice which gives the higher result for that sub-image. Computationally, this is a very time consuming solution, since the same operations will be applied for each slice. A better solution can be found in a future study.

4.5 Common Points of Consideration

There are some common points of consideration that do not vary depending on the method used, but are results of the overall approach or caused by. These drawbacks and concepts are detailed in this section.

4.5.1 Overall Drawbacks of the Approach

Beside the presented drawbacks for each method, there are also other important drawbacks that are not associated by the specific methods, but rather are caused by the overall approach, namely using template matching for RST invariant ATR.

4.5.1.1 Background

The most important problem is the fact that in almost all cases, the template image does not only consist of the target, but rather additional data in the form of background also leaks in, due to the fact that most of the objects are not shaped as perfect squares or circles.

The additional data in the form of background causes all the RST invariance methods to perform worse than their other applications in the literature, because the backgrounds in the sub-images are different than the background in the template image, as expected.

The background cannot be modeled as an additive noise to the template and hence cannot be gotten rid of using the standard methods of noise removal. The straightforward solutions to this problem, namely segmentation and/or shape analysis cannot be used since they contradict the reason of using template matching. Hence, the problem cannot be solved completely. Rather, the effect of the background variations can be lessened using some simple techniques such as windowing.

4.5.1.1.1 Windowing

Windowing can be used to reduce the effects of background and make translation appear as circular translation, and hence increase the dynamic range. Three types of windowing functions were studied in this thesis and these are Gaussian, Hanning and Hamming windows. Hanning filter proved to be the most successful among these. The best performing filter size was chosen as 4/3 times the template –and sub-image- size. The windowing tests are studied and analyzed in Chepter 5 in detail.

4.5.1.2 Illumination

Considering that the images are illumination distributions, the importance of illumination in image processing can be better understood. While using data that is exposed to sunlight and affected by weather conditions, this importance increases exponentially.

The angle that the target in the image receives the light of the sun greatly affects the results, so much that at least two templates should be used for each target, one for each side of receiving the light. This is not caused by a drawback of our methods, but rather is unavoidable while using template matching since the illumination distribution of the image, and hence of the target, varies greatly depending on the incidence angle of the light and the similarity metrics such as normalized cross-correlation are drastically affected.

4.5.2 Important Concepts to Pay Attention To

4.5.2.1 Centering the DC Component in FT

In MATLAB®, the FT results in a matrix in which the DC component is the first –uppermost left- pixel. However the zero-frequency should be in the center pixel to truly observe the rotation property of FT. Otherwise, the frequency domain representation would have to be rotated around the uppermost left pixel since FT rotates around the DC component, and combined with periodicity, calculation complexity would increase. A function called "fftshift" is used to shift zero-frequency component to the center of the spectrum, which performs as in the figure 4.4 below. The frequencies increase by getting farther away from the center.

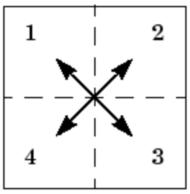


Figure 4.4: How the "*fftshift*" works

Note that "fftshift" is not enough to center the DC component completely, since there is a one pixel shift. To solve this problem, one column and one row are added to the right side of and under the frequency domain representation respectively. These column and row are the same as the leftmost column and uppermost row, since the FT is periodic.

4.5.2.2 The Discord between Circular and Square Operations

LPM is a transformation that takes a circular area from the Cartesian domain and maps it into a rectangular area in the log-polar domain. However, FT and bispectrum are defined on squared areas on 2-D. Because of this disunity, it is necessary to determine in which way the circular area to operate LPM on will be selected. The obvious two choices are the inner circle and the outer circle as shown in the figure below for an image.

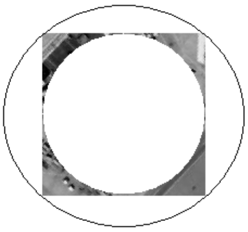


Figure 4.5: Inner and outer circles for LPM

In this study, the inner circle is used, because LPM is applied after FT or bispectrum, and the data disregarded by taking the inner cirle - data corresponding to the corners of the squared area – corresponds to very high frequencies, which do not affect the reconstructed image in a way noticeable by eye.

In addition, in this way, LPM can also be applied in image domain, besides frequency domain. Otherwise, in using the outer circle, outside of the squared area containing the data would have to be filled with data from neighboring sub-images – which is not applicable for the template image- or taken as zero or NaN, which would result in a situation such that for each template and sub-image the same locations of each LPM would always be equal, resulting in a domination of rotation parameters of 0, 90, 180 and 270 degrees being determined by phase correlation.

4.5.2.3 Interpolation in LPM

As explained before in section 3.5, while performing LPM, for each coordinate of the log-polar mapped image the corresponding (x,y) value from the Cartesian image is calculated, then the value of the image is obtained at this (x,y) using approximation or interpolation, since the (x,y) coordinates are not necessarily –and indeed in most cases- integers.

Both nearest neighbor approximation and bilinear interpolation were used in this study and overall, better results were obtained with bilinear interpolation than with nearest neighbor, as expected. Bilinear interpolation was adopted for the rest of this study.

Bicubic interpolation was also used, but it was observed that it did not provide a significant improvement over bilinear interpolation.

4.6 Exhaustive Approach

To truly discern the necessity of our RST invariant methods, the performances should be compared with the exhaustive case, namely using templates at least for every rotation, and possibly for every translation and scale, though these can be avoided.

Exhaustive approach has been implemented in this study to be compared in performance to the RST-invariant methods. 36 rotated templates have been used for each target. Scale invariance is obtained as a first step using the information on the altitude that the test image is obtained and the size of the target. Translation invariance is obtained by reconstruction using phase correlation.

However, the exhaustive method has two drawbacks. One of the drawbacks is that the exhaustive case is generally not feasible, since template images containing each target in each angle of rotation can't be obtained in most cases. Note that generating these template images artificially can be used to avoid this drawback.

The second drawback is the computation time. A possible approach to reduce this computation time involves using eigenvalue reduction. In this approach, the template images of the target with different rotation angles are represented by a smaller number of eigenvalues and eigenvectors. This approach will be examined in the next section.

4.7 Eigen Approach

Eigen approach has been used in this thesis for performance comparison with the RST-invariant methods. The method has been derived from the book of Trucco and Verri [61]. The step-by-step description of the method used in this study is as follows:

1) For each target, the template images which contain the target's rotated versions are taken. In this study, a number of 36 manually rotated template images were used for each target. Ideally a number of 360 images, one for each degree of rotation, should be used, but this is obviously not feasible in real-life applications.
2) The energy for each template image is normalized
3) DFT is taken for translation invariance
4) The template images are represented as vectors
5) The average vector is calculated for each target over these vectors
6) The covariance matrix of the vectors is computed
7) Eigenvalues and eigenvectors corresponding to the target are computed. In this study, since 36 template images were used for each target, there are a total of 36 eigenvalues for each target.

8) The first k eigenvalues and eigenvectors are kept, and the rest is discarded. In this study k is taken as equal to one fourth of the total number of eigenvalues, which is 9.
9) The k-dimensional eigenspace points corresponding to each vector are computed. Hence, 36 k-dimensional vectors are obtained.
10) For each sub-image, the normalization, DFT and vector representation steps are done. Then, the k-dimensional eigenspace point for the sub-image is computed.
11) The closest k-dimensional eigenspace point to the k-dimensional eigenspace point of the sub-image is found.
12) NCC between the corresponding template image and the sub-image is calculated.
13) A threshold is used to discern if the sub-image contains the target.

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 Introduction to Experiments

In this chapter, the results of the conducted experiments are presented and studied in detail. The chapter is formed up according to the nature of the experiments conducted. Note that although some experiments are to give the overall performance of the methods, some are to discern how a certain variation or additional operation affects the performance. These variations include whether to use overlapped blocks, windowing, multiple bispectrum slices and artificial backgrounds for the templates.

After the tests on variations and parameters, the five methods used in this study are compared in performance. This comparison is conducted in a subgroup of the database. After the performance comparison tests, the best performing method is analyzed in performance in more detail and in various situations.

The performances of the methods are measured in terms of receiver operation characteristics (ROC), i.e. recall-vs-precision graphs. In terms of true positive (hit), false positive (false alarm) and false negative (miss), these are obtained as:

$$Precision = \frac{tp}{tp + fp} = \frac{Hit}{Hit + False\ Alarm} \tag{68}$$

$$Recall = \frac{tp}{tp + fn} = \frac{Hit}{Hit + Miss} \tag{69}$$

5.2 Test Data

In this study, the data for testing was collected from all over the world using *Google Earth* ®. The database was mostly focused on airplanes - both civilian and military, while containing other types of test data also.

5.3 Experiments on Variations and Parameters

In this section, the experiments are conducted not to give the overall performance of a specific method, but rather to determine how a certain variation affects the results or the performances of the methods.

5.3.1 Overlap vs. Non-overlap

The sub-images are taken in the size of the templates, but they can either be taken as non-overlapping or overlapping. Even though the methods are translation invariant, their basic forms are based on the assumption that the target is in the center of the sub-image. Our revision of the methods based on using multiple peaks and fixing translation after fixing the rotation and scale overcomes this to some extent, but the larger part of the target in the sub-image the better the results will be, obviously.

The tests on overlapping were conducted with the method presented in Section 4.5 and with non-overlapping and overlapping blocks with the overlap ratios of ½ and ¾. A single bispectrum slice was used for the tests in this section. Although a large number of template and test images were used in these experiments, only two of such template–test image pairs are presented here for space considerations. These two examples are chosen such as to convince the reader that the overall results of these experiments are the same as those presented in Section 5.3.1.4. While presenting the test images, the sub-images are separated by red lines. The blocks that are bound in green are those sub-images that result in a final NCC result greater than a given threshold, taken as 0.5 here. The NCC results are also presented in all of these sections for the corresponding tests. Note that the x-axis of the NCC results corresponds to the number of the sub-image being tested.

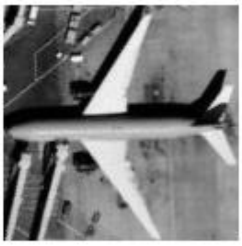


Figure 5.1: Template image for the first overlapping test

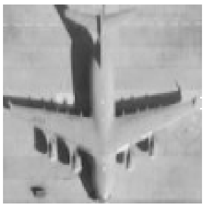


Figure 5.2: Template image for the second overlapping test

5.3.1.1 Non-overlapping Blocks

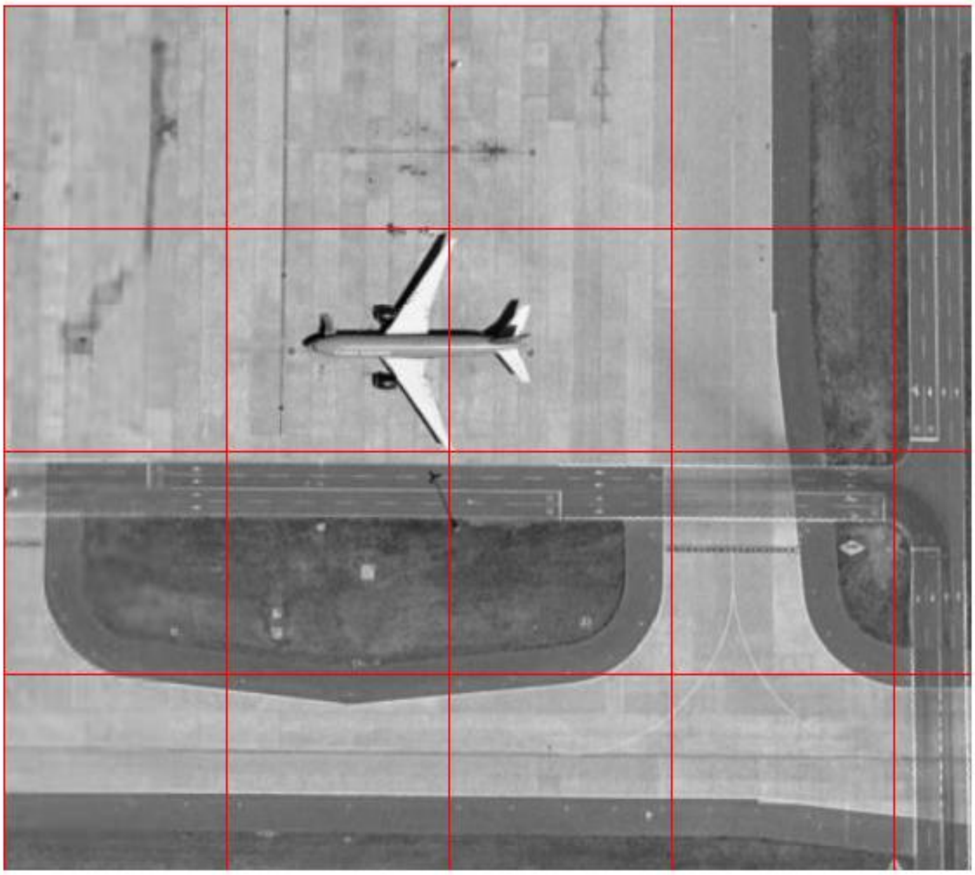


Figure 5.3: Test image for the first template with the non-overlapping sub-images shown

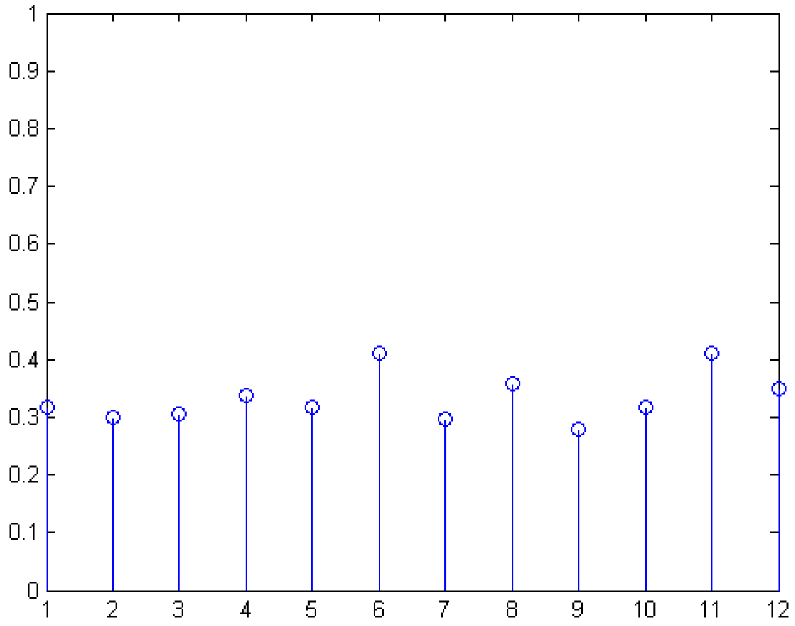


Figure 5.4: Final NCC results for Figure 5.3

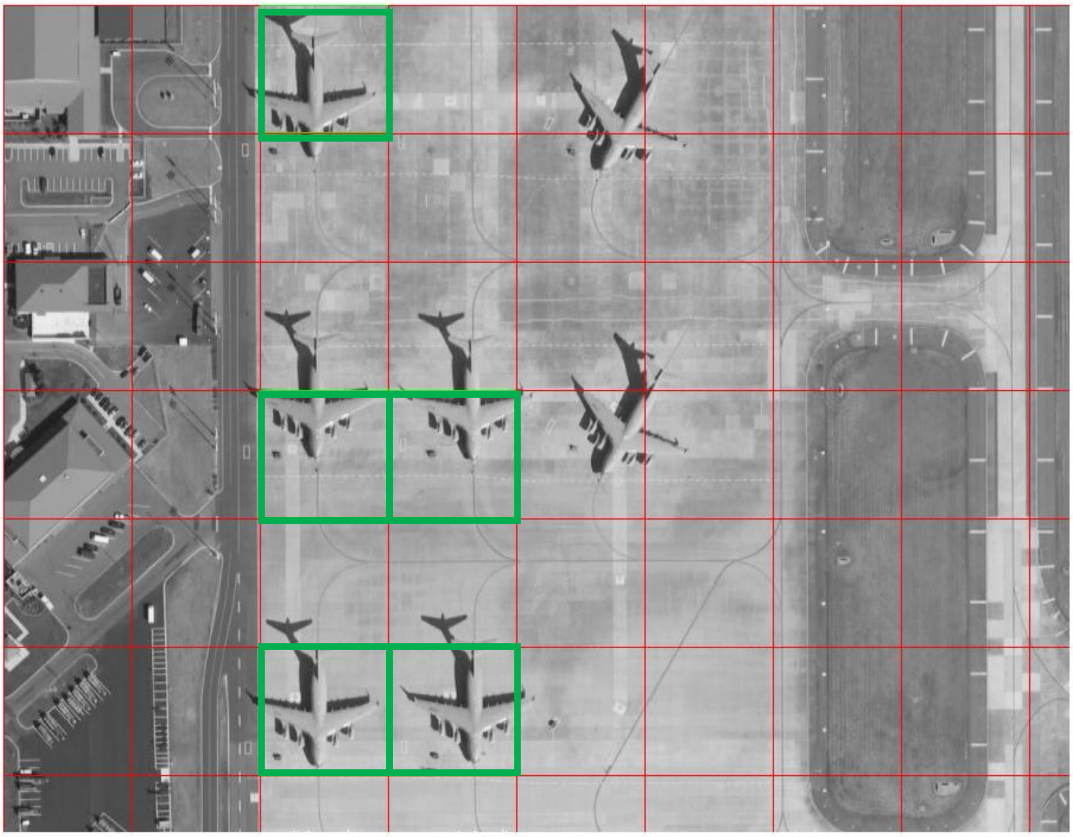


Figure 5.5: Test image for the second template with the non-overlapping sub-images shown

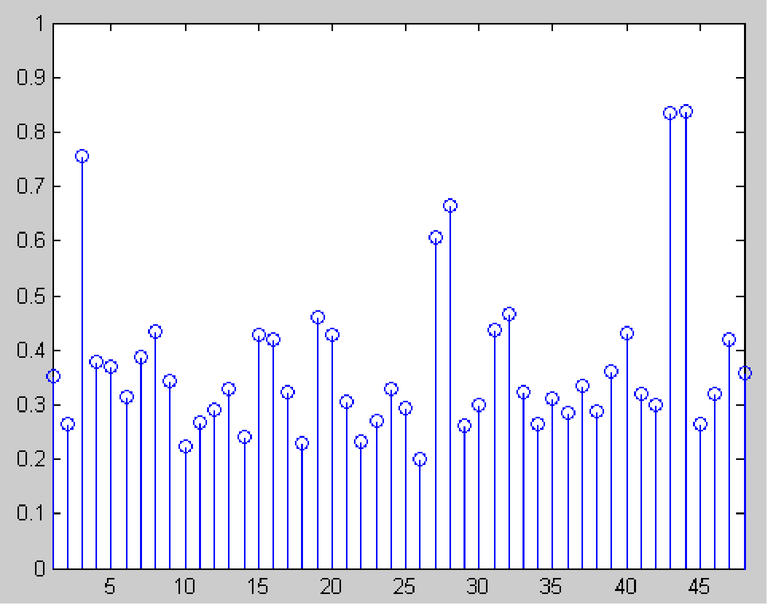


Figure 5.6: Final NCC results for Figure 5.5

5.3.1.2 Overlapping Blocks with ½ Overlap Ratio

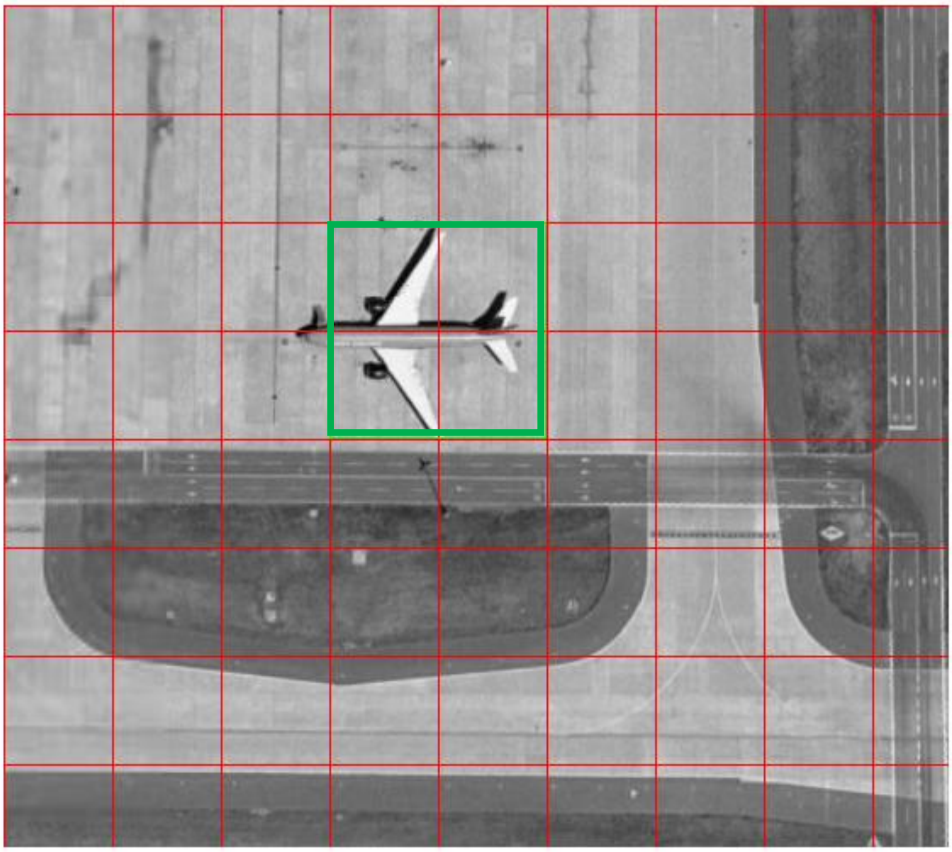


Figure 5.7: Test image for the first template with the ½ overlapping sub-images shown

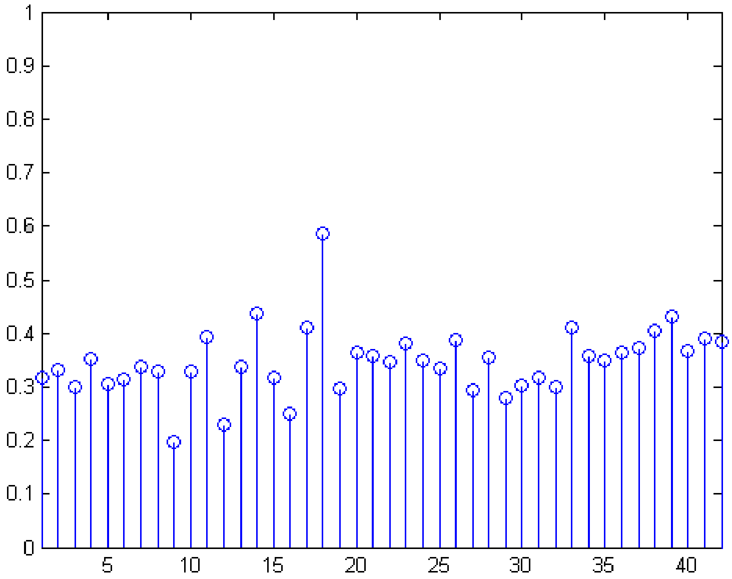


Figure 5.8: Final NCC results for Figure 5.7

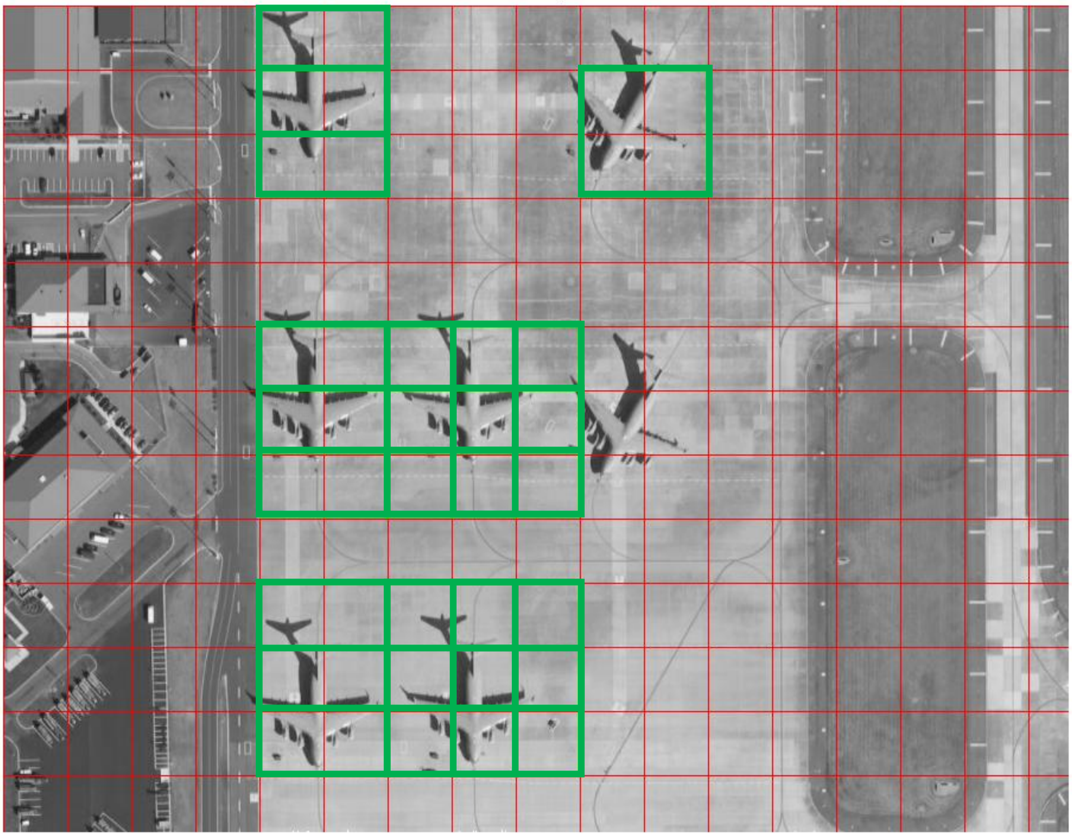


Figure 5.9: Test image for the second template with the ½ overlapping sub-images shown

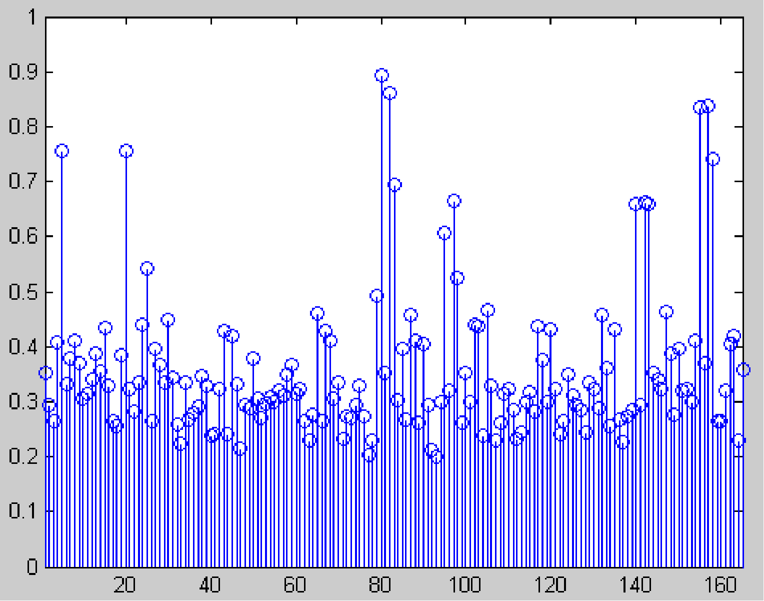


Figure 5.10: Final NCC results for Figure 5.9

5.3.1.3 Overlapping Blocks with ¾ Overlap Ratio

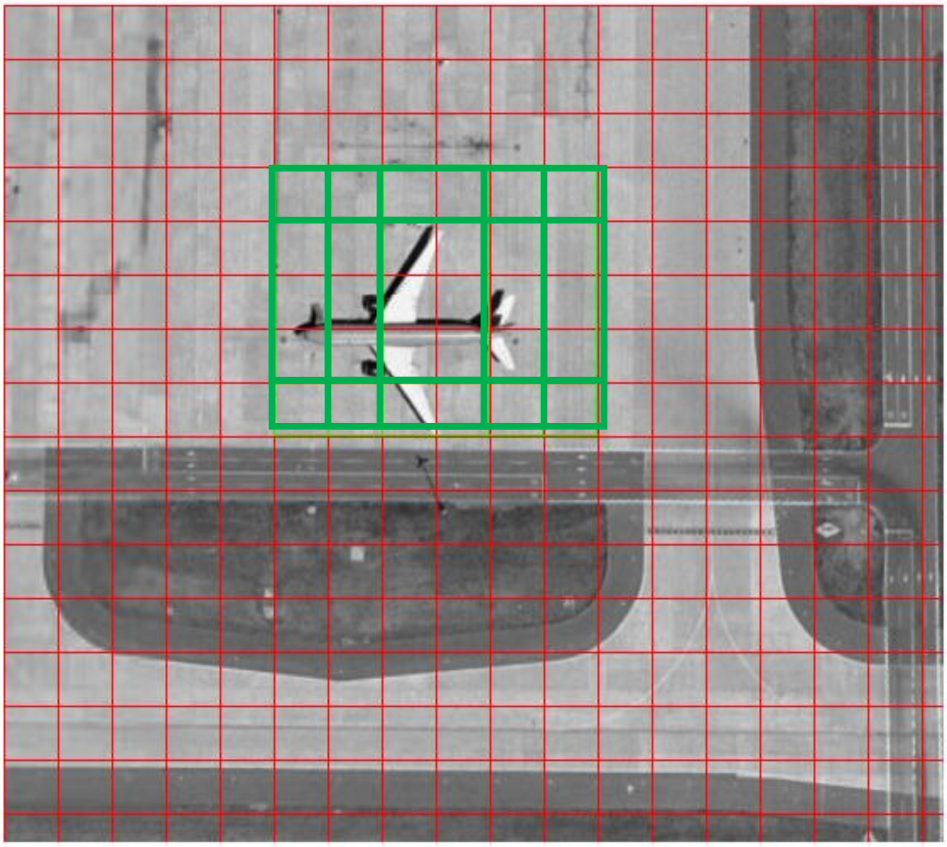


Figure 5.11: Test image for the first template with the ¾ overlapping sub-images shown

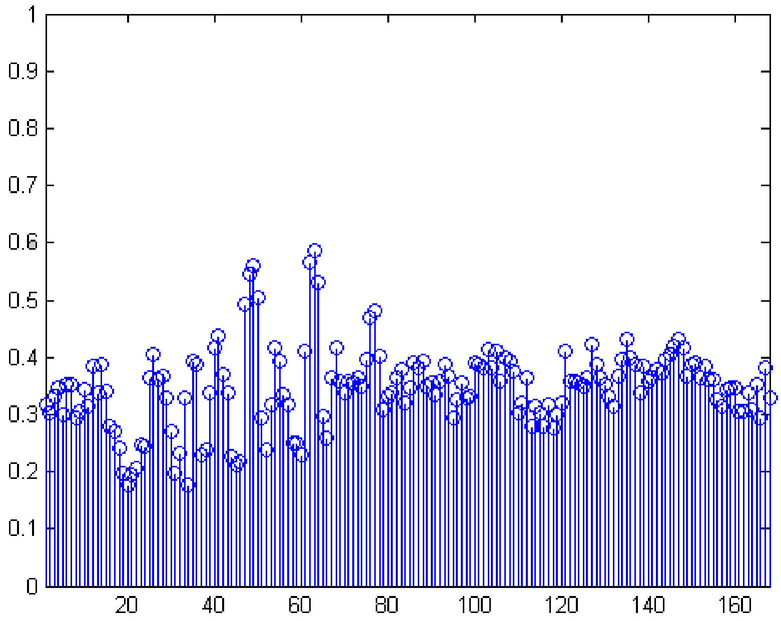


Figure 5.12: Final NCC results for Figure 5.11

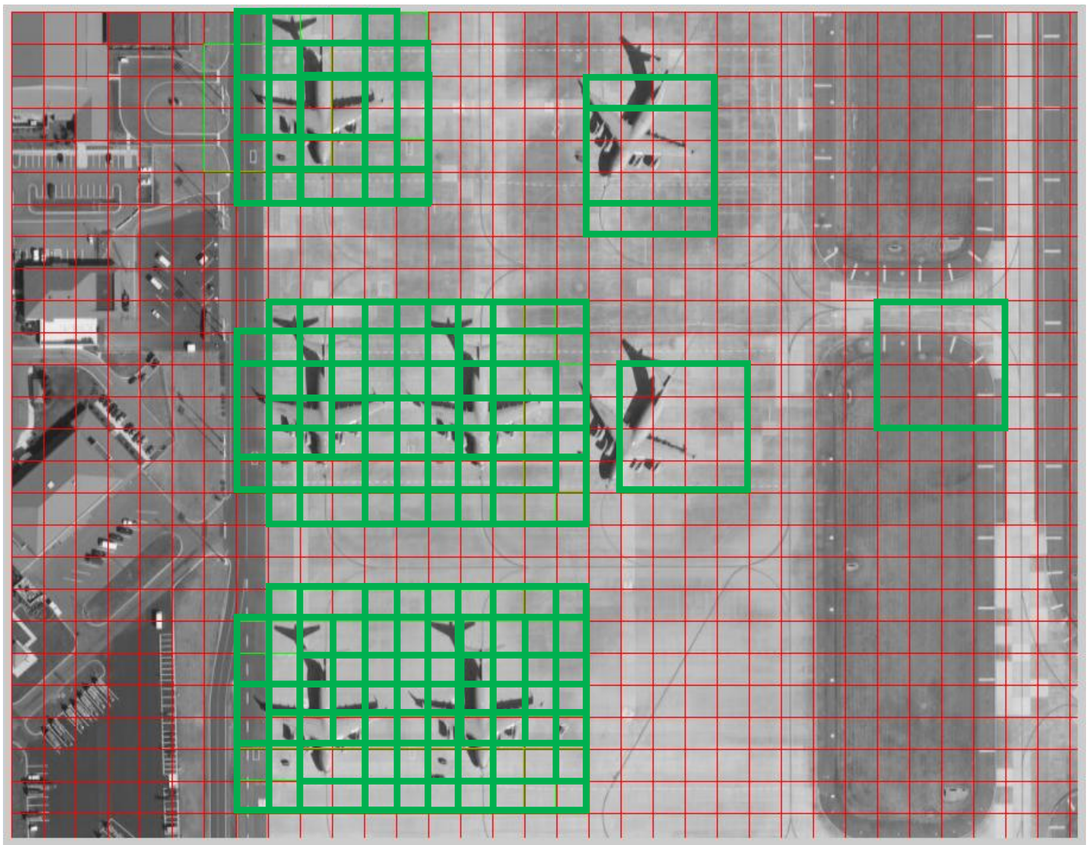


Figure 5.13: Test image for the second template with the ¾ overlapping sub-images shown

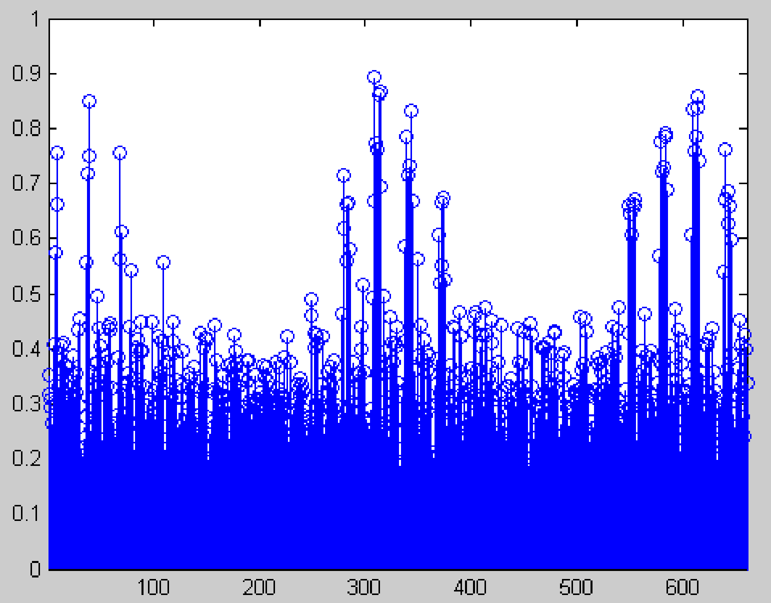


Figure 5.14: Final NCC results for Figure 5.13

5.3.1.4 Result of the Experiments on Overlapping

In the light of the experiments, it was observed that an overlap of ratio ½ is crucial to ensure that a large part of the target is in one of the sub-images and improves the results significantly. Also, in general, an overlap ratio ¾ provided with better results, even though it increases the computation time considerably. Note that the increase in computation time is directly proportional to the increase in the number of sub-images, which can be easily observed from the x-axis of the plots on NCC results for the data presented here. An overlap ratio of ¾ is adopted in the rest of this study and all the other experiments presented will be those conducted using an overlap ratio of ¾ for the sub-images.

5.3.2 Windowing Analysis

As discussed in Chapter 4, windowing can be used to decrease the effect of background and make translation appear as circular translation, and hence obtain better results. Presented in this section are the results of experiments on the effects of windowing functions on the performance.

Three similar windowing methods are implemented in this study, Gaussian windowing, Hamming windowing and Hanning windowing. These windowing functions are applied to both the templates and the sub-images. Presented below in Figure 5.15, Figure 5.16 and Figure 5.17 are these windowing functions.

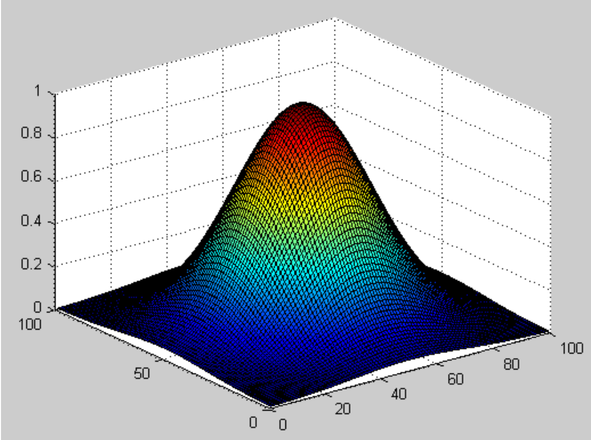


Figure 5.15: Gaussian window

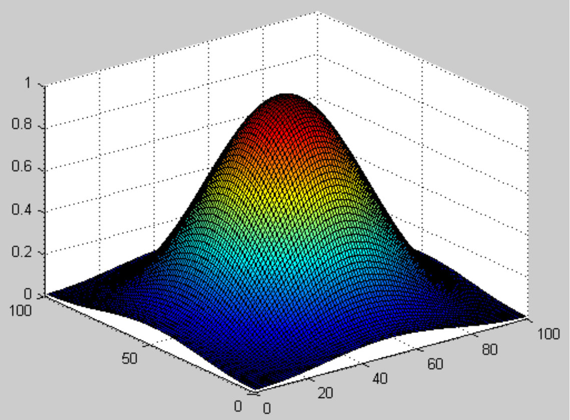


Figure 5.16: Hamming window

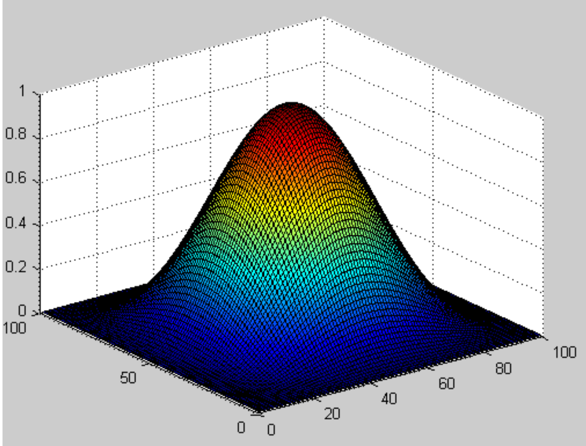


Figure 5.17: Hanning window

Two templates and two test images are presented in this section. These two tests were chosen among all the conducted experiments on windowing because they serve to enable the reader observe the results derived from the overall experiments. The templates that are used are shown below.

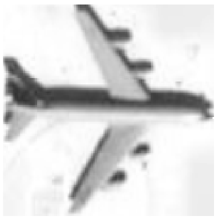


Figure 5.18: Template image for the first windowing test

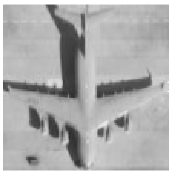


Figure 5.19: Template image for the second windowing test

5.3.2.1 Without Windowing

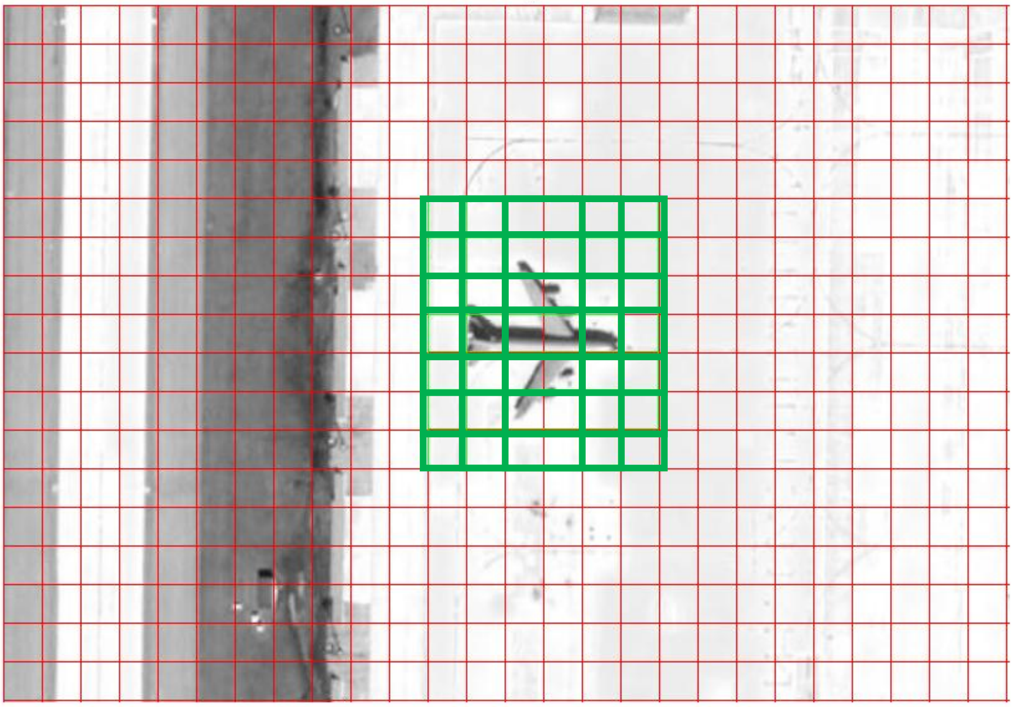


Figure 5.20: Results for the first windowing test, without windowing (threshold=0.7)

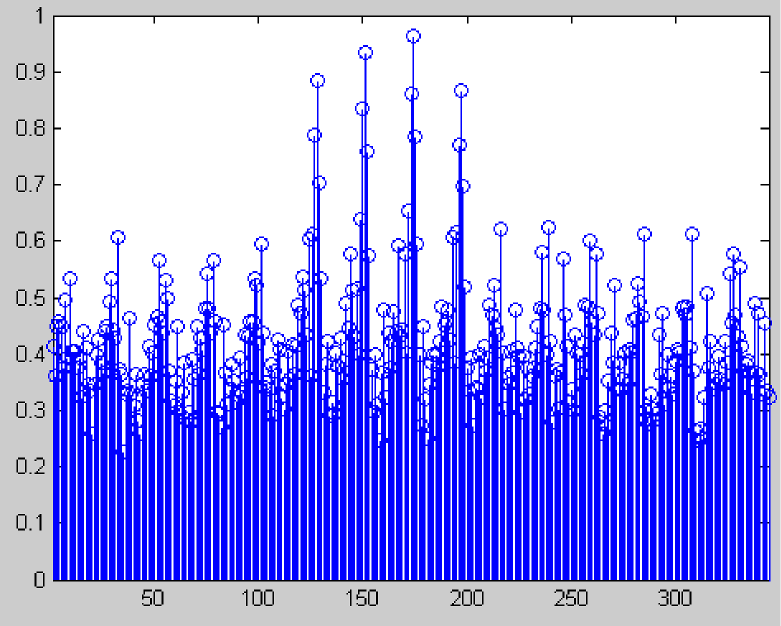


Figure 5.21: Final NCC results for the first windowing test without windowing

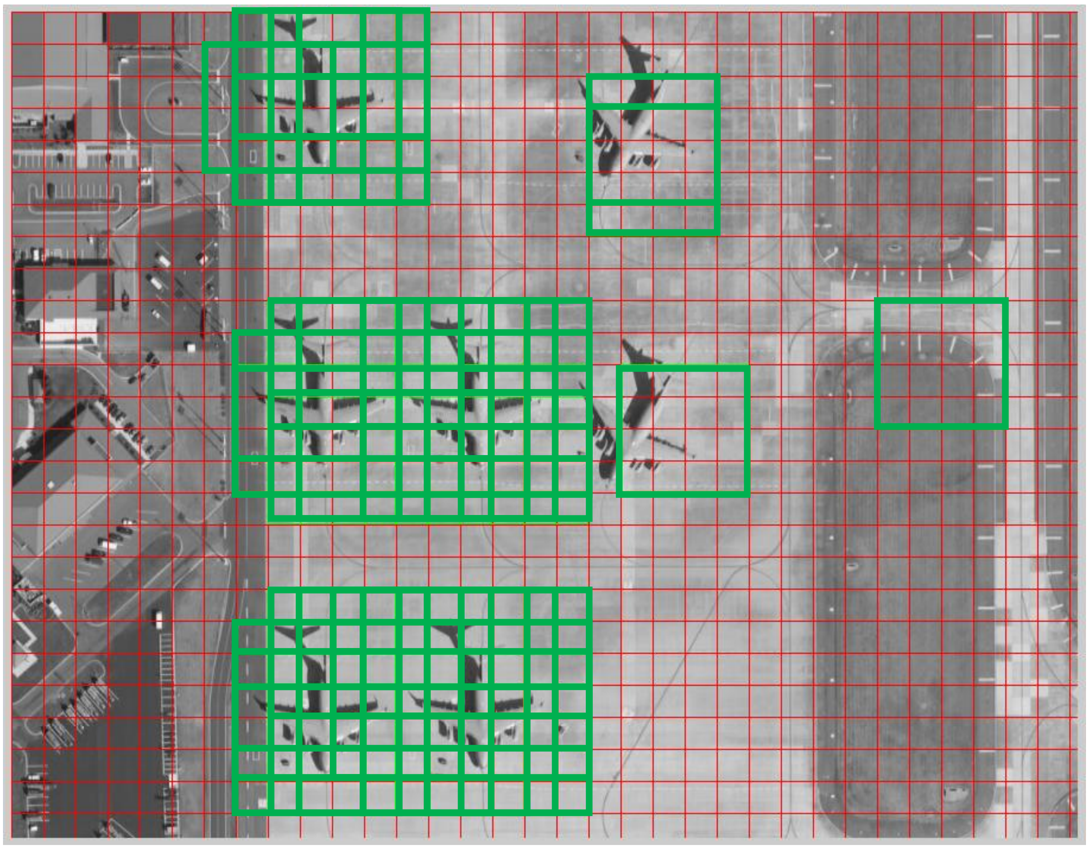


Figure 5.22: Results for the second windowing test, without windowing (threshold=0.5)

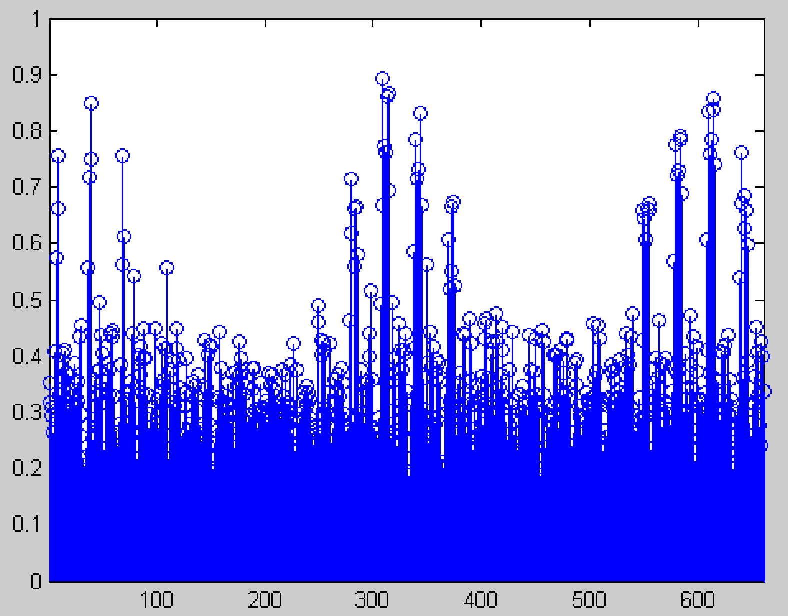


Figure 5.23: Final NCC results for the second windowing test without windowing

5.3.2.2 Gaussian Windowing

$$w(n) = \cos\left(\frac{\pi n}{N-1} - \frac{\pi}{2}\right) = \sin\left(\frac{\pi n}{N-1}\right) \qquad (70)$$

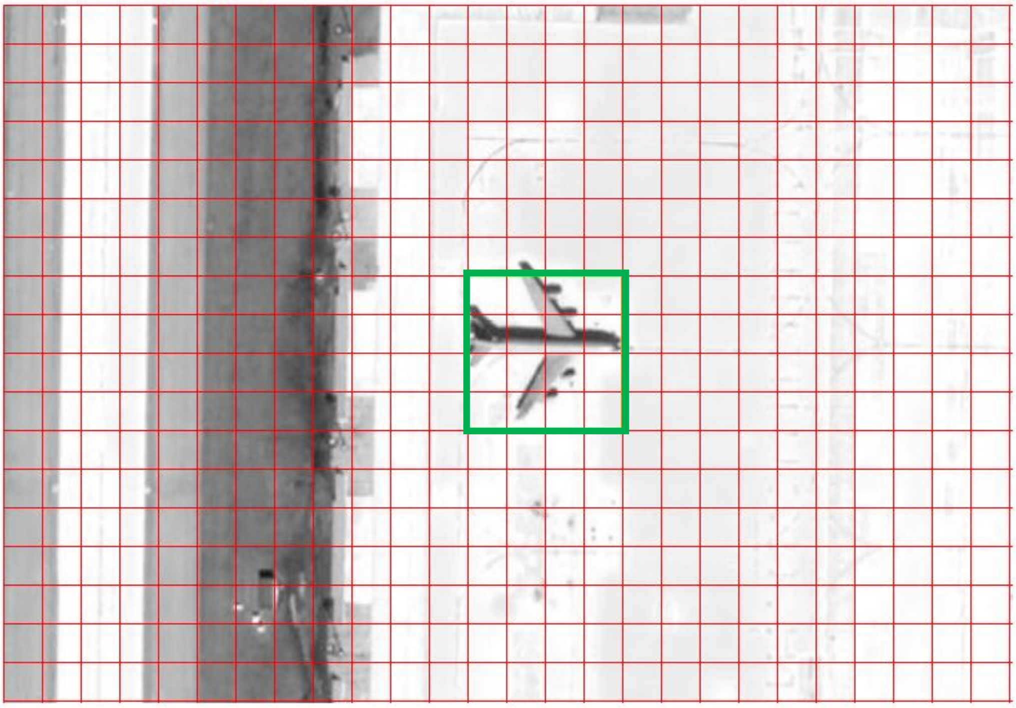


Figure 5.24: Results for the first windowing test, with Gaussian windowing (threshold=0.7)

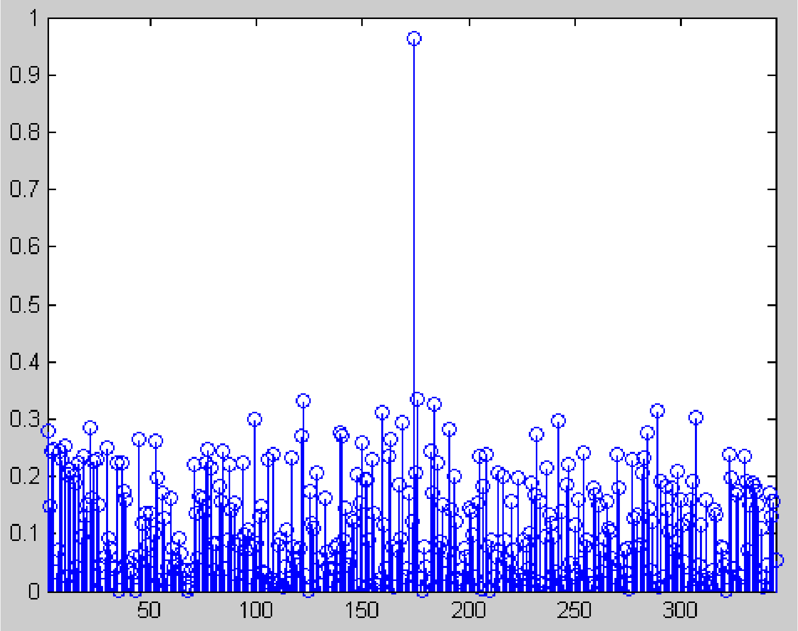


Figure 5.25: Final NCC results for the first windowing test with Gaussian windowing

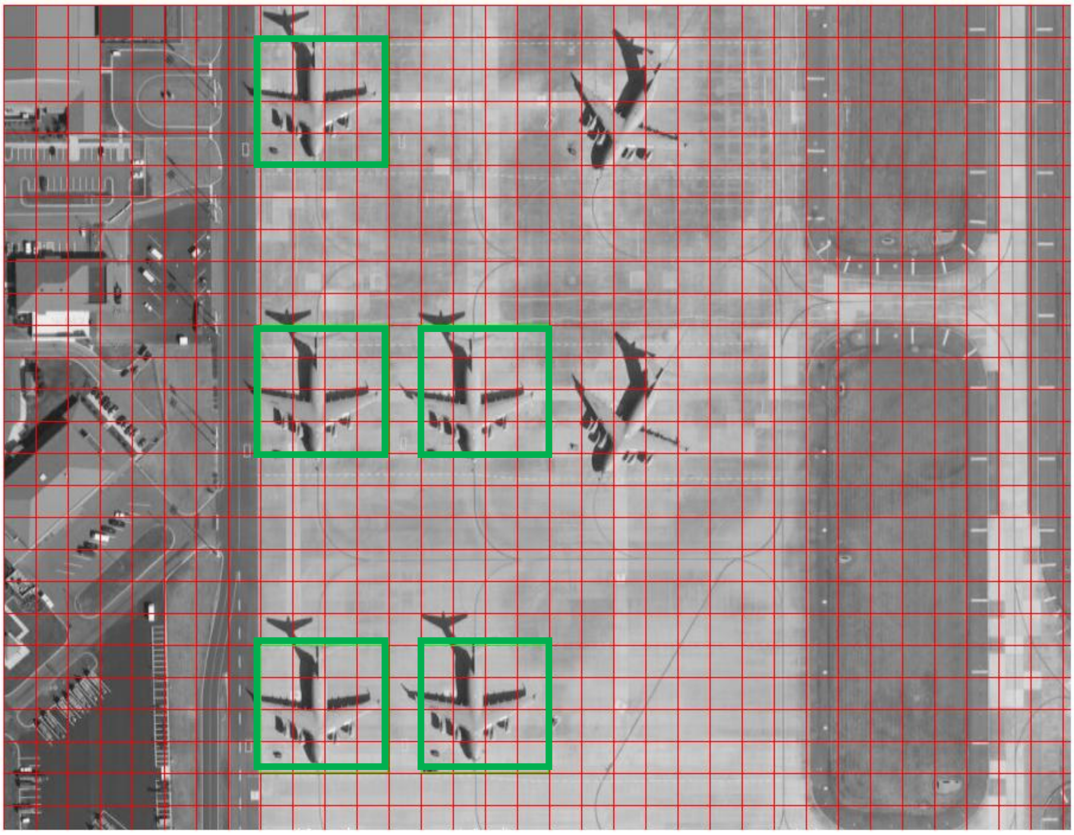


Figure 5.26: Results for the second windowing test, with Gaussian windowing (threshold=0.5)

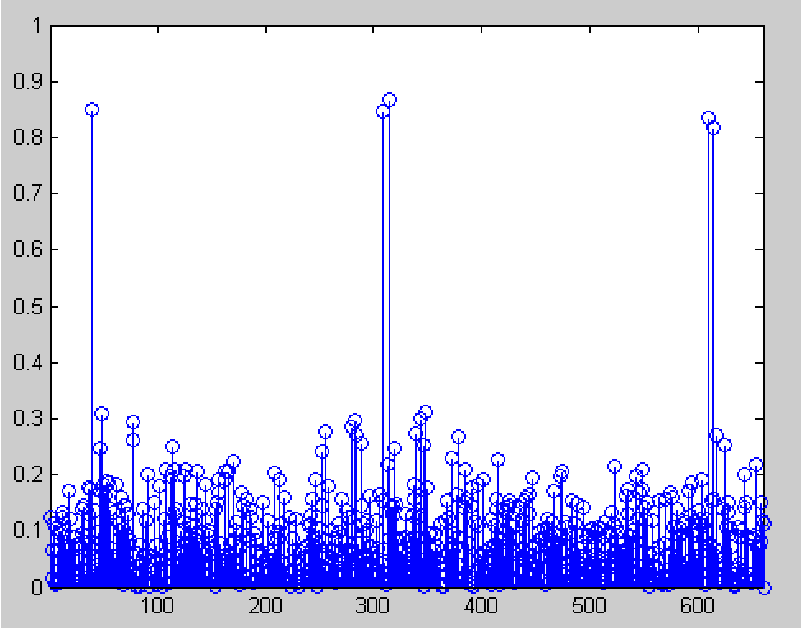


Figure 5.27: Final NCC results for the second windowing test with Gaussian windowing

5.3.2.3 Hamming Windowing

$$w(n) = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{N-1}\right) \tag{71}$$

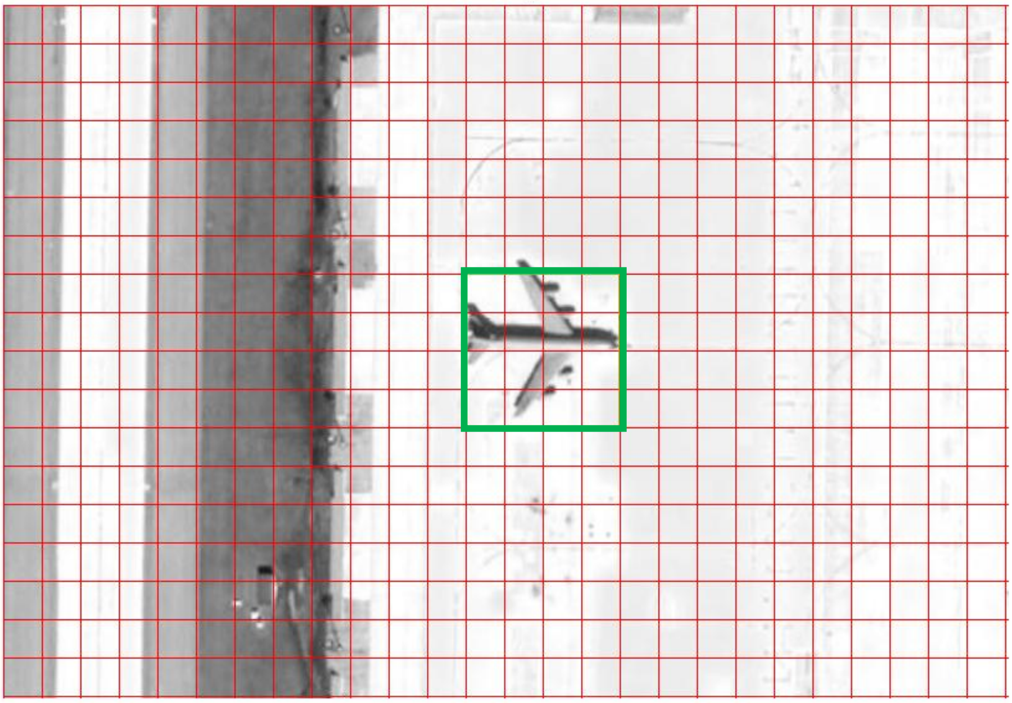


Figure 5.28: Results for the first windowing test, with Hamming windowing (threshold=0.7)

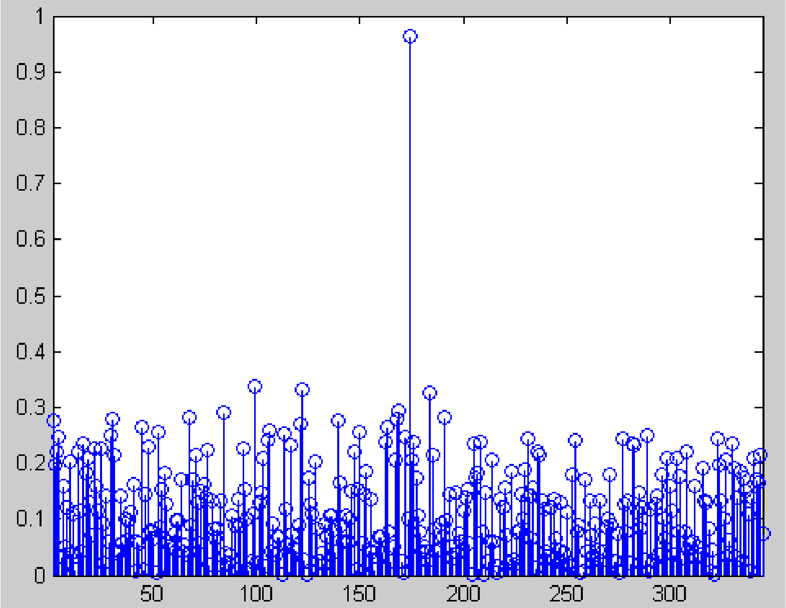


Figure 5.29: Final NCC results for the first windowing test with Hamming windowing

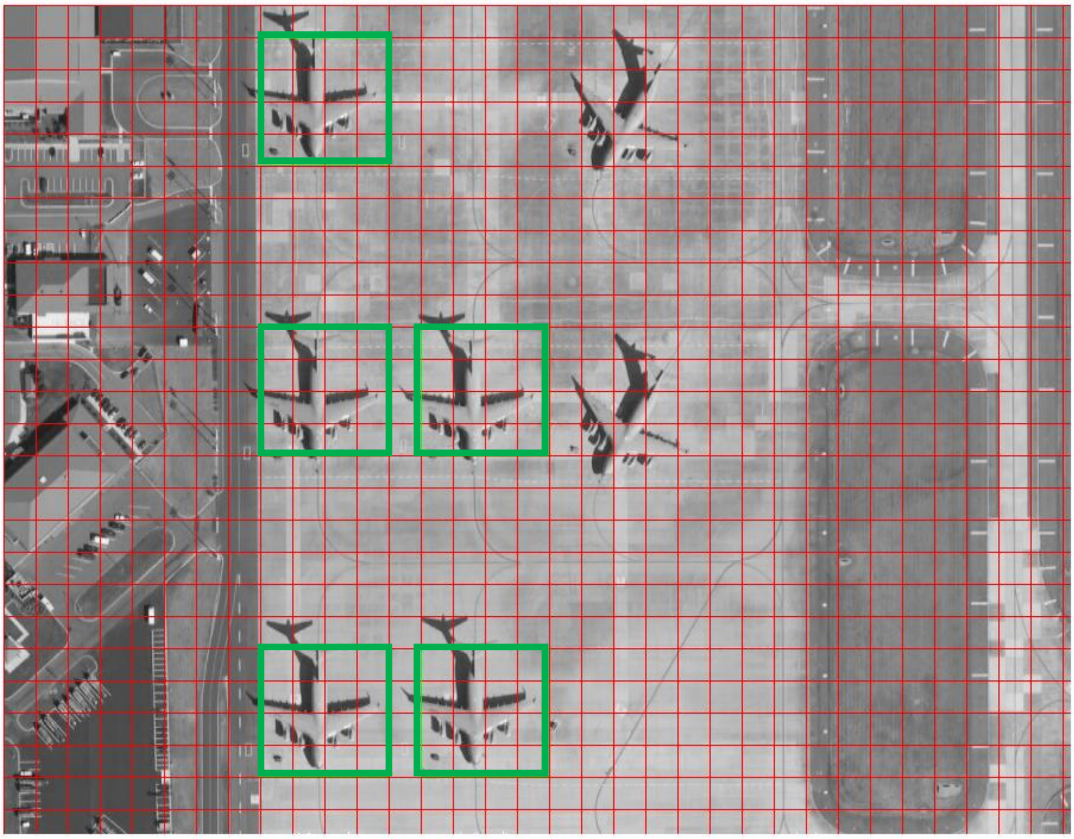


Figure 5.30: Results for the second windowing test, with Hamming windowing (threshold=0.5)

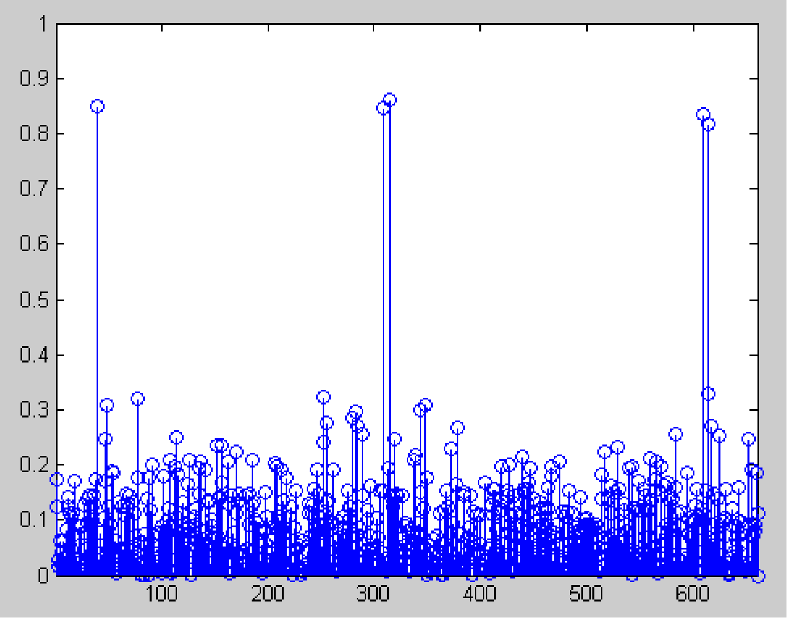


Figure 5.31: Final NCC results for the second windowing test with Hamming windowing

5.3.2.4 Hanning Windowing

$$w(n) = 0.5 \cdot \left(1 - \cos\left(\frac{2\pi n}{N - 1}\right)\right) \tag{72}$$

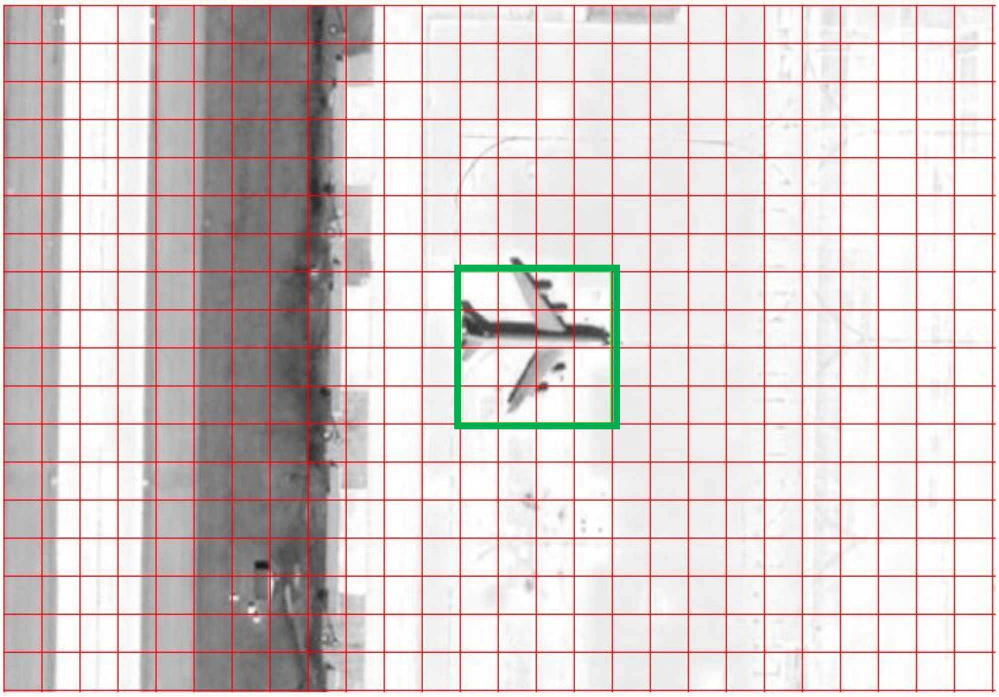


Figure 5.32: Results for the first windowing test, with Hanning windowing (threshold=0.7)

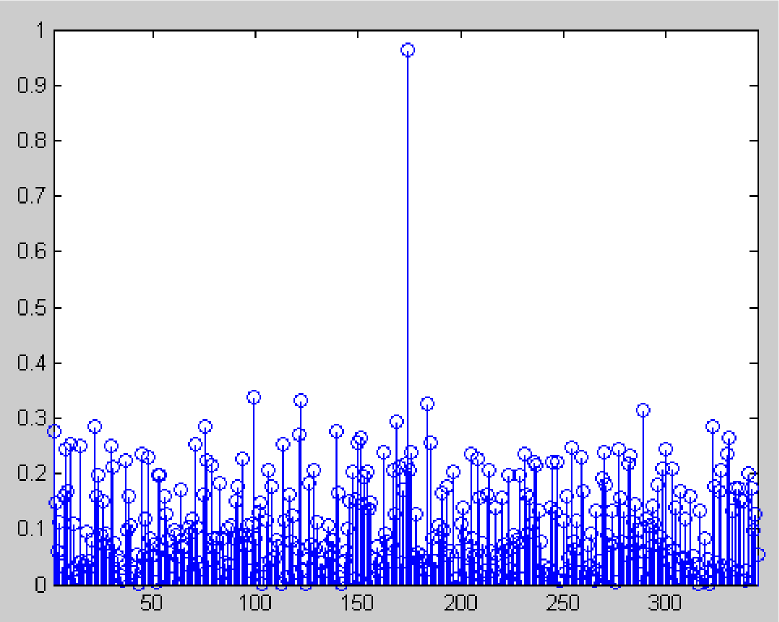


Figure 5.33: Final NCC results for the first windowing test with Hanning windowing

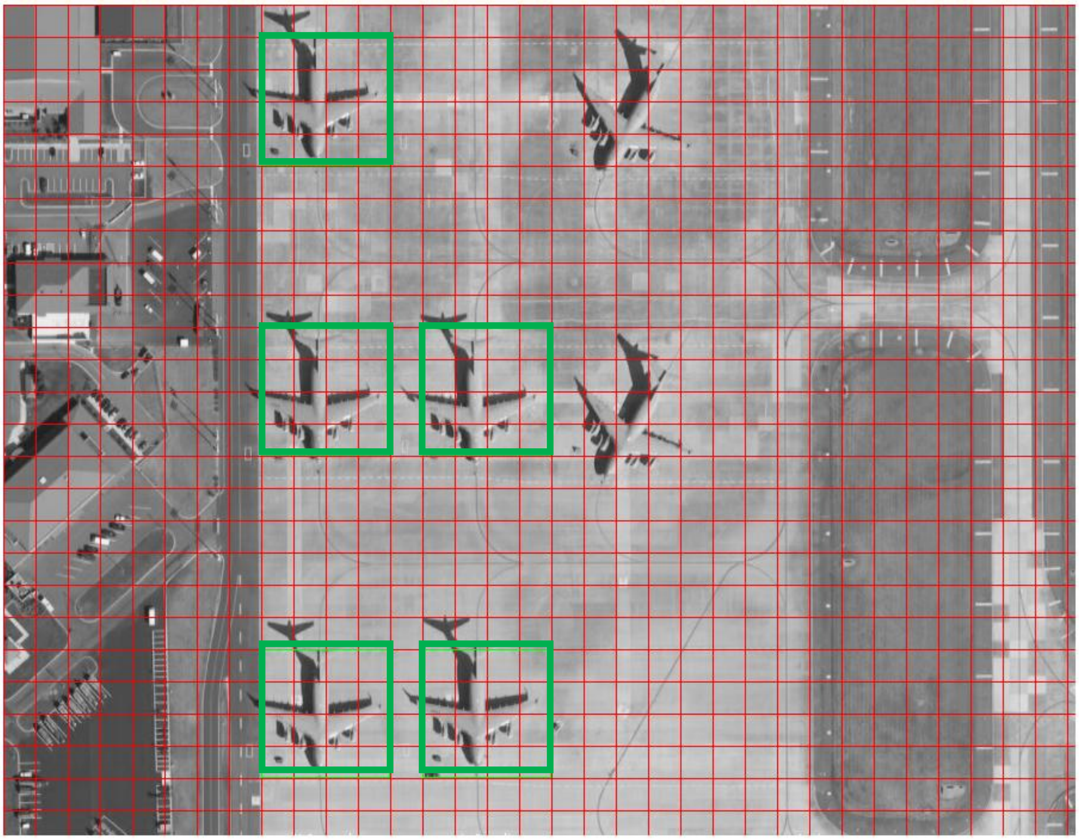


Figure 5.34: Results for the second windowing test, with Hanning windowing (threshold=0.5)

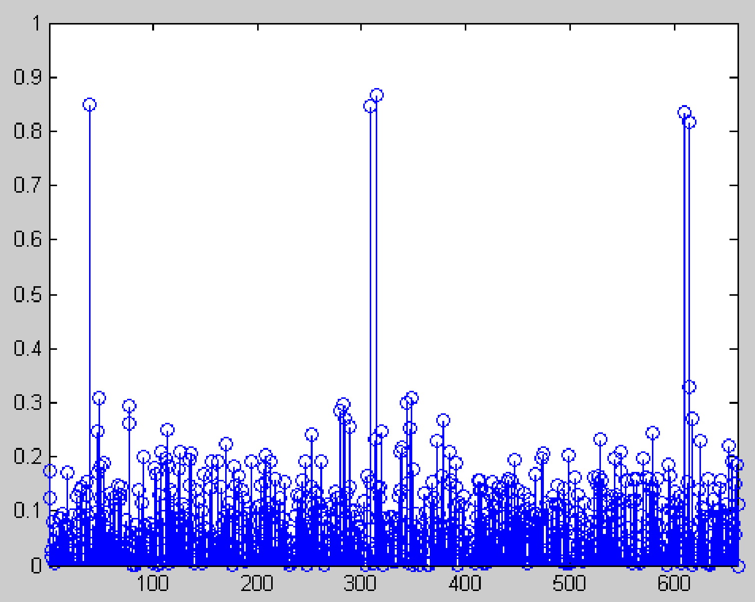


Figure 5.35: Final NCC results for the second windowing test with Hanning windowing

5.3.2.5 Result of the Experiments on Windowing

As can be seen from the two experiments presented above, the three windowing methods performed very similarly, with Hanning windowing proving very slightly better results overall. Through the experiments on windowing, it was observed that windowing can be used to increase the dynamic range of the results, and hence decrease the number of false alarms, significantly and hence also make adaptive thresholding easier. This can also be observed in the two tests presented in this section. However, windowing can increase the number of misses, as can be seen in the second experiment presented. This is caused when the target in the sub-image is not right in the center of the sub-image, but falls more into the windowed part. Although an overlap ratio of ¾ decreases this possibility, it does not nullify it, and hence the number of misses can somewhat increase with windowing. However, the gain is greater than the lost. Hanning windowing is adopted for the rest of this study.

5.3.3 Using Artificial Background for the Template

As explained before in Section 4.5.1.1, additional data in the form of background affects the results significantly and is the main difference of this study from the somewhat similar studies that can be found in the literature. Using artificial background for the template is one of the first things that come to mind to solve the background drawback. In this section, an example for the tests on using artificial background for the template is presented and the results of the overall tests are explained with conclusions drawn.

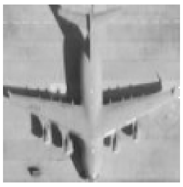


Figure 5.36: Template image with natural background

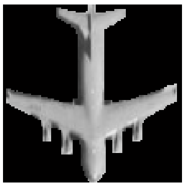


Figure 5.37: Template image with artificial background

5.3.3.1 Template with Natural Background

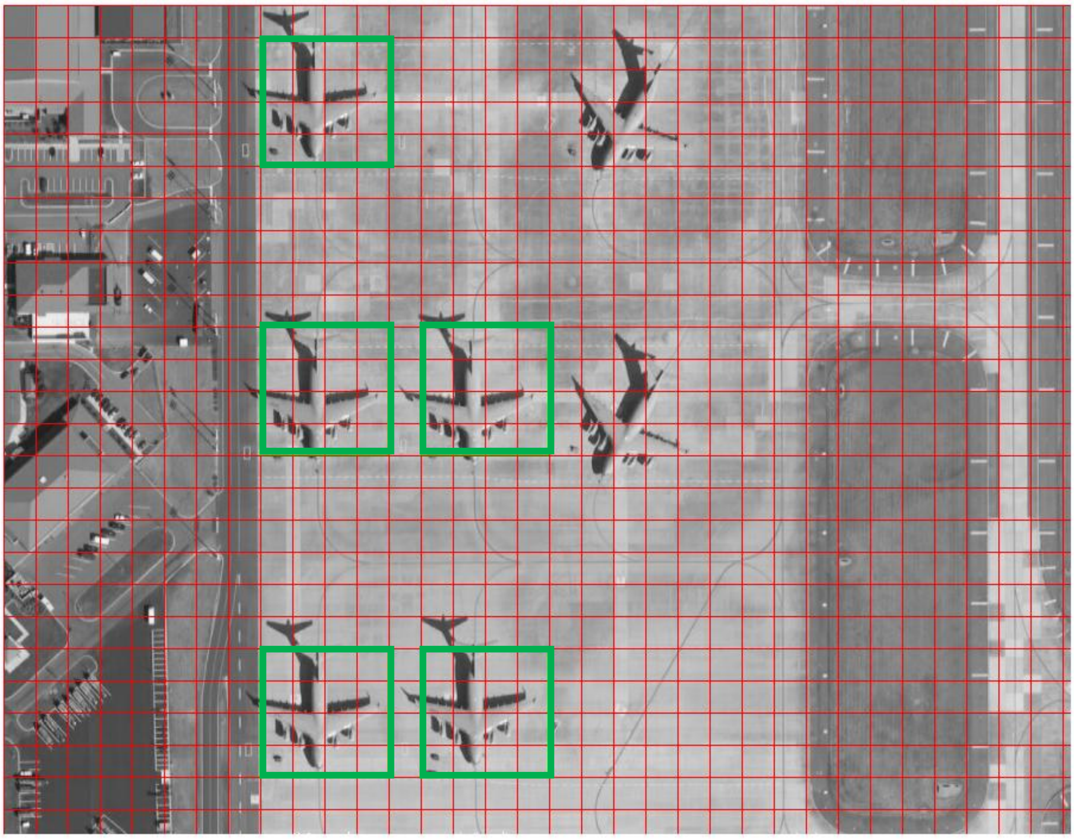


Figure 5.38: Results for the template image with natural background (using Hanning windowing)

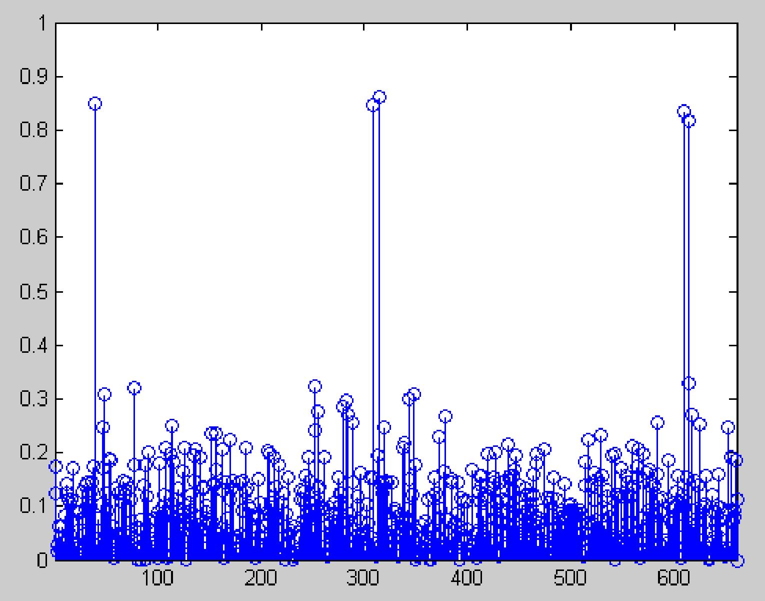


Figure 5.39: Final NCC results for the template image with natural background (using Hanning windowing)

5.3.3.2 Template with Artificial Background

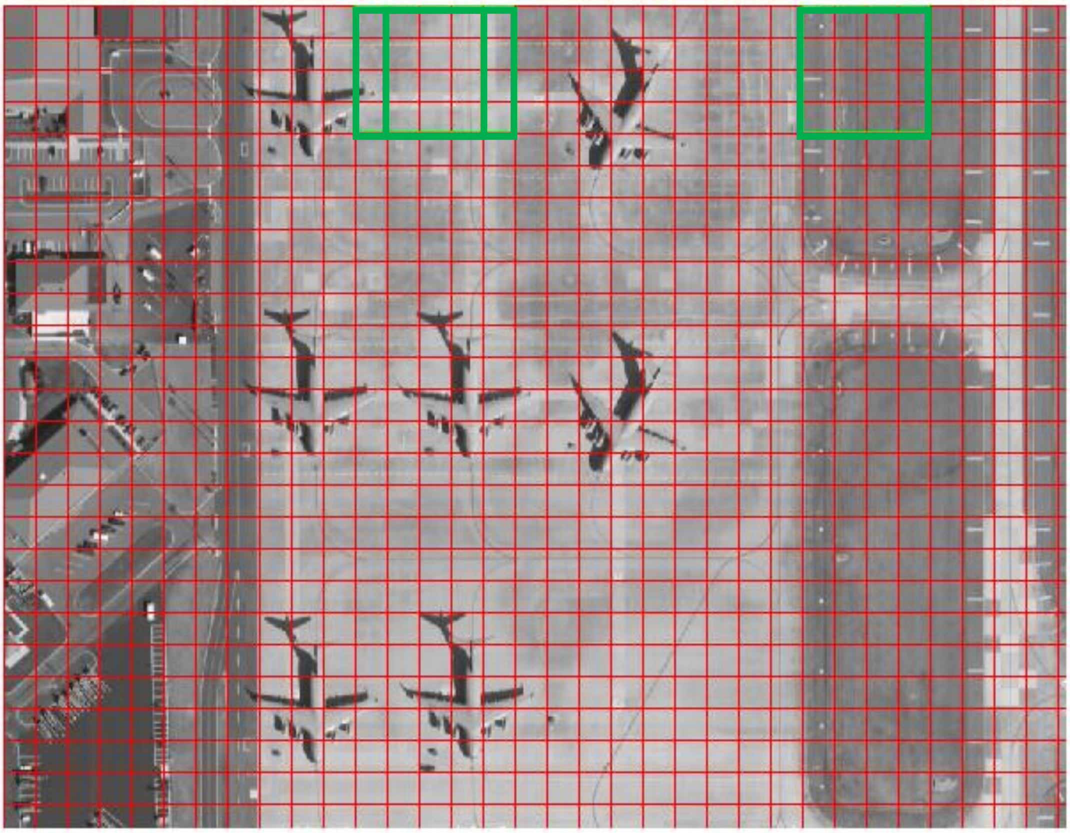


Figure 5.40: Results for the template image with artificial background (using Hanning windowing)

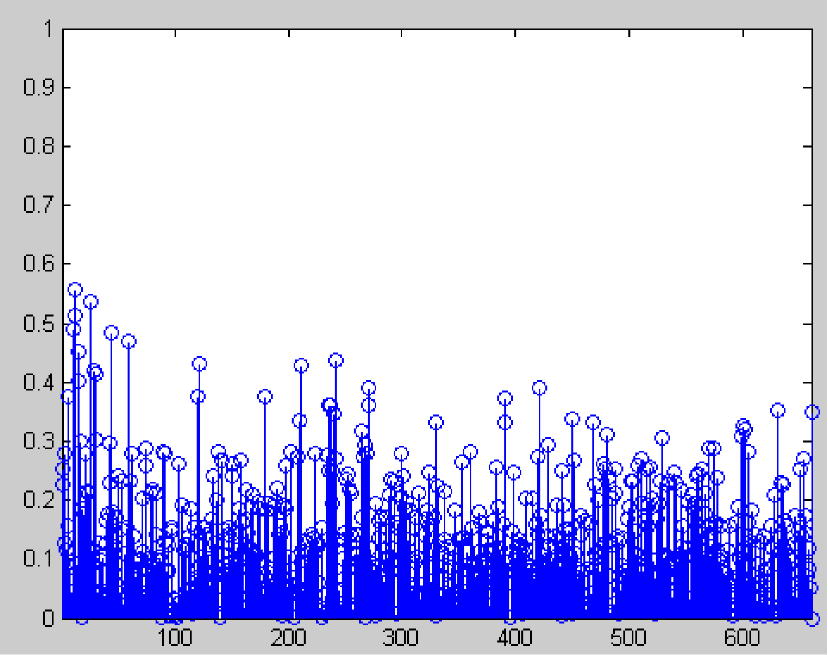


Figure 5.41: Final NCC results for the template image with artificial background (using Hanning windowing)

5.3.3.3 Result of the Experiments on using Artificial Background for the Template

As can also be observed from the example presented above, using artificial background for the template does not improve the results. On the contrary, the results are generally degraded by using artificial background. The reason for this is that although what we wish to match is the template, the background also takes up a big part of the template image and actually, both are tried to be matched in the test images. This is also the reason that targets that are on much different backgrounds than the targets in the template image are hard to be detected and recognized.

5.3.4 Slice Analysis

As discussed in Chapter 4, the choice of which bispectrum slice to use, i.e., the bispectrum slice constant, affects the results. This effect is indistinct in some images, while important in others. The straightforward solution to this drawback is to use a multiple number of slices, which result in more computation time, but better results. However, first, the fact that the results vary according to slice choice should be proven. Given below are a template image and a test image containing 13 targets of the same type as in the template image. The targets in the test image are aligned in various directions, which ensure that RST invariance can be better observed. After the template image and the test image are presented, the resulting normalized cross-correlation results are presented for different bispectrum slice constants, followed by the results for using multiple slices.

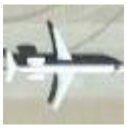


Figure 5.42: Template image for the slice analysis test

5.3.4.1 Single Slices

Given below are the results for single slices, starting from the first slice, up to the tenth.

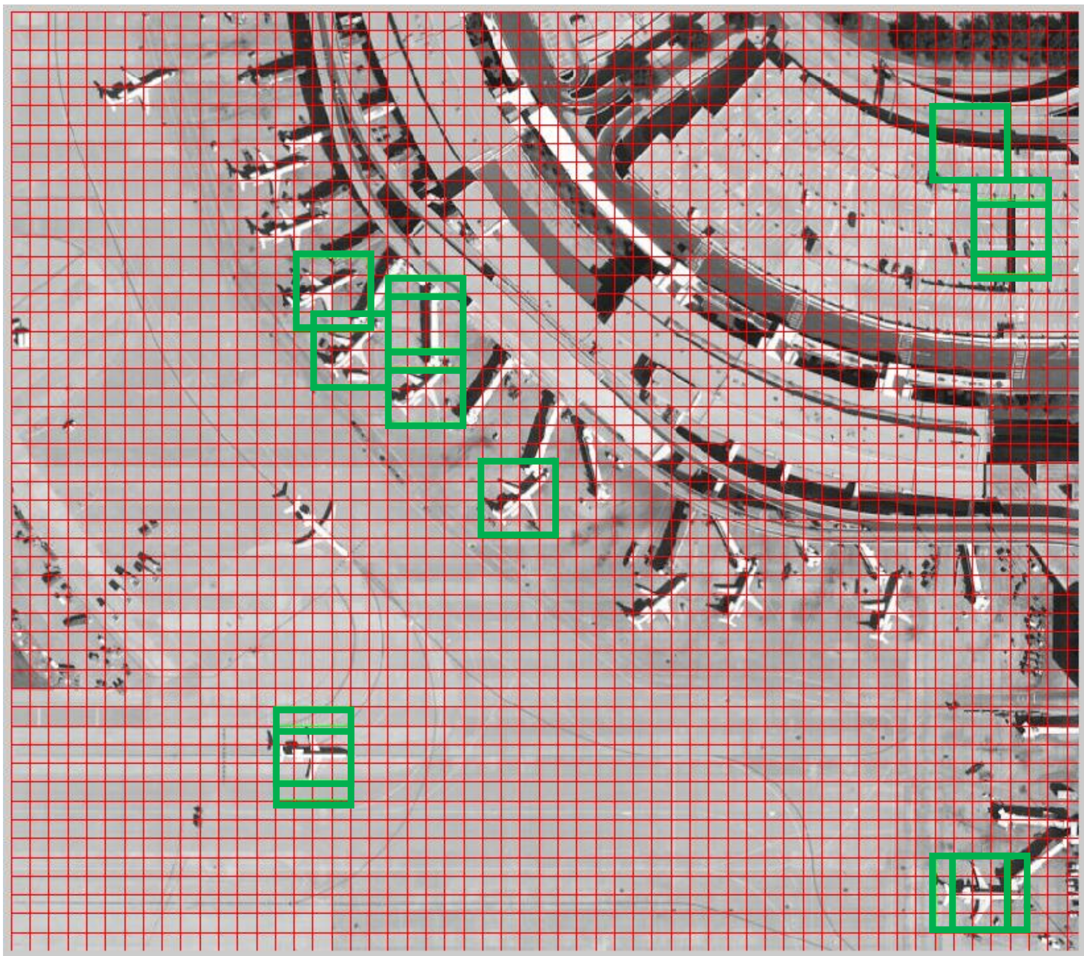


Figure 5.43: Results using bispectrum slice constant = 1, threshold=0.4

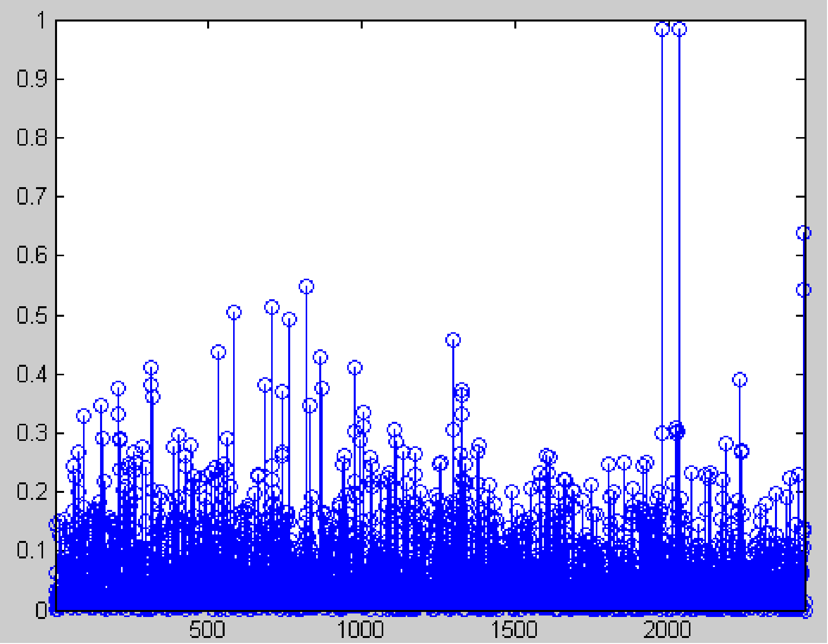


Figure 5.44: NCC results using bispectrum slice constant = 1

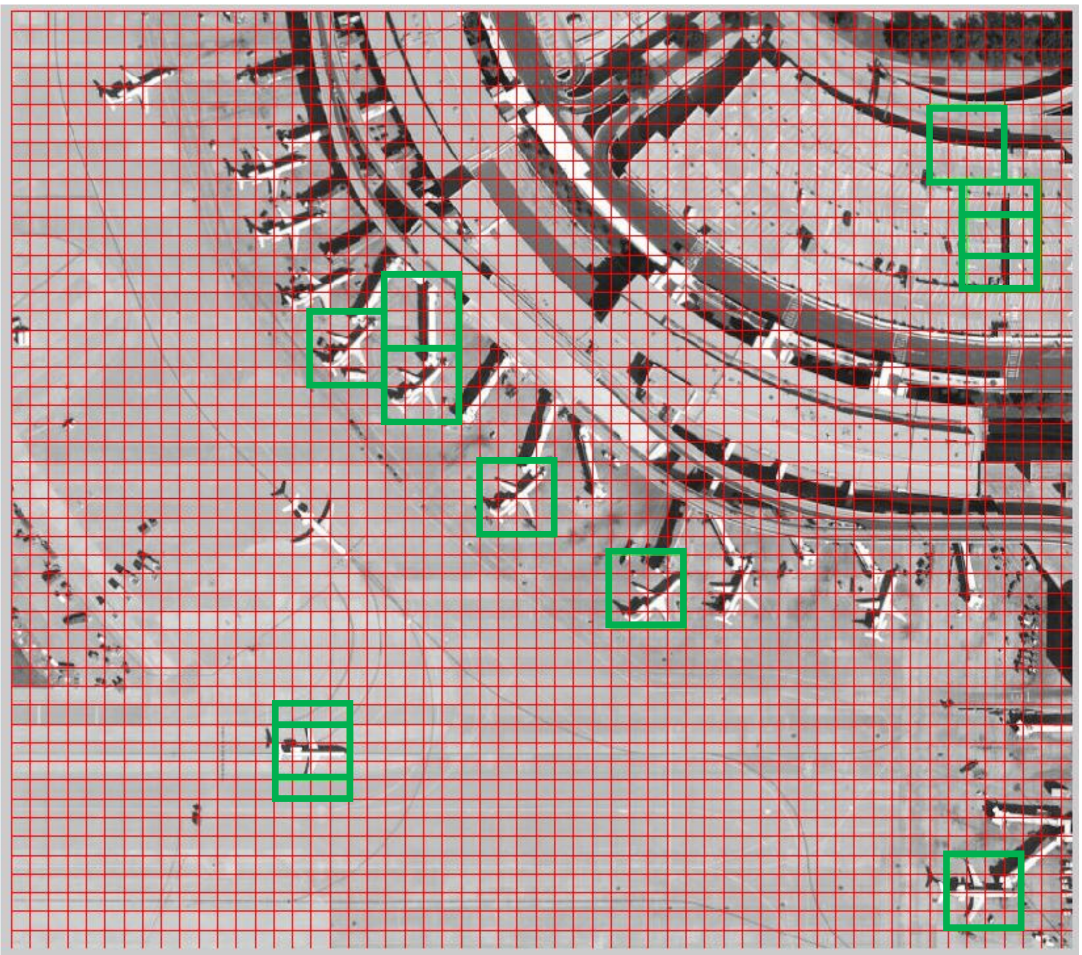


Figure 5.45: Results using bispectrum slice constant = 2, threshold=0.4

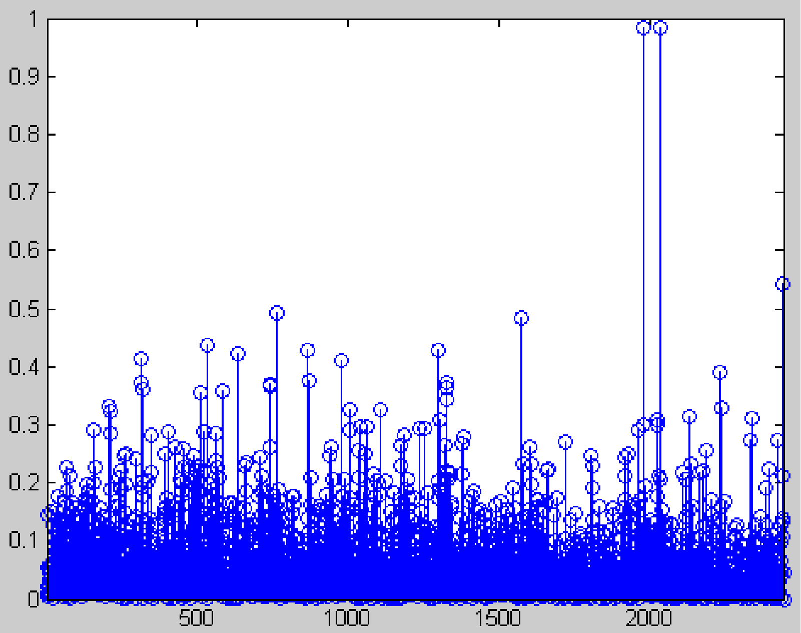


Figure 5.46: NCC results using bispectrum slice constant = 2

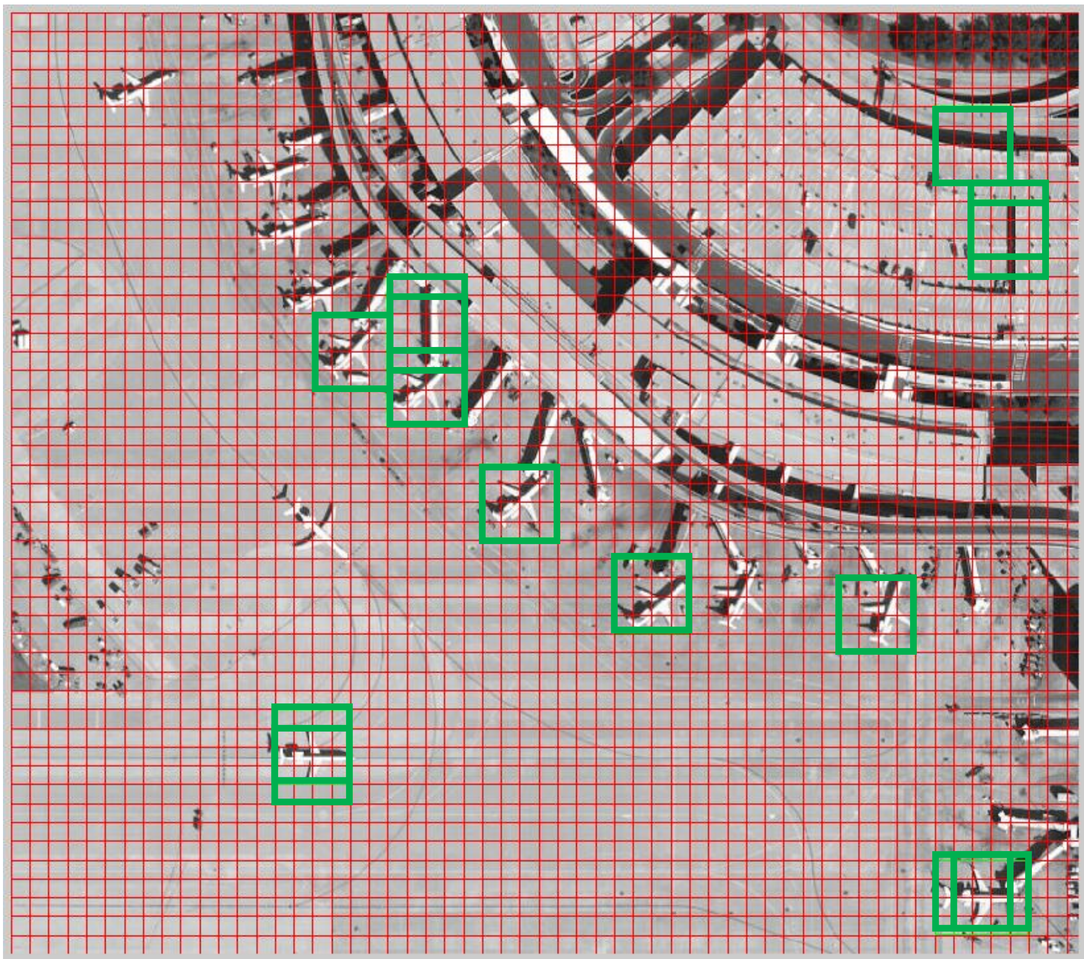


Figure 5.47: Results using bispectrum slice constant = 3, threshold=0.4

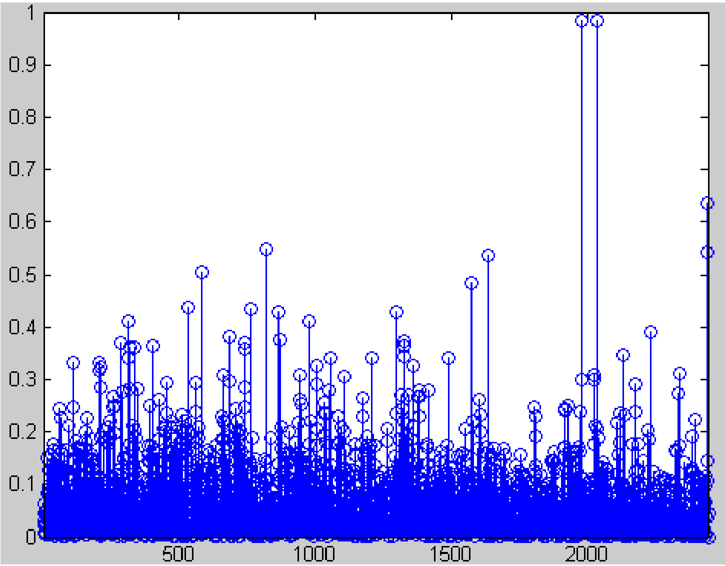


Figure 5.48: NCC results using bispectrum slice constant = 3

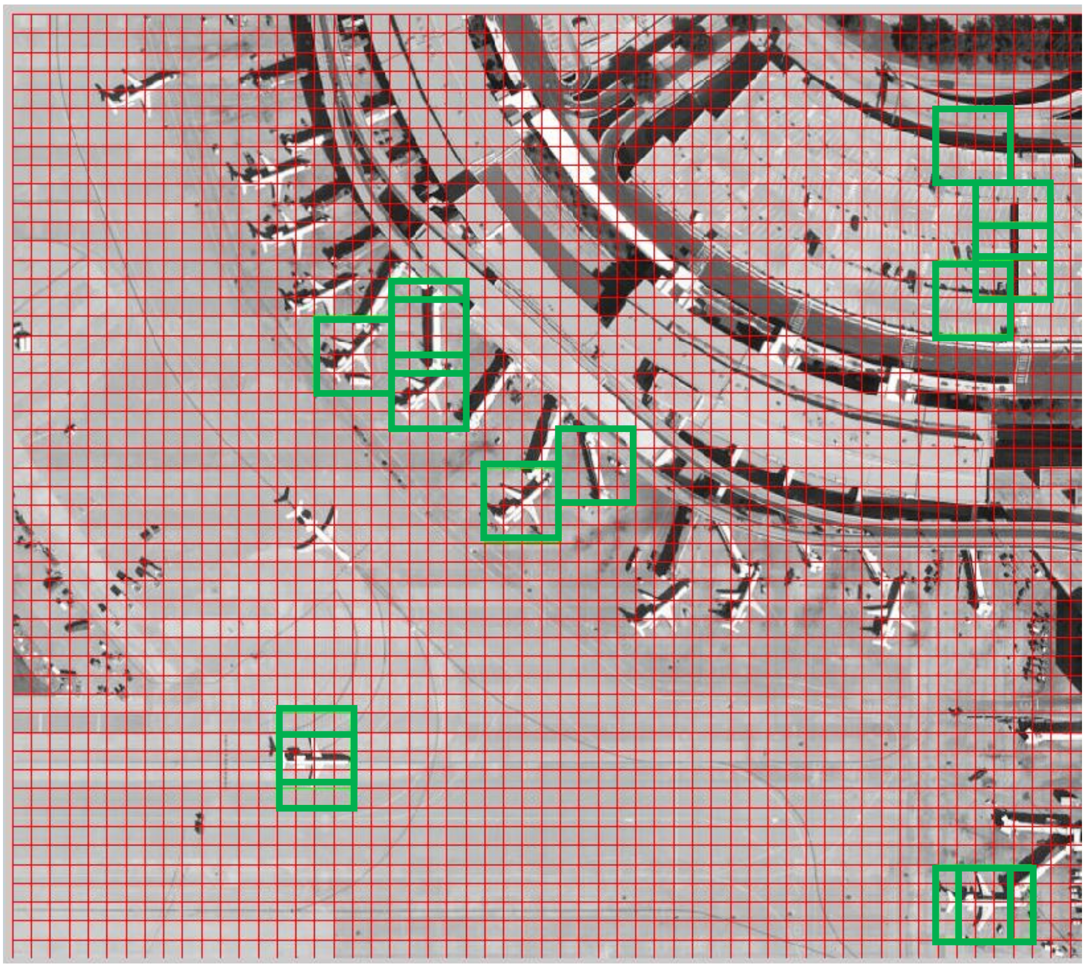


Figure 5.49: Results using bispectrum slice constant = 4, threshold=0.4

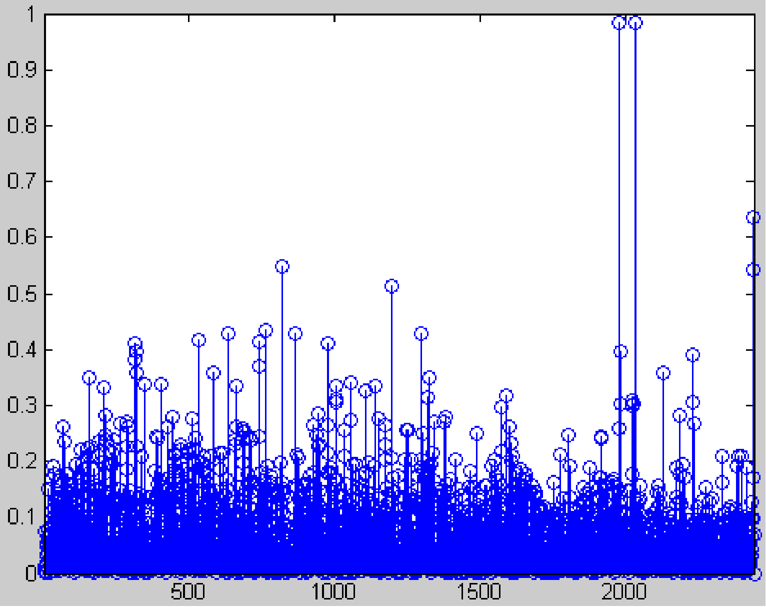


Figure 5.50: NCC results using bispectrum slice constant = 4

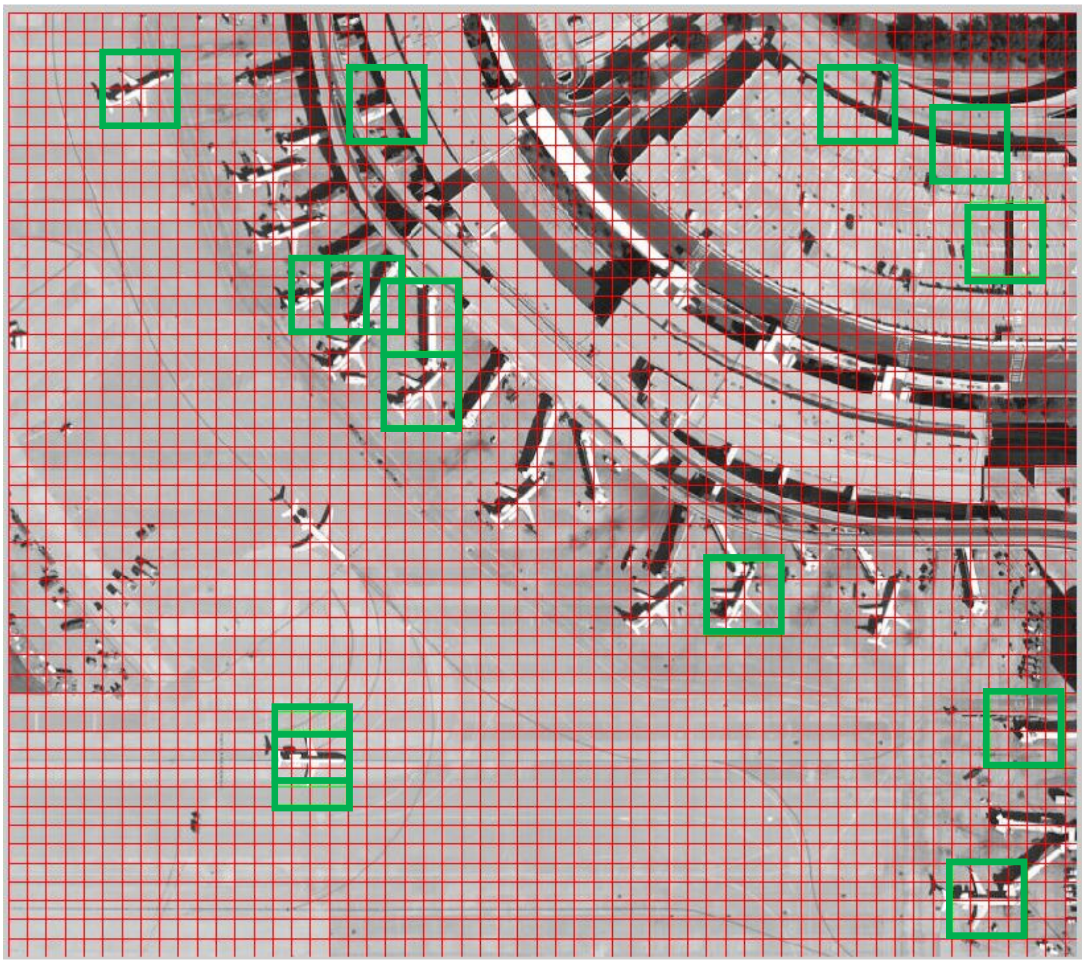


Figure 5.51: Results using bispectrum slice constant = 5, threshold=0.4

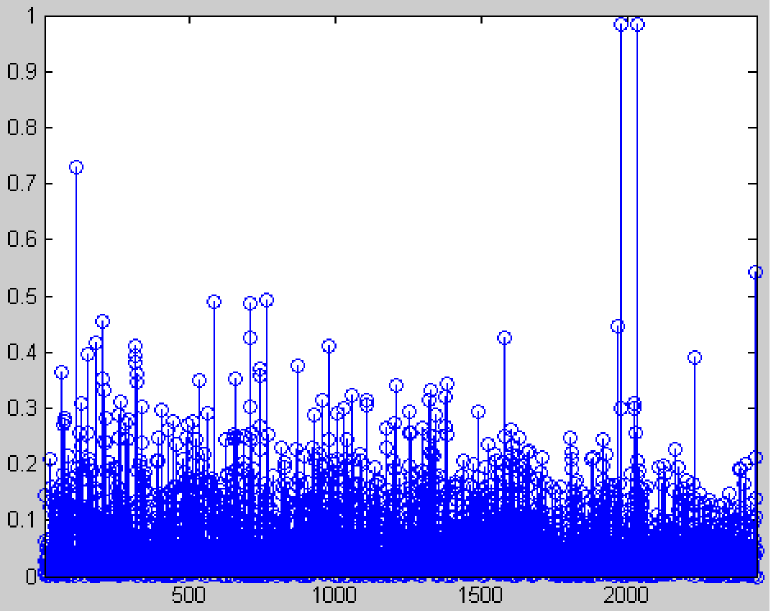


Figure 5.52: NCC results using bispectrum slice constant = 5

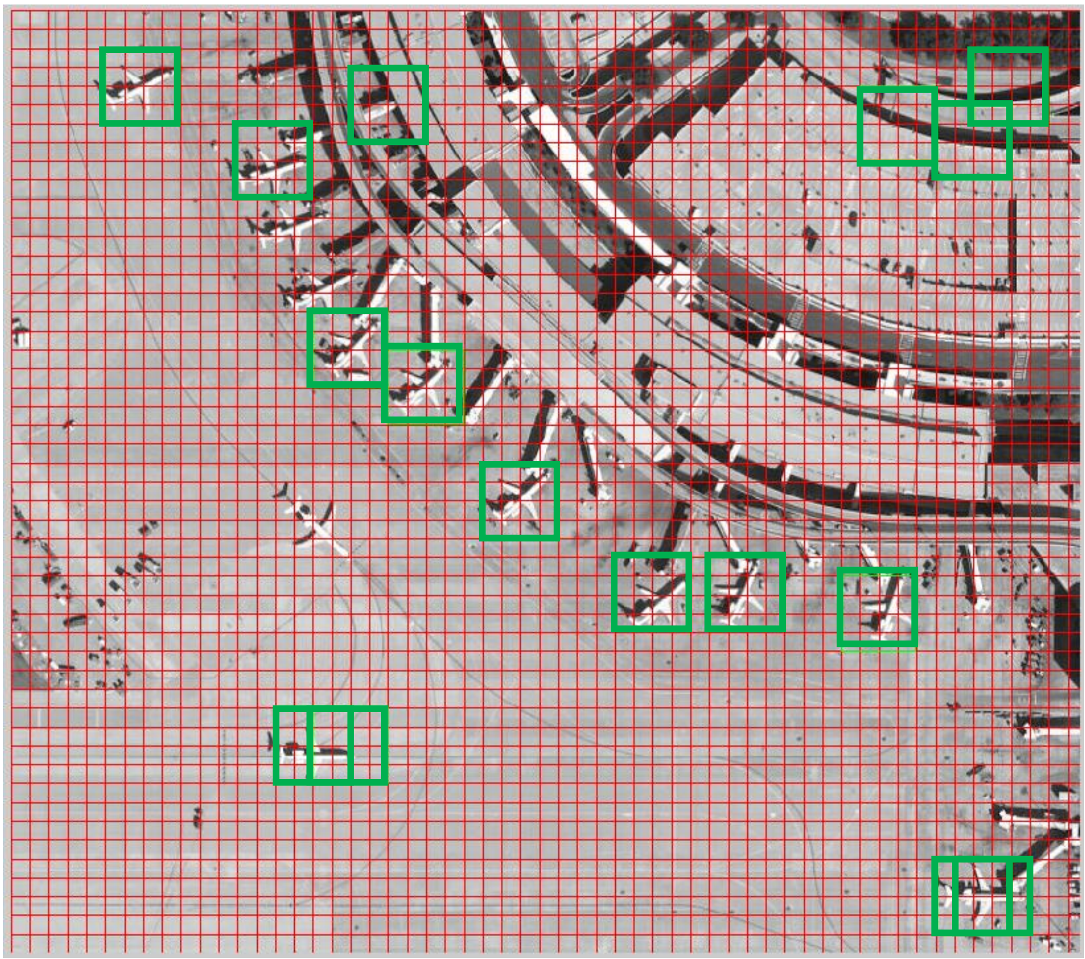


Figure 5.53: Results using bispectrum slice constant = 6, threshold=0.4

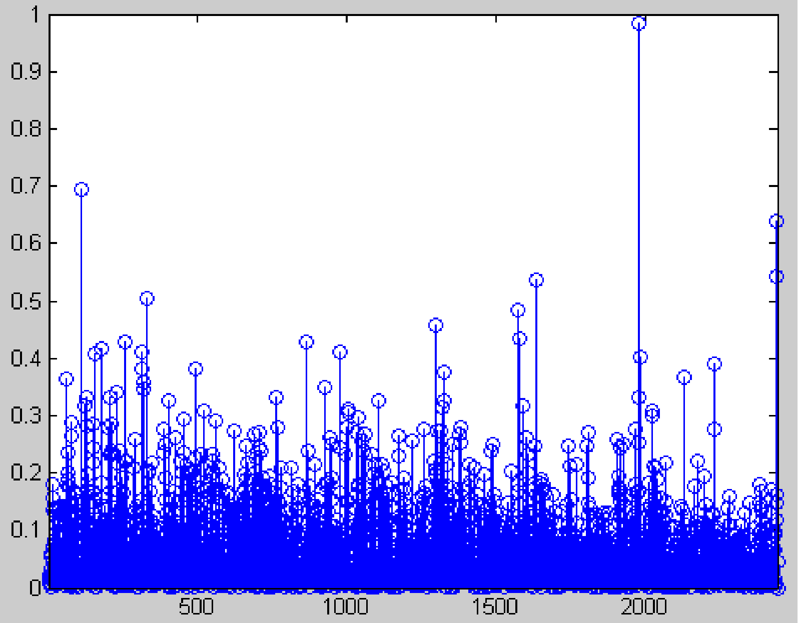


Figure 5.54: NCC results using bispectrum slice constant = 6

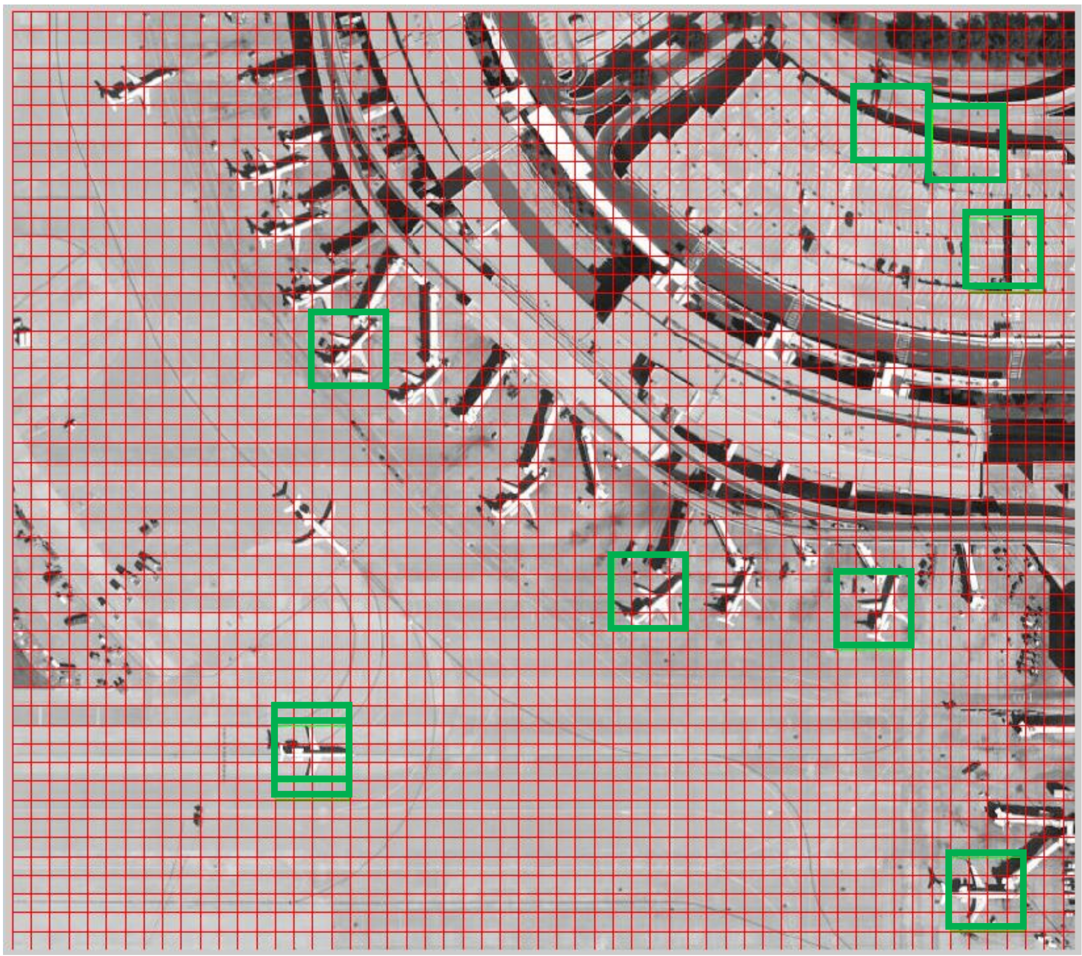


Figure 5.55: Results using bispectrum slice constant = 7, threshold=0.4

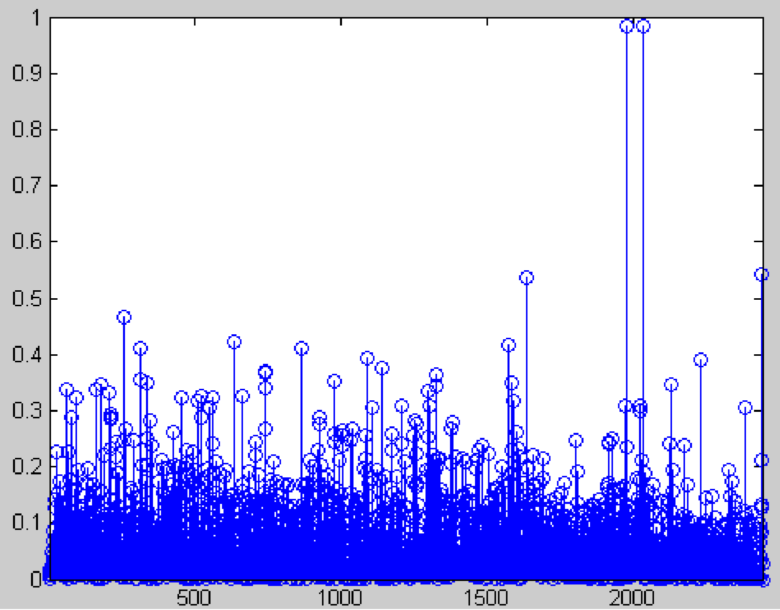


Figure 5.56: NCC results using bispectrum slice constant = 7

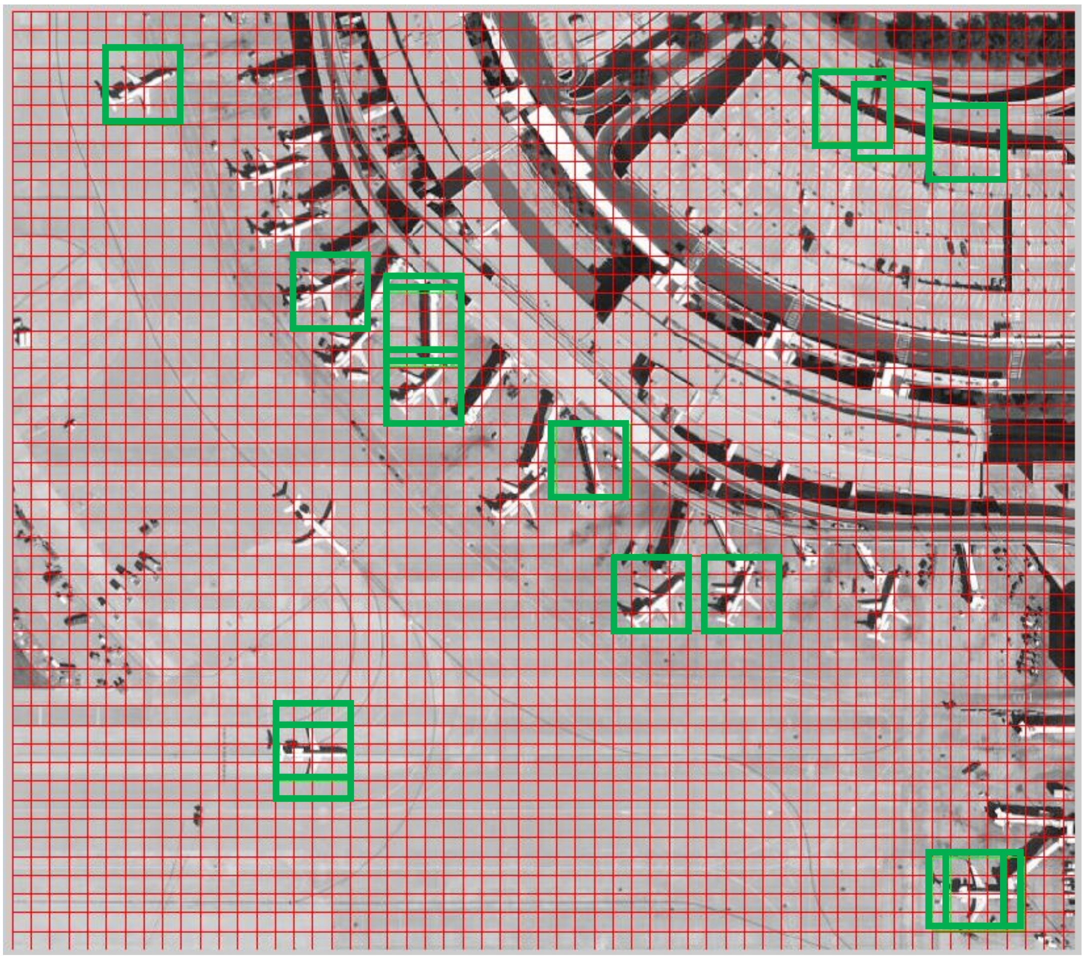


Figure 5.57: Results using bispectrum slice constant = 8, threshold=0.4

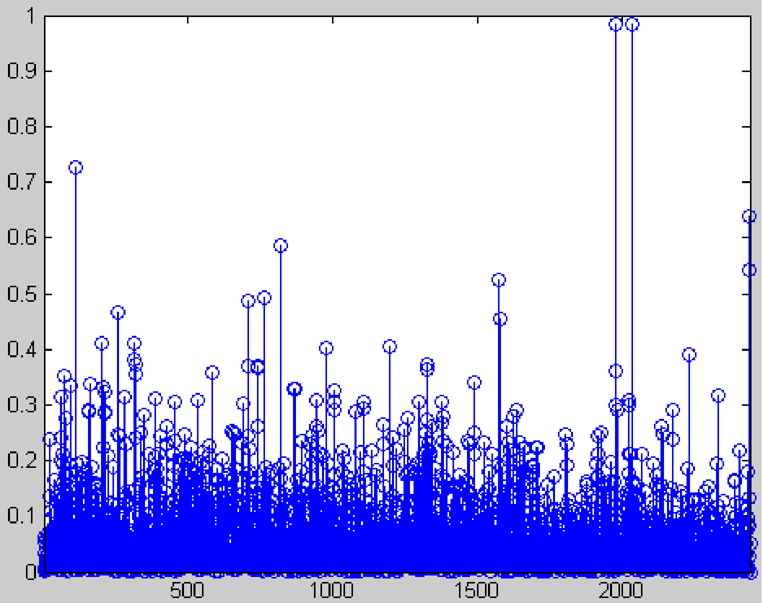


Figure 5.58: NCC results using bispectrum slice constant = 8

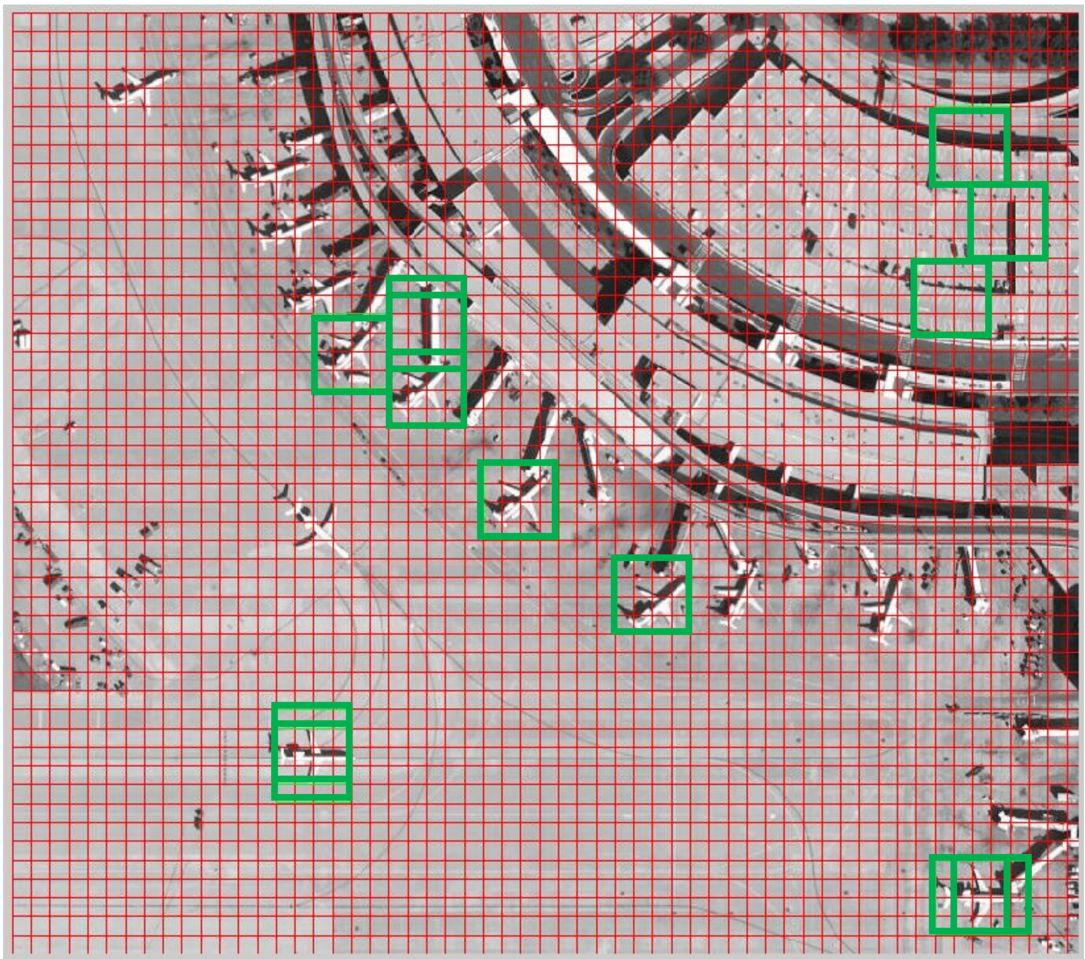


Figure 5.59: Results using bispectrum slice constant = 9, threshold=0.4

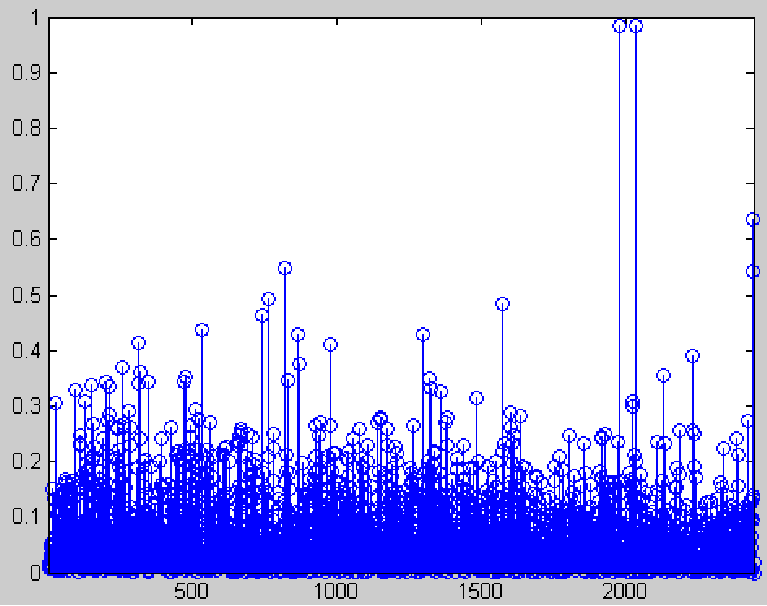


Figure 5.60: NCC results using bispectrum slice constant = 9

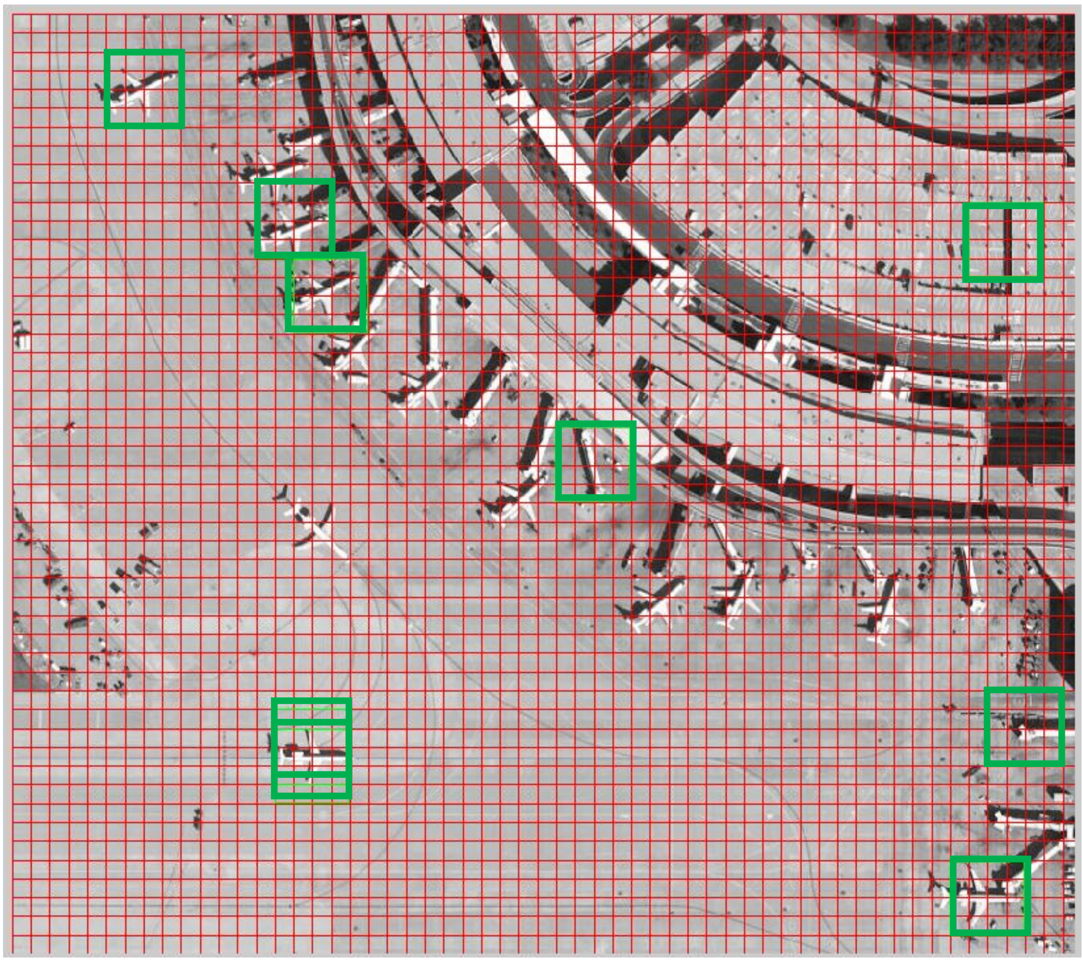


Figure 5.61: Results using bispectrum slice constant = 10, threshold=0.4

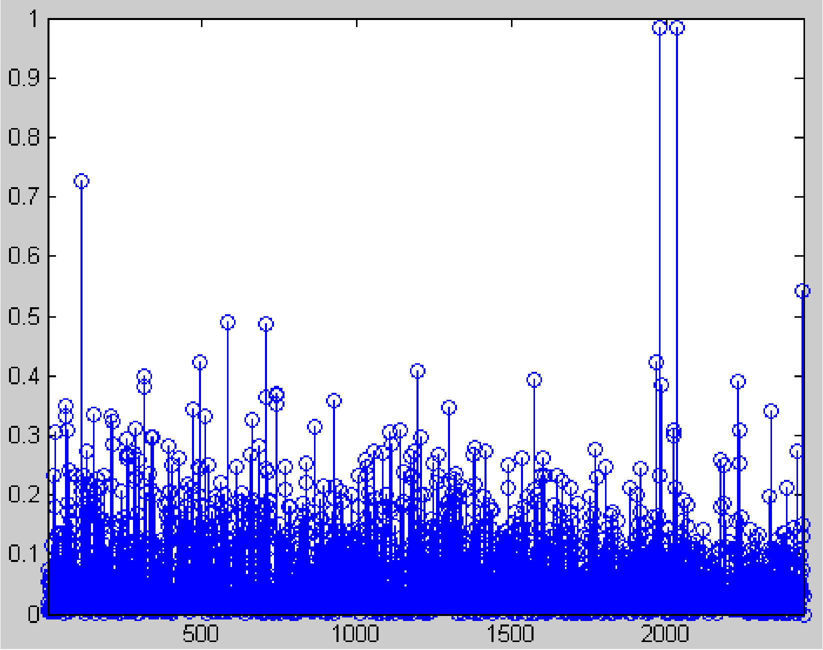


Figure 5.62: NCC results using bispectrum slice constant = 10

As can be seen from the results above, the sixth slice, i.e. a bispectrum slice constant of six, gives the best results for the given template image and the test image. However, more importantly, it is observed that the correlation results, and hence the hits, misses and false alarms vary according to the bispectrum slice constant. Given below are the results for using multiple slices, by which for each sub-image, the slice giving the highest correlation result is used, i.e. the highest correlation result is taken as final. This will obviously raise the number of false alarms, but provide a higher number of hits and a better performance.

5.3.4.2 Multiple Slices

Given below are the results for the same template image and test image, but for multiple slices. Two different results are presented, the first being the result of using the first five slices and the second the first ten.

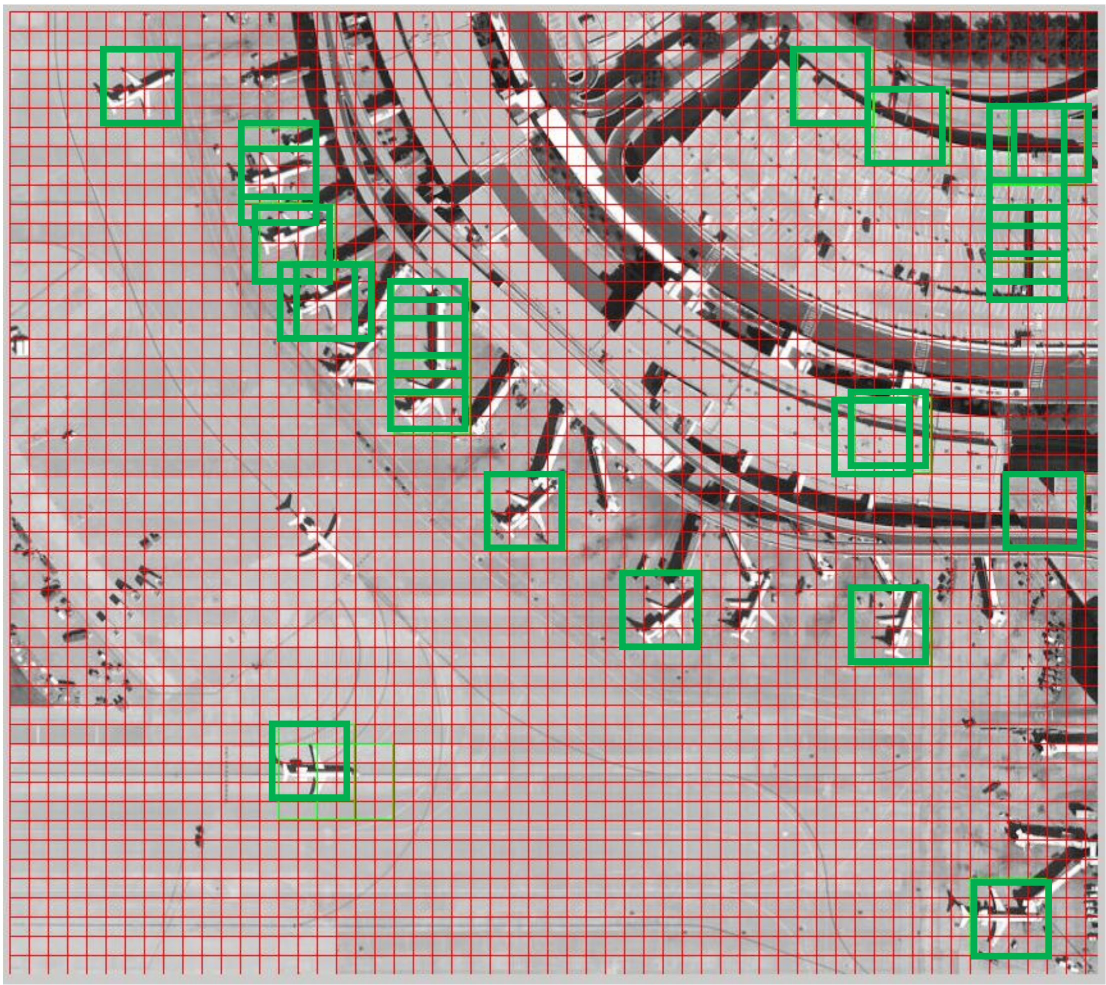


Figure 5.63: Results using the first five bispectrum slice constants, threshold=0.4

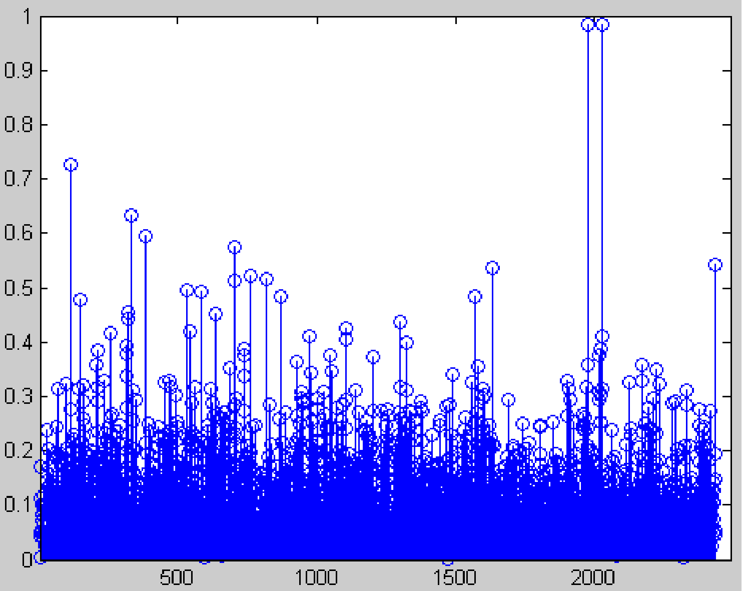


Figure 5.64: NCC results using the first five bispectrum slice constants

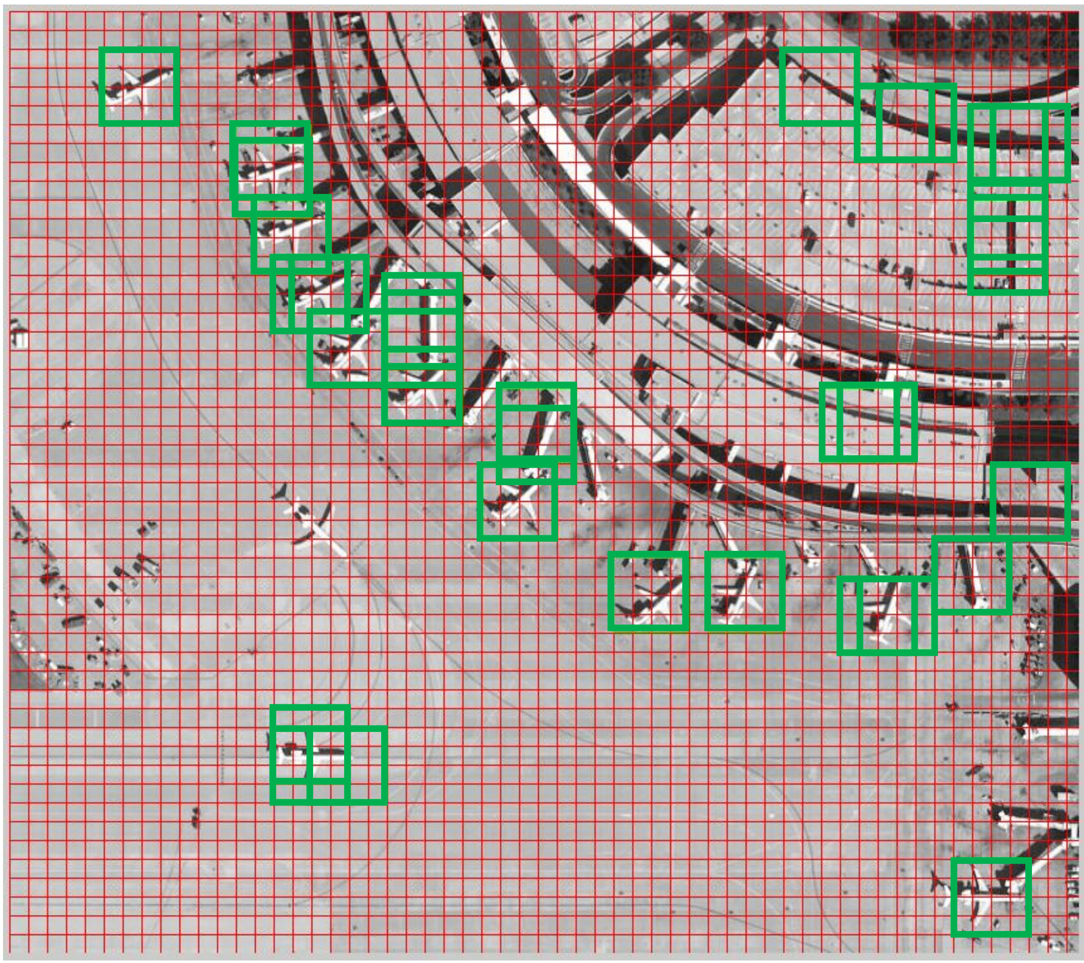


Figure 5.65: Results using the first ten bispectrum slice constants, threshold=0.4

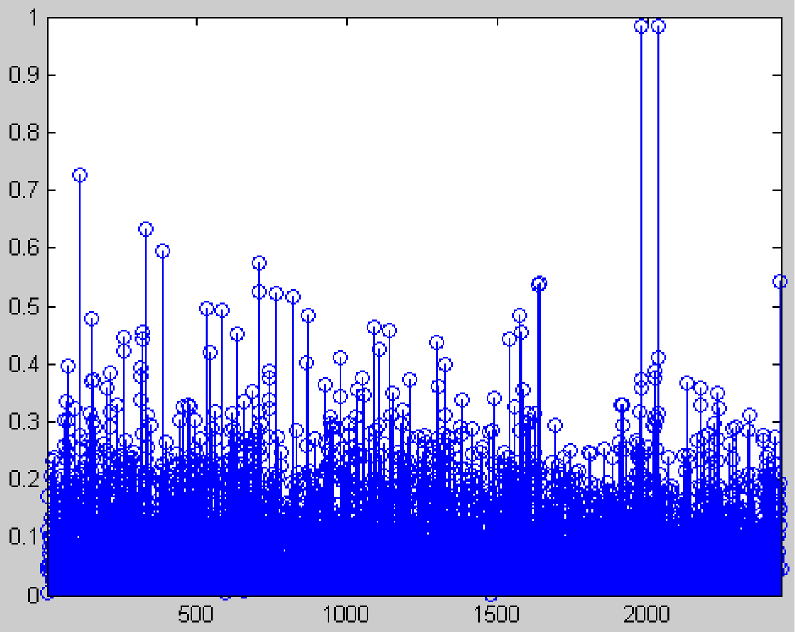


Figure 5.66: NCC results using the first ten bispectrum slice constants

5.3.4.3 Result of the Experiments on Slice Analysis

As can be seen from the exemplary tests presented above in this section, the choice of bispectrum slice constant can affect the results. This effect can be easily observed in the test image chosen for these tests, while in another test image, the effect may not have been observable. Using a larger number of slices generally does provide an improvement in the results, which is significant in some cases. However, the computation time is increased to the number of slices times the calculation time for the same method using a single slice. Hence, the increase in computation is also significant. Also, using multiple slices can decrease precision, since the number of false alarms can increase. Whether to use multiple slices or not, and how many slices to use if decided on using, are choices to be made by the user of the method.

5.4 Performance Tests

In this section, the performances of the methods used in this study are compared. These methods are, in order, template matching using FMT, template matching using reconstruction with FMT, template matching using reconstruction with bispectrum and LPM, the eigen approach presented in Chapter 4 and the exhaustive case. Note that both the eigen approach and the so-called exhaustive case in this study involve translation invariance by phase correlation.

5.4.1 Effect of Bispectrum slice constant on Recall vs. Precision

The effect of bispectrum slice constant on performance will be presented one more time in this section, but this time in terms of recall and precision. Note that the results presented are only for the given template image and test image, and in no way generally-applicable. In other words, a particular bispectrum slice constant that gives the best results for a test image can be the worst for another. Or the effect of bispectrum slice constant can be completely negligible for another template image and test image duo.

Figure 5.67: Template image for the effect of bispectrum slice constant on recall vs. precision

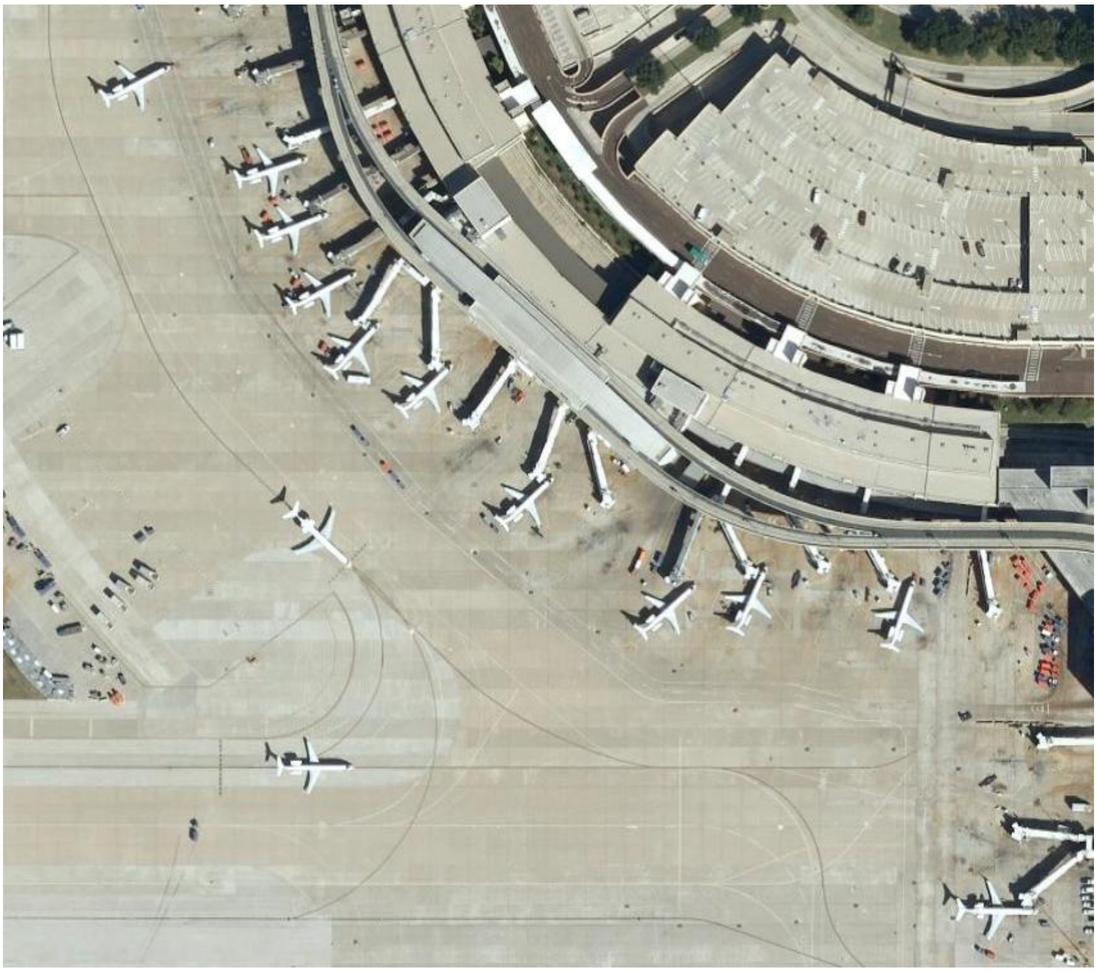


Figure 5.68: Test image for the effect of bispectrum slice constant on recall vs. precision

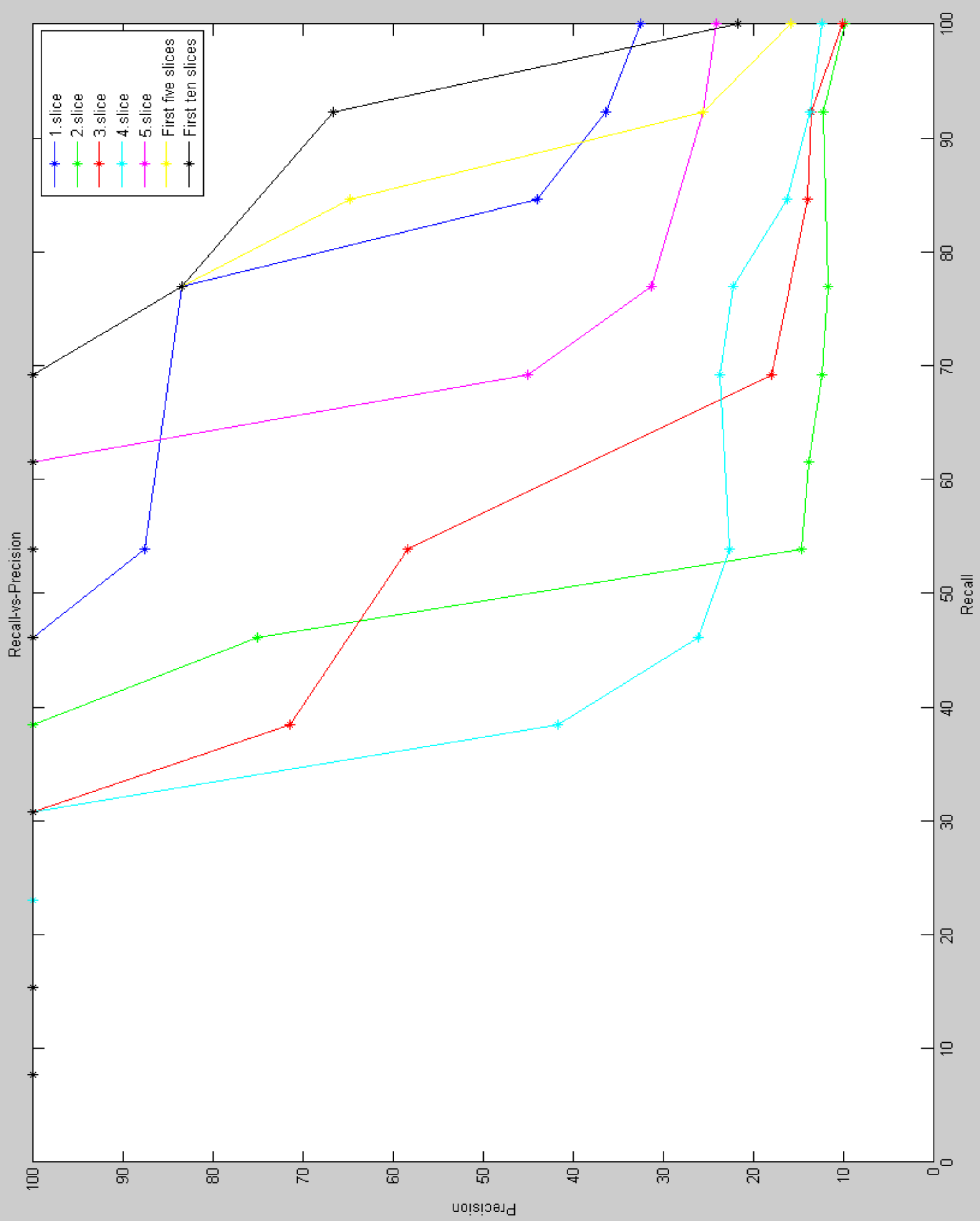


Figure 5.69: Recall vs. precision graph to observe the effect of bispectrum slice constant

As expected from the results of Section 5.3.4, it can be seen that different bispectrum slice constants result in different recall vs. precision graphs and using multiple slices generally results in better performance in terms recall and precision, for this template image and test image pair.

5.4.2 First Performance Comparison Test

A subgroup of the database is used in this section to compare the performances of the methods used in this study. A single template image is used for three test images, which contain a total number of 26 targets of the same type as the target in the template image. The target is a type of civilian aircraft. The subgroup is realistic in terms of real-life applications, a large number of targets are covered with just one template and, most important of all, the targets are aligned in various directions, which ensures that the RST invariance methods are necessary and their performances are better observed.

5.4.2.1 Test Data for the First Test

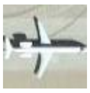


Figure 5.70: Template image for the first performance comparison test

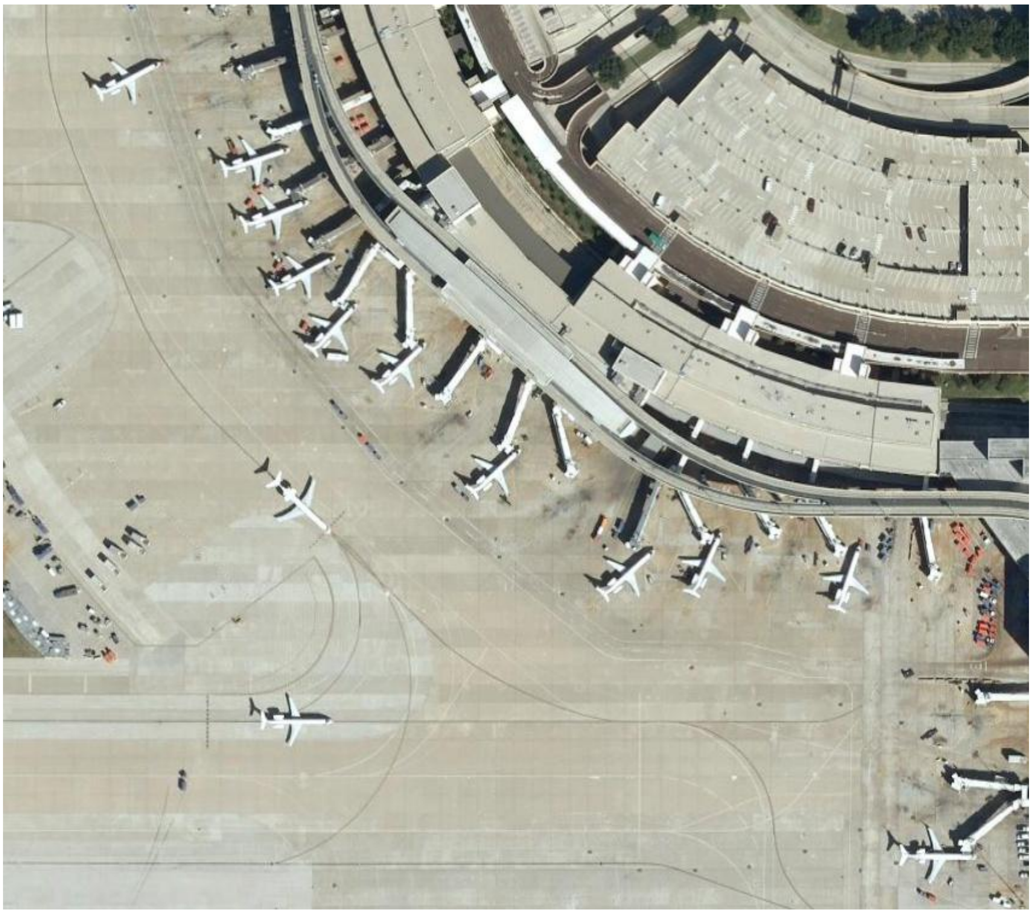


Figure 5.71: First test image for the first performance comparison test

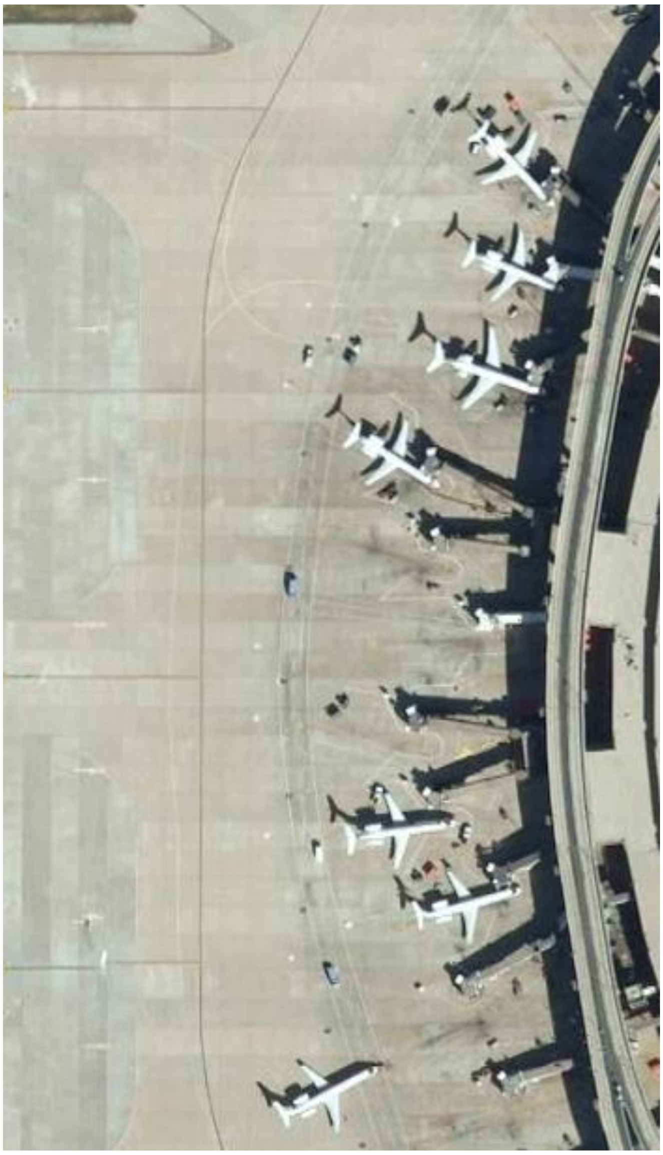


Figure 5.72: Second test image for the first performance comparison test

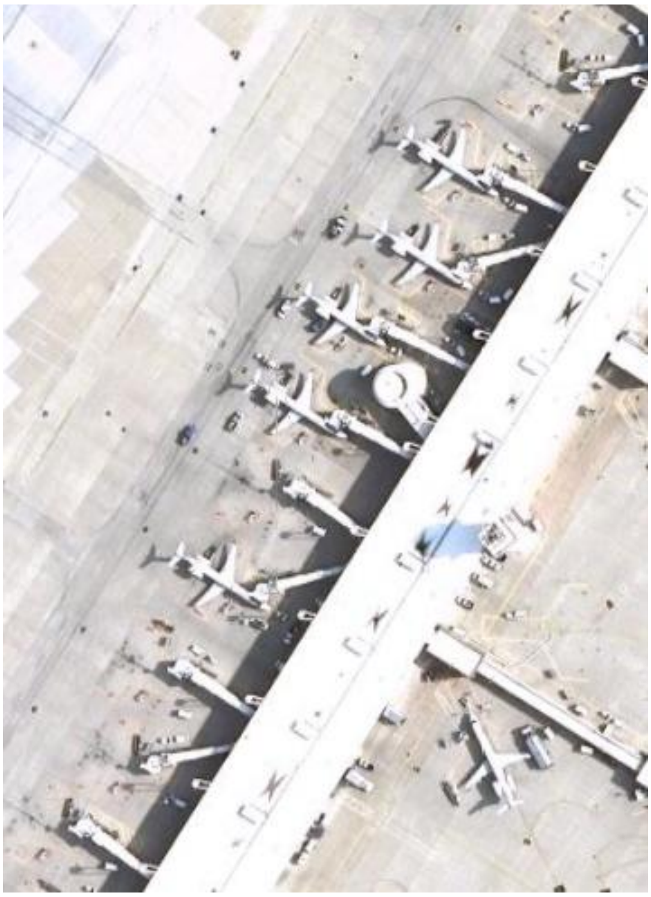


Figure 5.73: Third test image for the first performance comparison test

5.4.2.2 Recall vs. Precision Graphs for the First Test

Given below in Figure 5.74 are the performance graphs of the methods used in this study, for the first test group, in terms of recall vs. precision. Note that for the reconstruction-based method using bispectrum and LPM, both the results for using the first five bispectrum slices and the first ten bispectrum slices are presented. The analysis of the performance graphs will be presented in Section 5.4.2.3.

Given below in Figure 5.75 is the comparison of the performances of the last method using different bispectrum slice constants. Only the results for the first five bispectrum slice constants will be presented for representation and clarity purposes, but it will be enough to observe that different bispectrum slice constant do result in different recall vs. precision graphs, and hence have different performances.

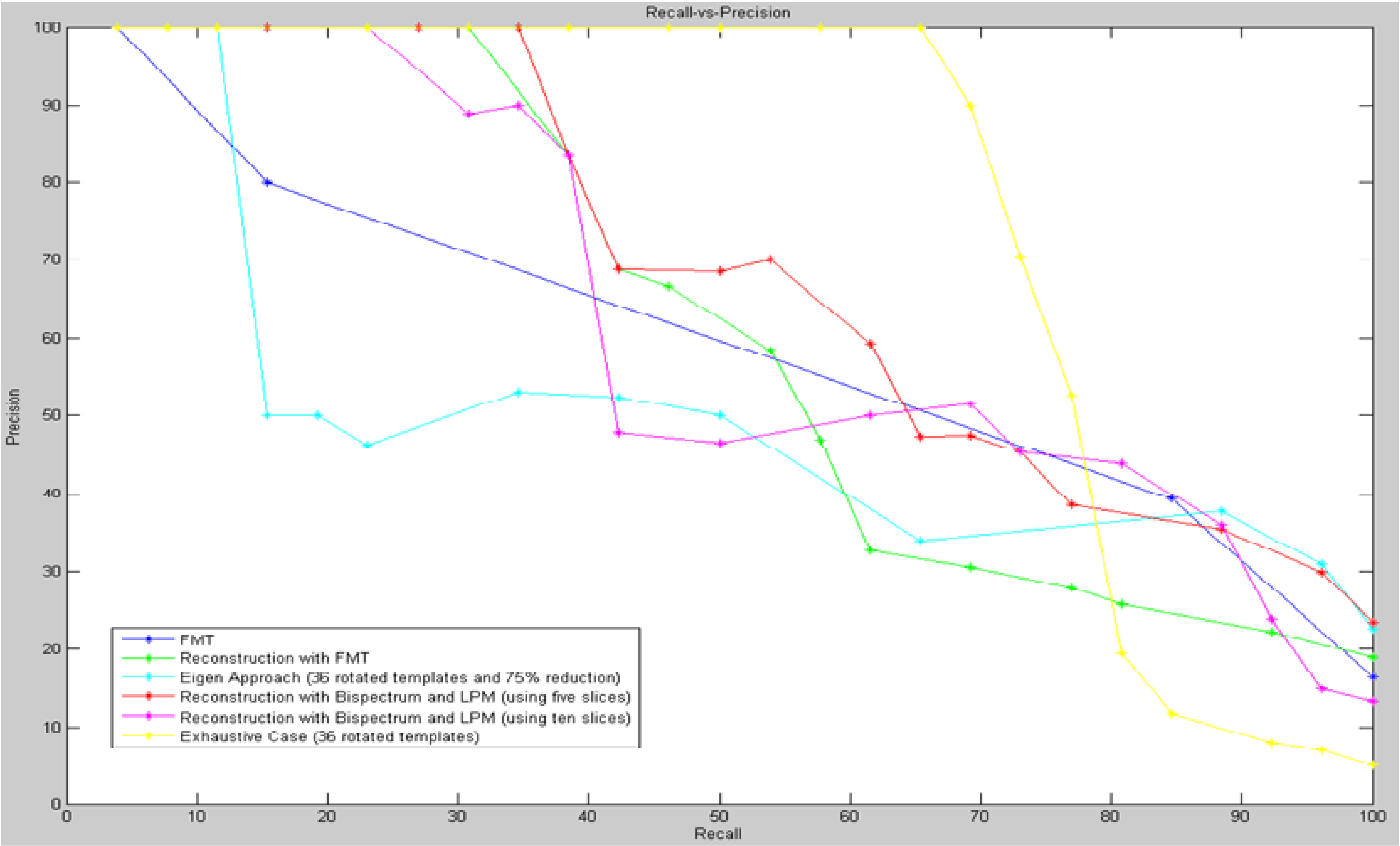


Figure 5.74: Results for the first performance comparison test

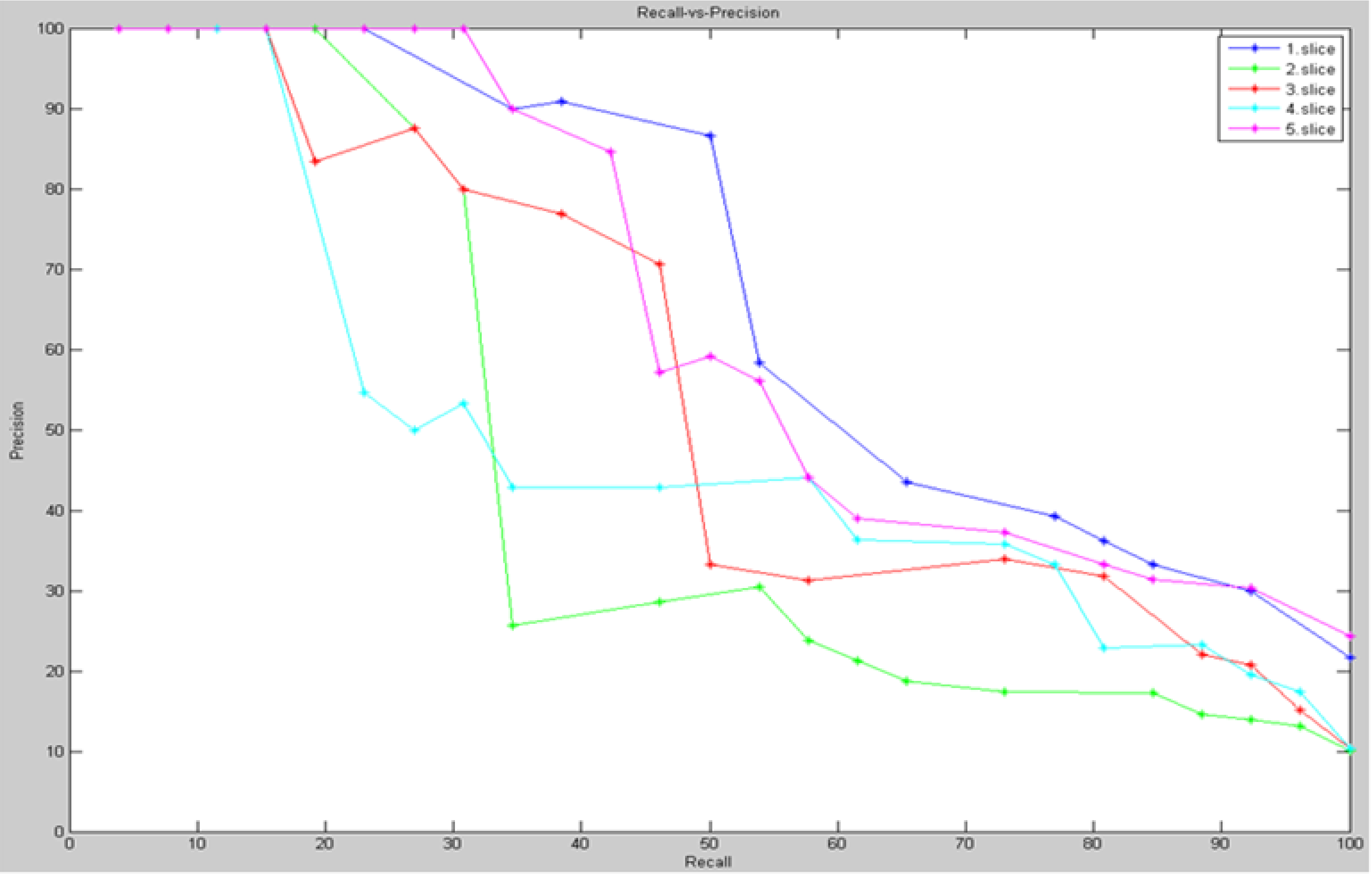


Figure 5.75: Performance graphs for different bispectrum slice constants of the last method for the first performance comparison test

5.4.2.3 Results of the First Performance Comparison Test

As can be seen from Figure 5.74, moderate performance curves were obtained for this test group.

The exhaustive case with translation invariance was the best performing method from low to middle recall rates and the worst performing method for high recall rates. This behavior is hard to explain and does not appear in any other performance test conducted throughout the scope of this study.

Overall, and excluding the exhaustive case, the reconstruction-based method using bispectrum and LPM can be said to have performed the best among other methods.

5.4.3 Second Performance Comparison Test

In this test group, another group of civilian aircrafts were selected as targets. 14 test images which contain a total number of 40 targets, of three target types, were used. Three template images were used for the three target types.

5.4.3.1 Test Data for the Second Test

Given below are the three template images and the fourteen test images used for the performance test in this section.

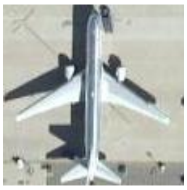


Figure 5.76: First template image for the second performance comparison test

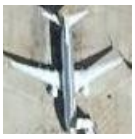


Figure 5.77: Second template image for the second performance comparison test

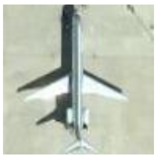


Figure 5.78: Third template image for the second performance comparison test

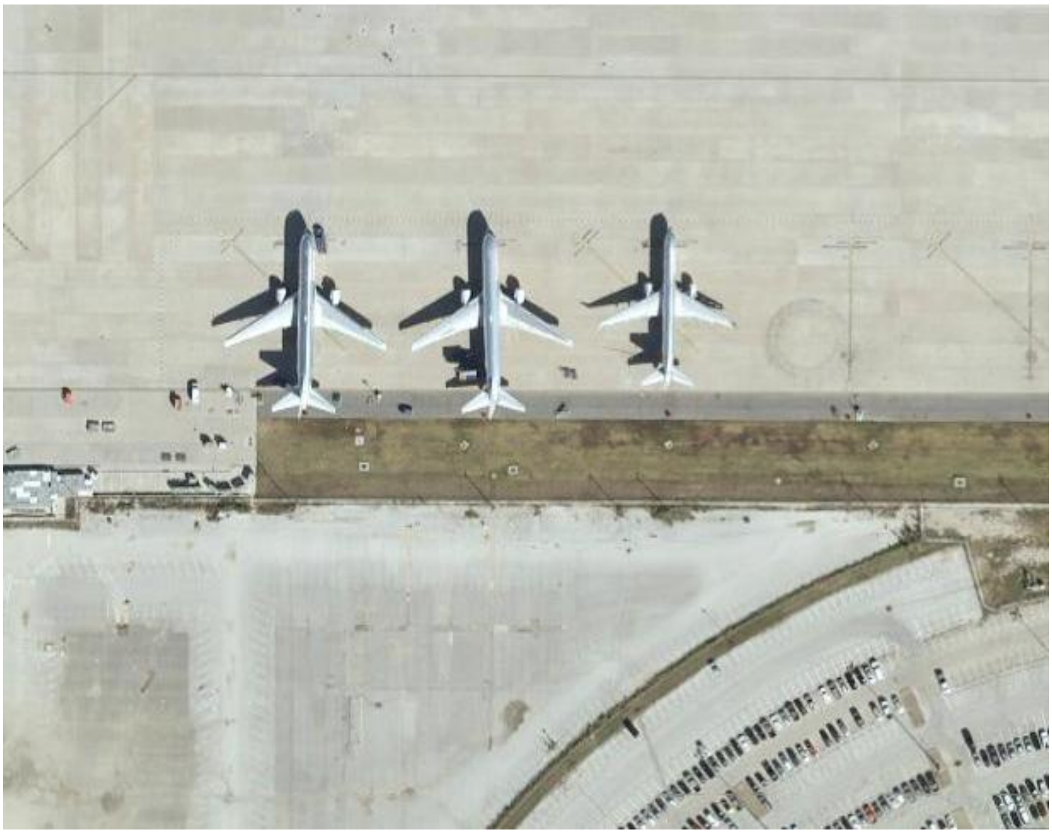


Figure 5.79: First test image for the second performance comparison test

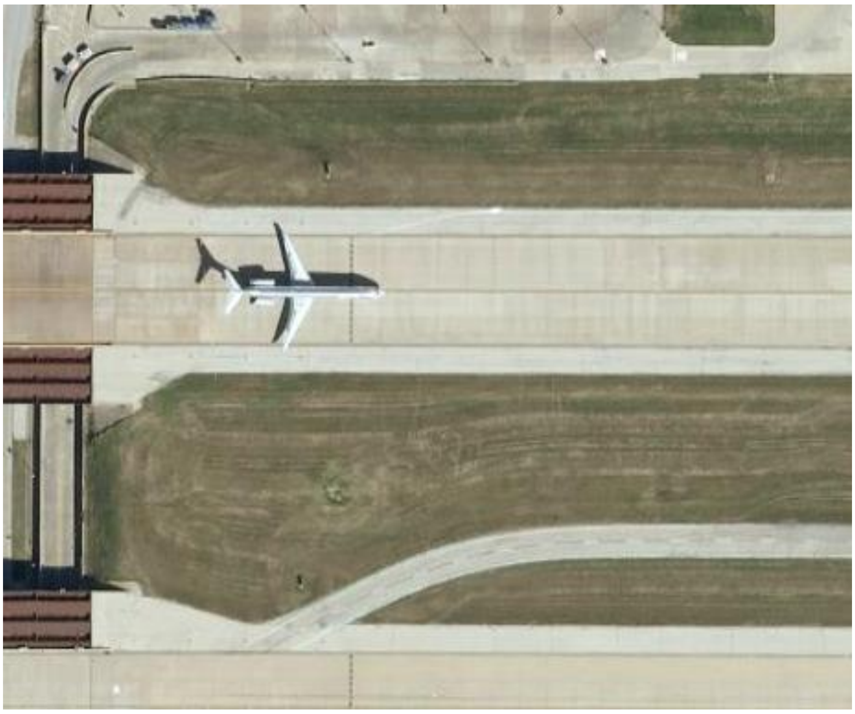


Figure 5.80: Second test image for the second performance comparison test

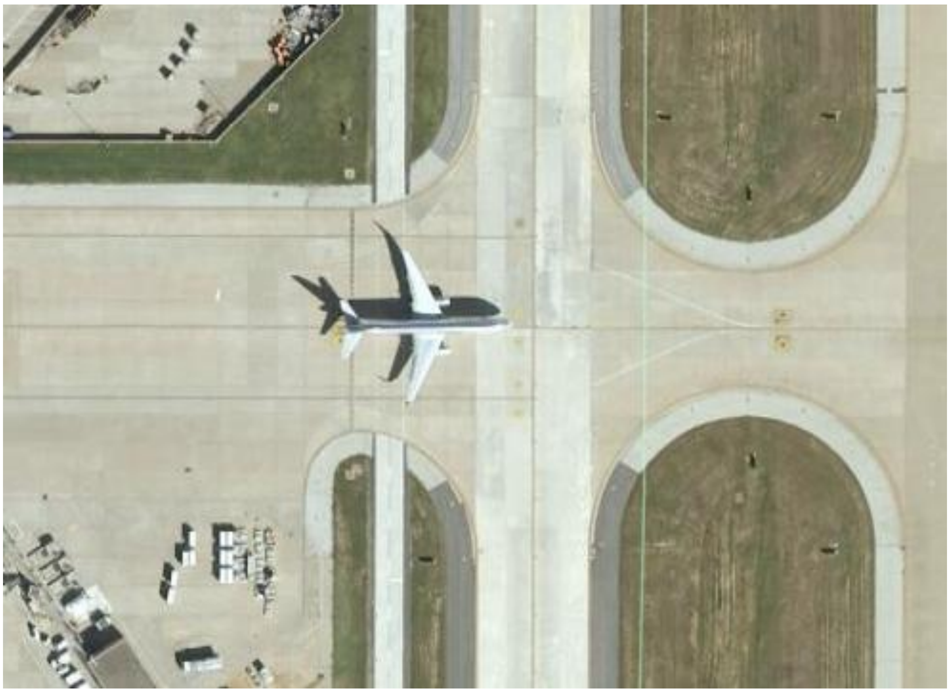


Figure 5.81: Third test image for the second performance comparison test

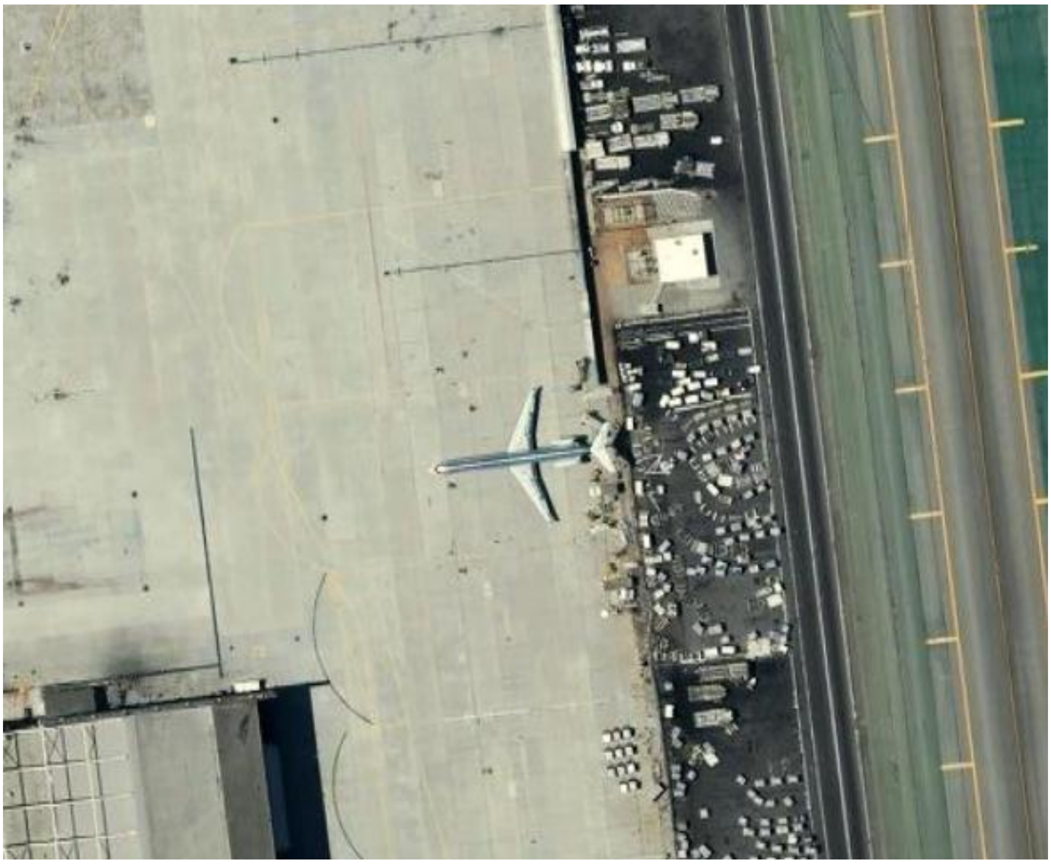


Figure 5.82: Fourth test image for the second performance comparison test

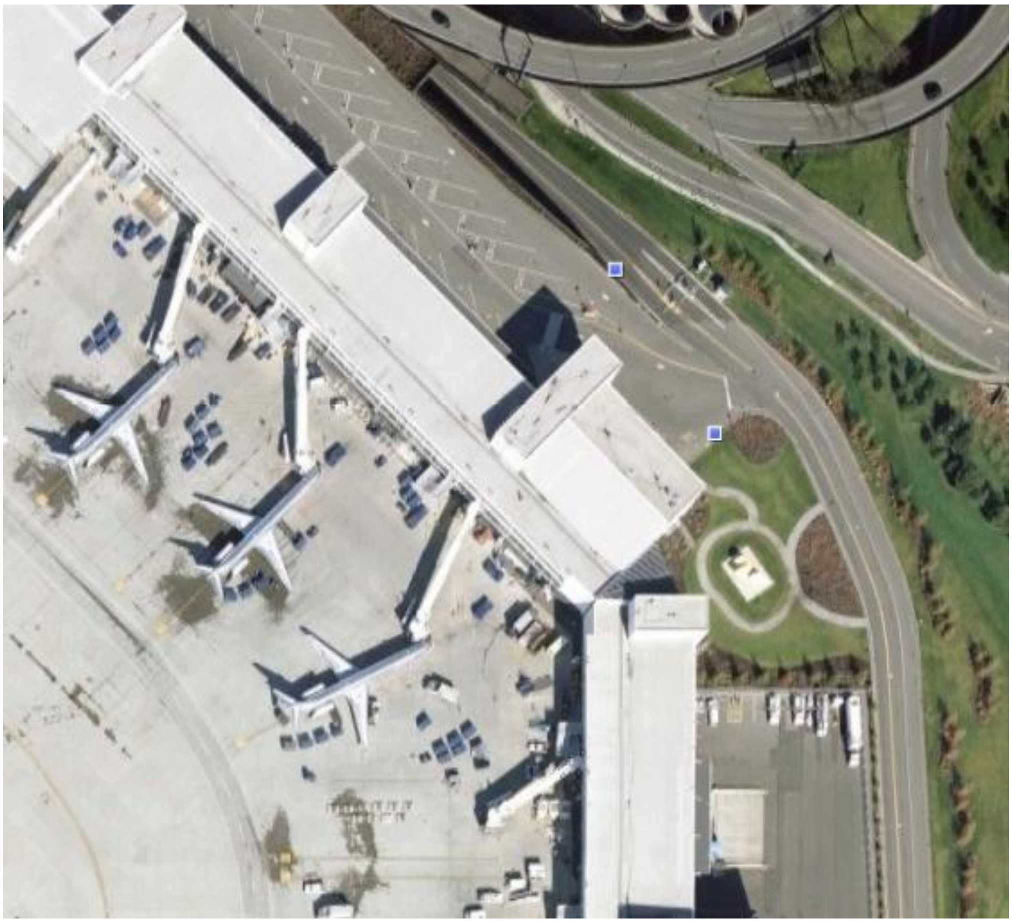


Figure 5.83: Fifth test image for the second performance comparison test

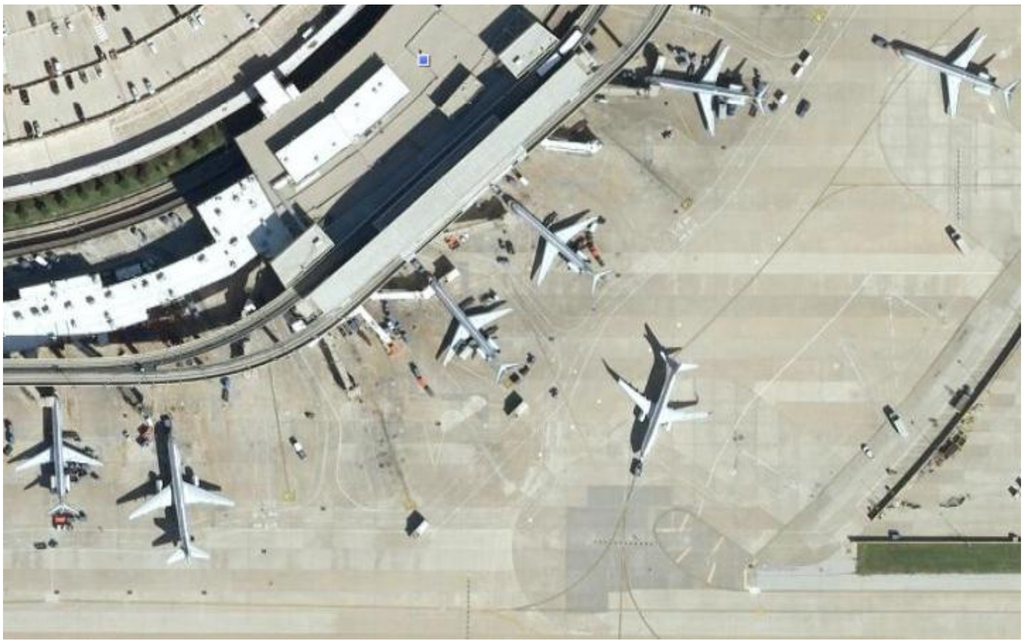


Figure 5.84: Sixth test image for the second performance comparison test

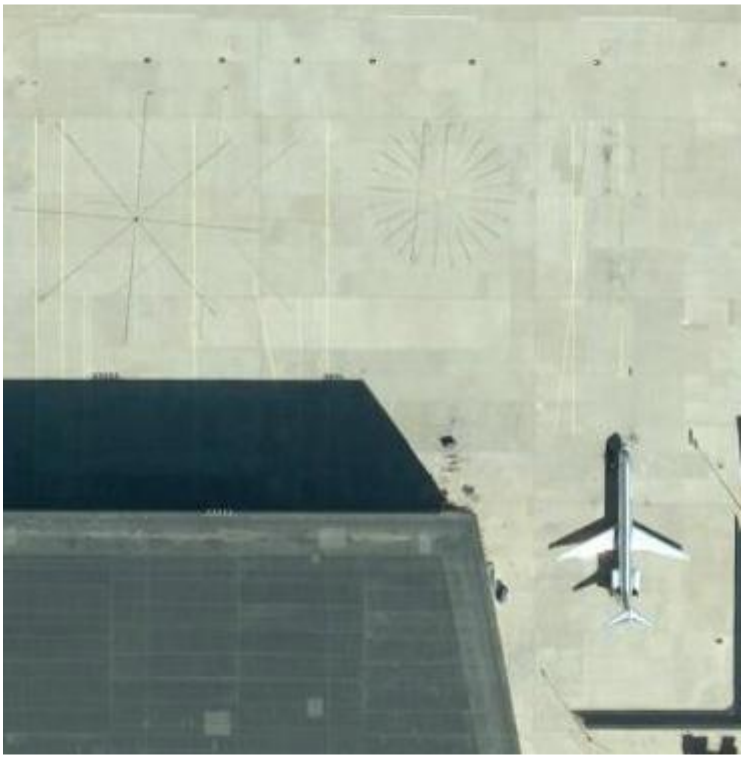


Figure 5.85: Seventh test image for the second performance comparison test

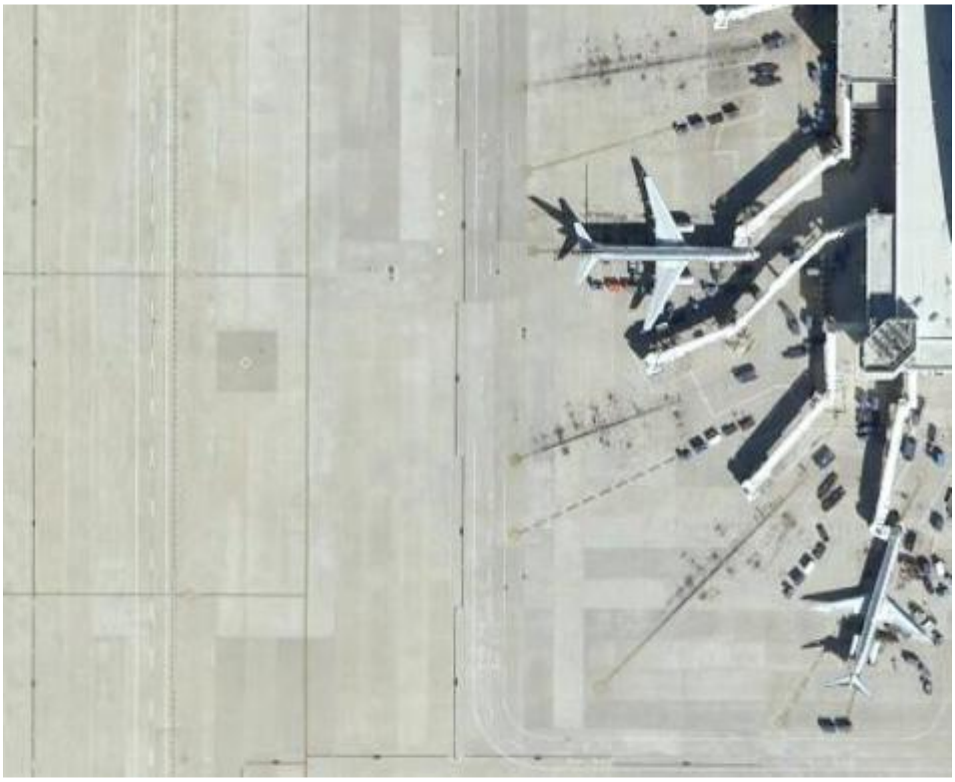


Figure 5.86: Eighth test image for the second performance comparison test

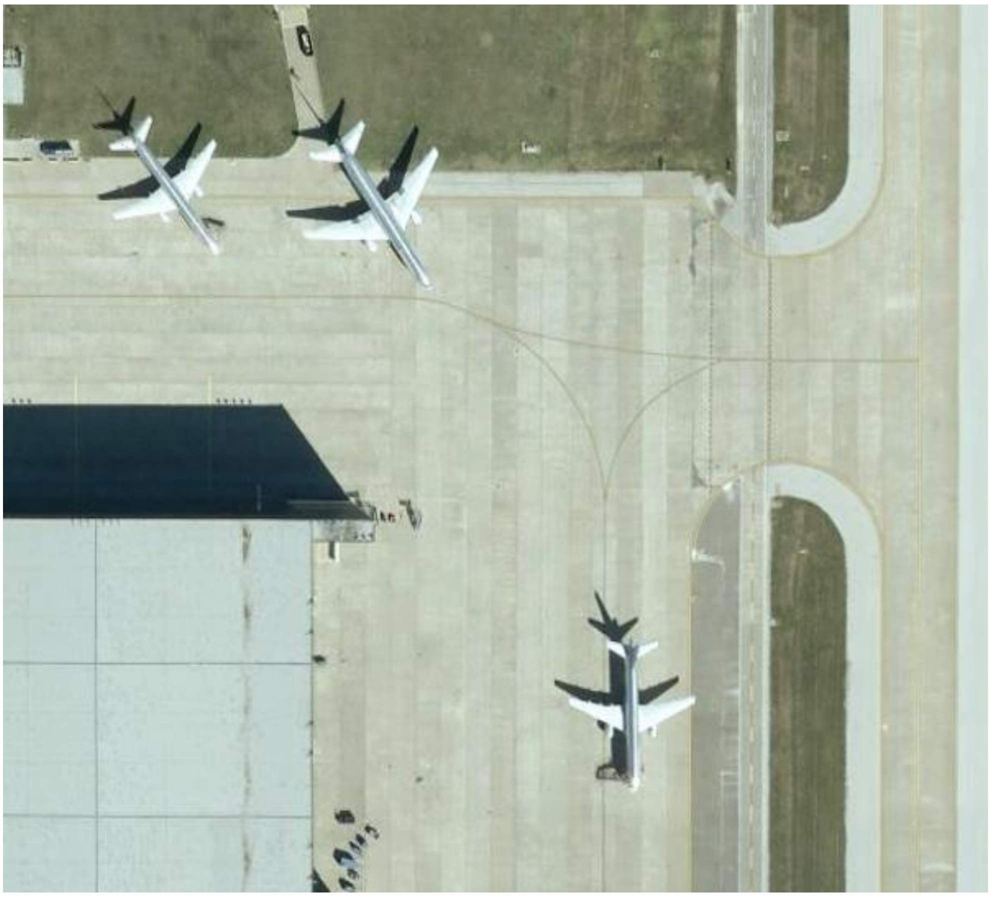


Figure 5.87: Ninth test image for the second performance comparison test

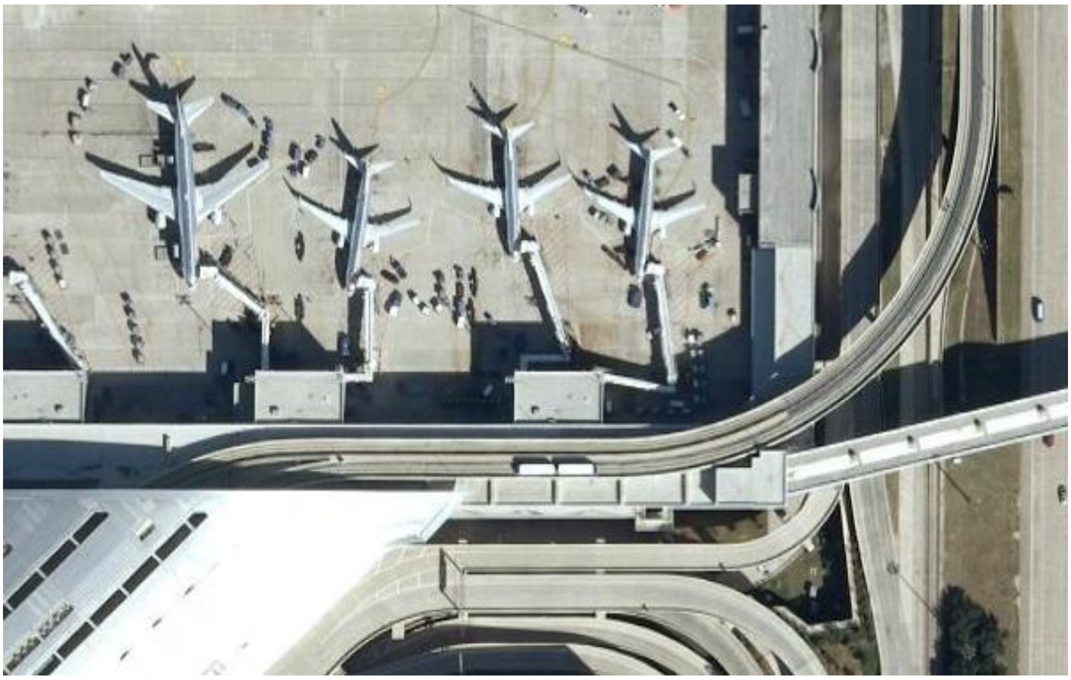


Figure 5.88: Tenth test image for the second performance comparison test

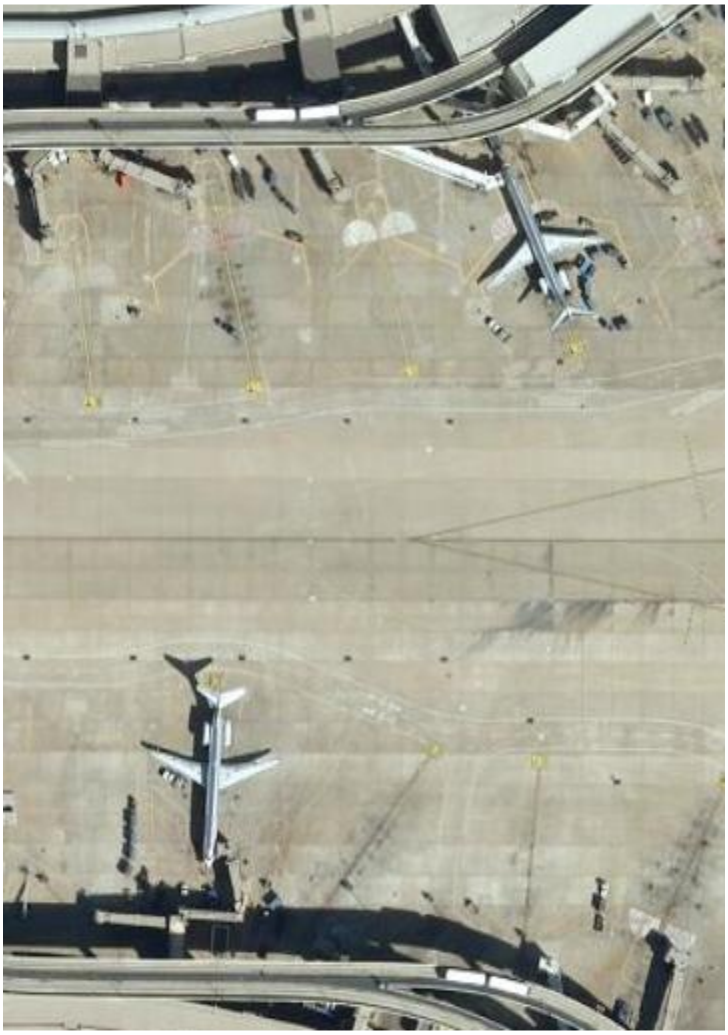


Figure 5.89: Eleventh test image for the second performance comparison test

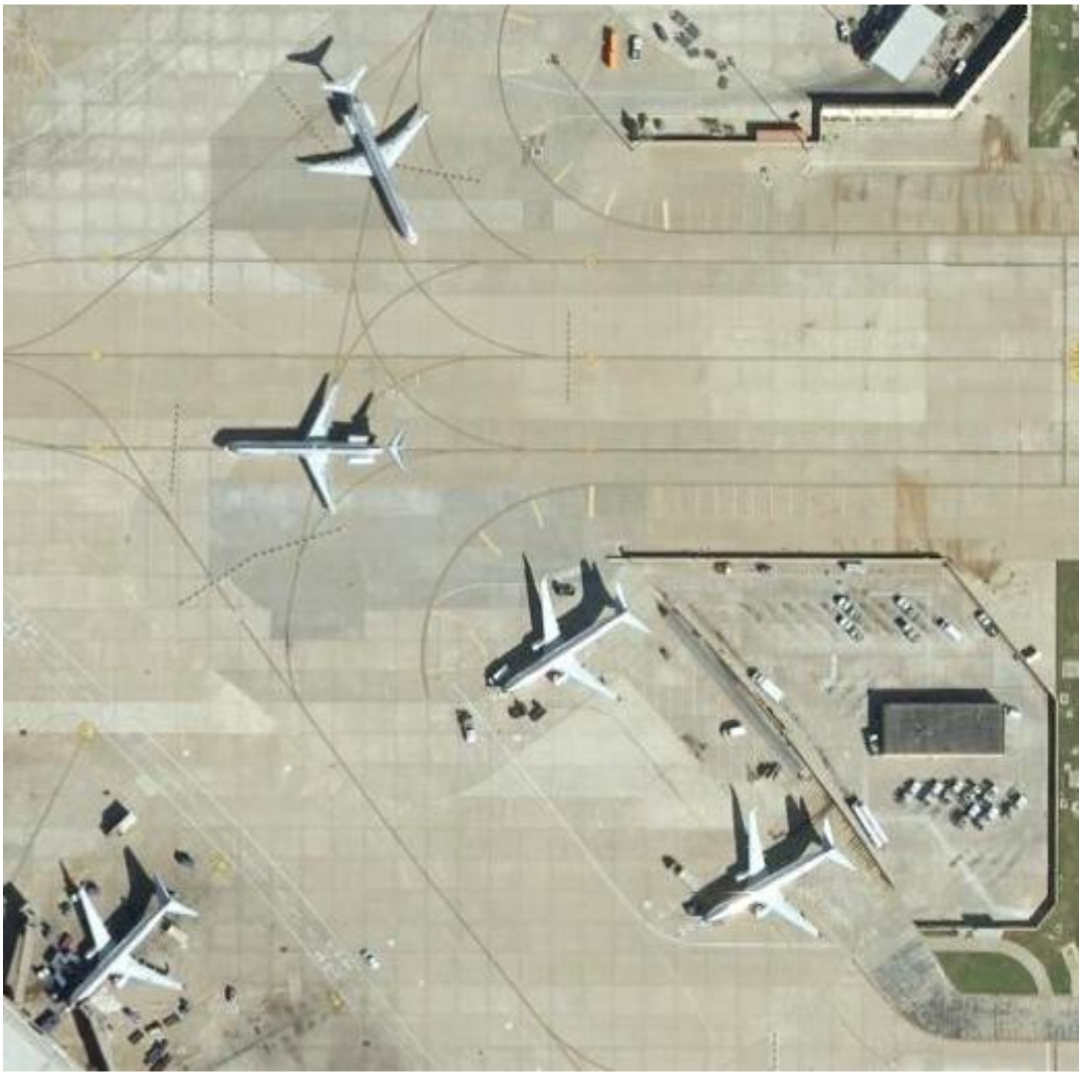


Figure 5.90: Twelfth test image for the second performance comparison test

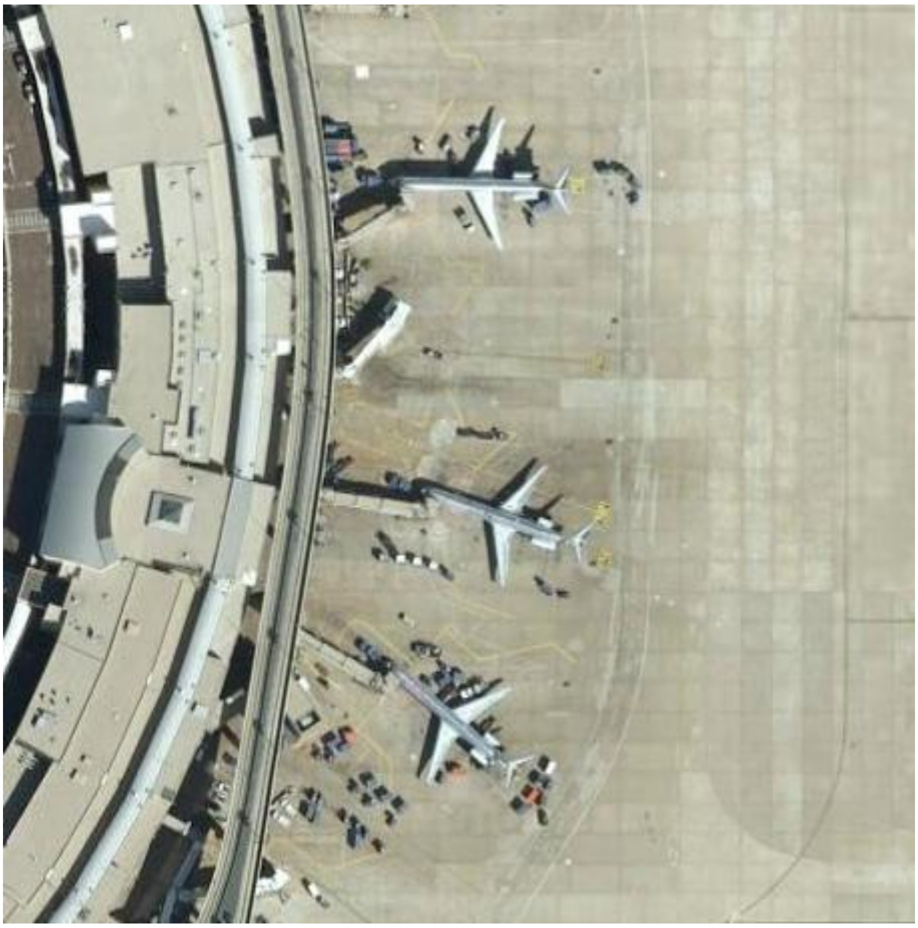


Figure 591: Thirteenth test image for the second performance comparison test

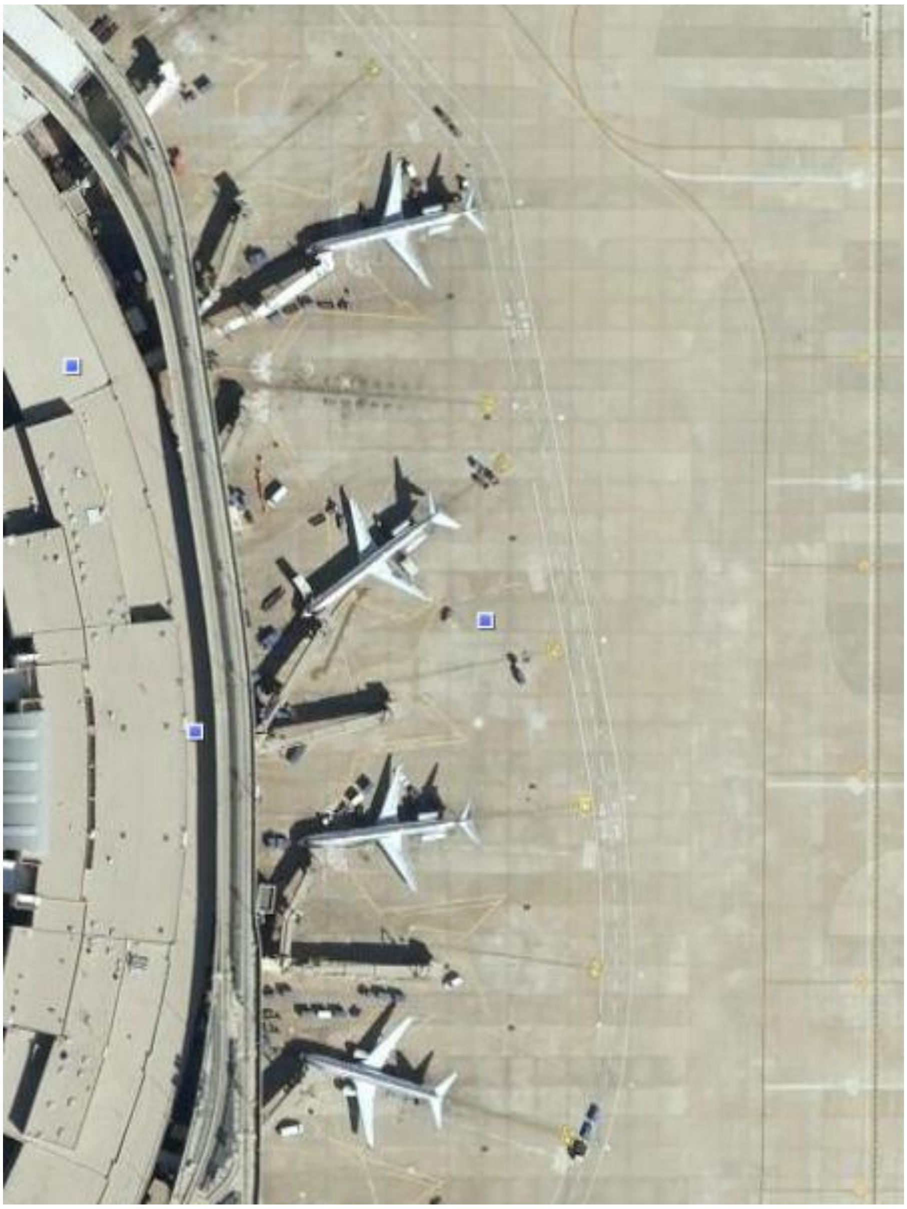


Figure 5.92: Fourteenth test image for the second performance comparison test

5.4.3.2 Recall vs. Precision Graphs for the Second Test

Given below in Figure 5.93 are the performance curves of the methods used in this study for the given test data. The performance curves are presented in terms of recall vs. precision.

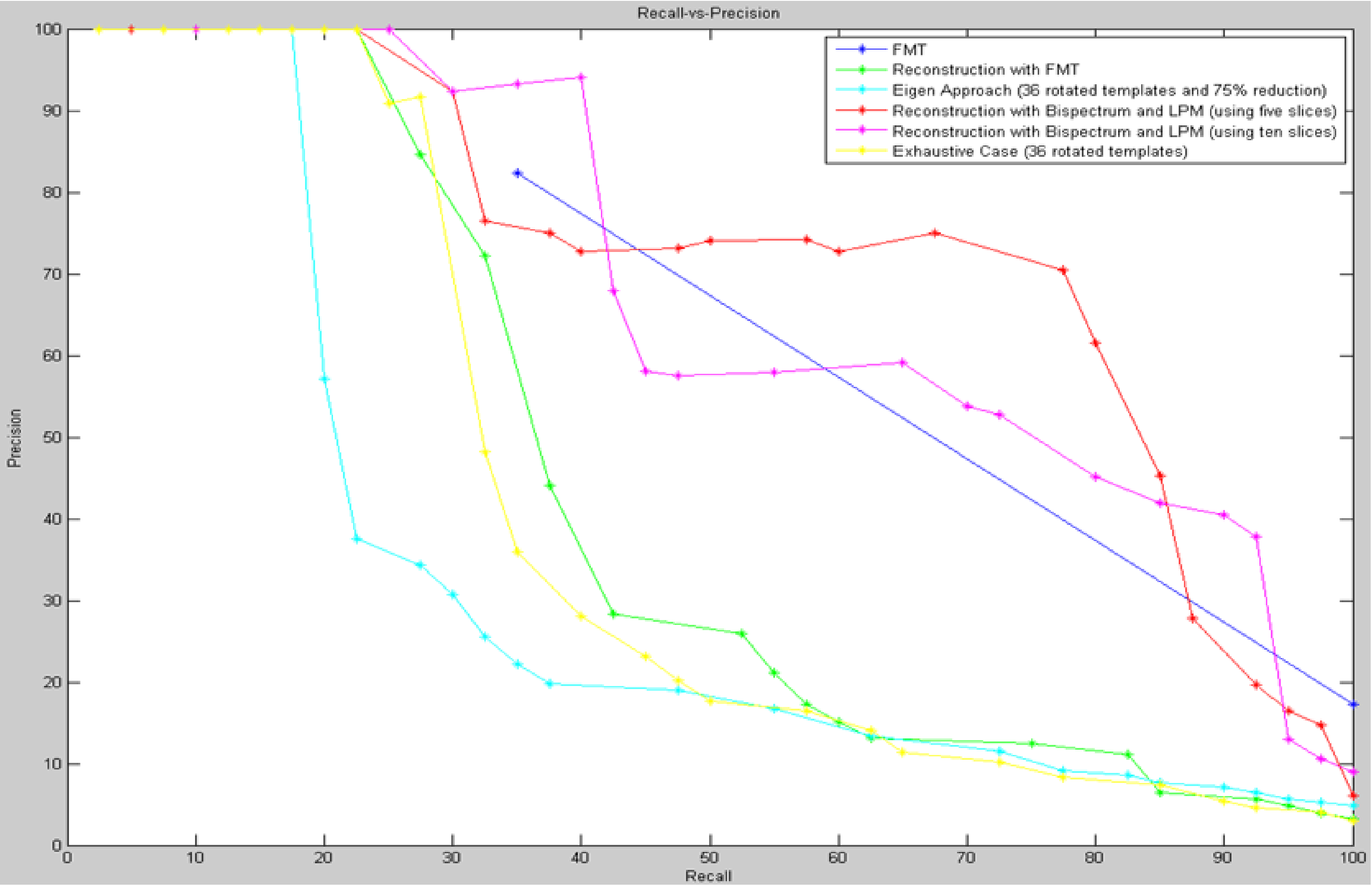


Figure 5.93: Results for the second perfomance comparison test

5.4.3.3 Results of the Second Performance Comparison Test

For this performance test, the eigen approach using 36 rotated templates and 75% eigenvalue reduction provided the worst performance curve over all the techniques used in this study.

Exhaustive case using 36 rotated templates provided a better curve than the eigen approach, as expected, but was short of the performance curves of the RST invariant methods.

The reconstruction-based RST-invariant method using FMT had a lower performance curve than expected, which was almost equal to the performance curve of the exhaustive method using 36 rotated templates.

The performance of the method using FMT seems high, but it should be noted that this curve is a linearization between just two pairs of possible recall and precision points.

Overall, the RST invariant method using bispectrum and LPM provided the highest recall vs. precision rates for this test group. The performance curves of the method using five and ten bispectrum slices are competing, and hence it is difficult to choose one over the other, but a more efficient way to choose slices, instead of using multiple slices, would solve this drawback.

5.4.4 Third Performance Comparison Test

Another subgroup of the database is used in this section to compare the performances of the methods used in this study. This subgroup contains 38 targets which are camouflaged jet fighters. The false alarm rate is expected to increase, and hence a lower rate of precision is predicted due to the similarity between the targets and the background. A single template image is used to detect all 38 targets in 7 test images of various sizes.

5.4.4.1 Test Data for the Third Performance Test

In this section, the template image and the test images for the second performance comparison test will be presented.

Figure 5.94: Template image for the third performance comparison test

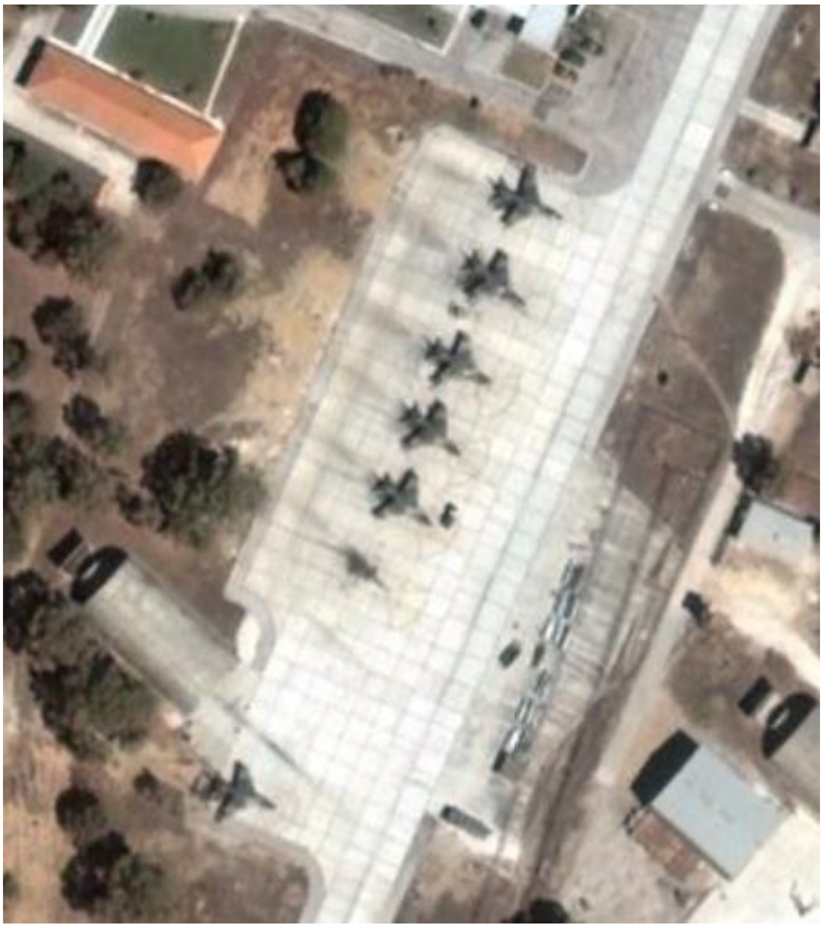


Figure 5.95: First test image for the third performance comparison test

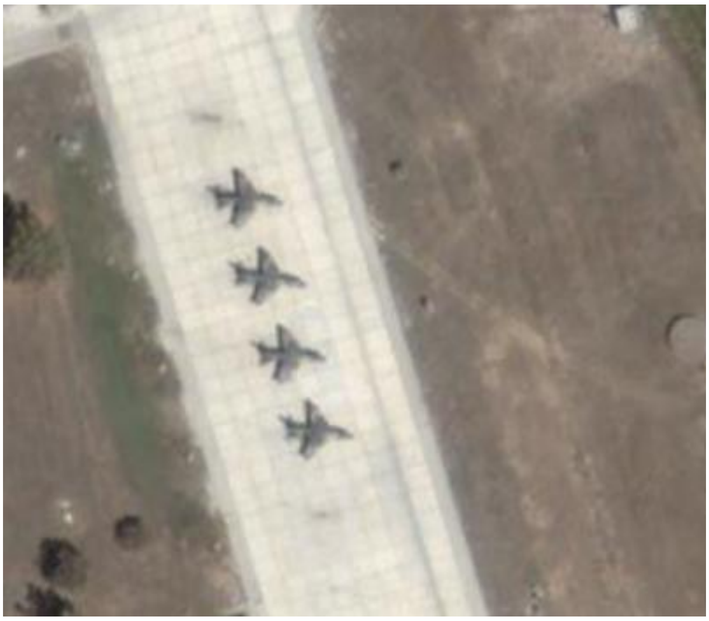


Figure 5.96: Second test image for the third performance comparison test

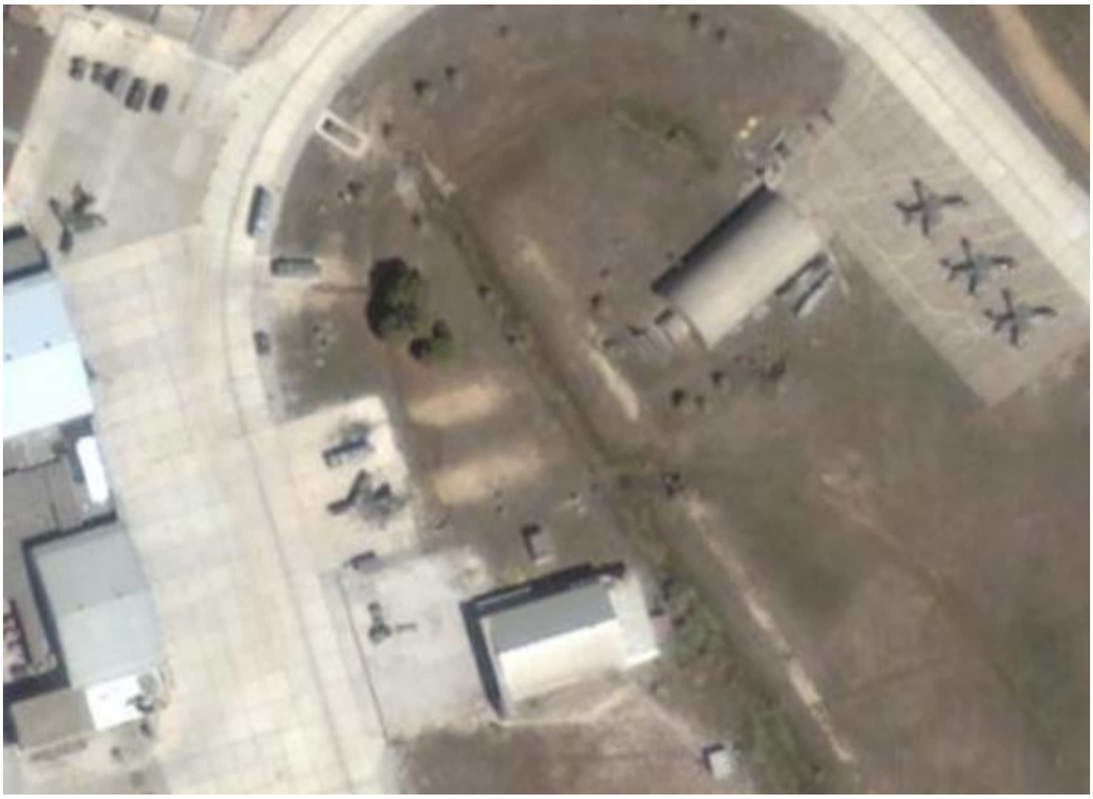


Figure 5.97: Third test image for the third performance comparison test

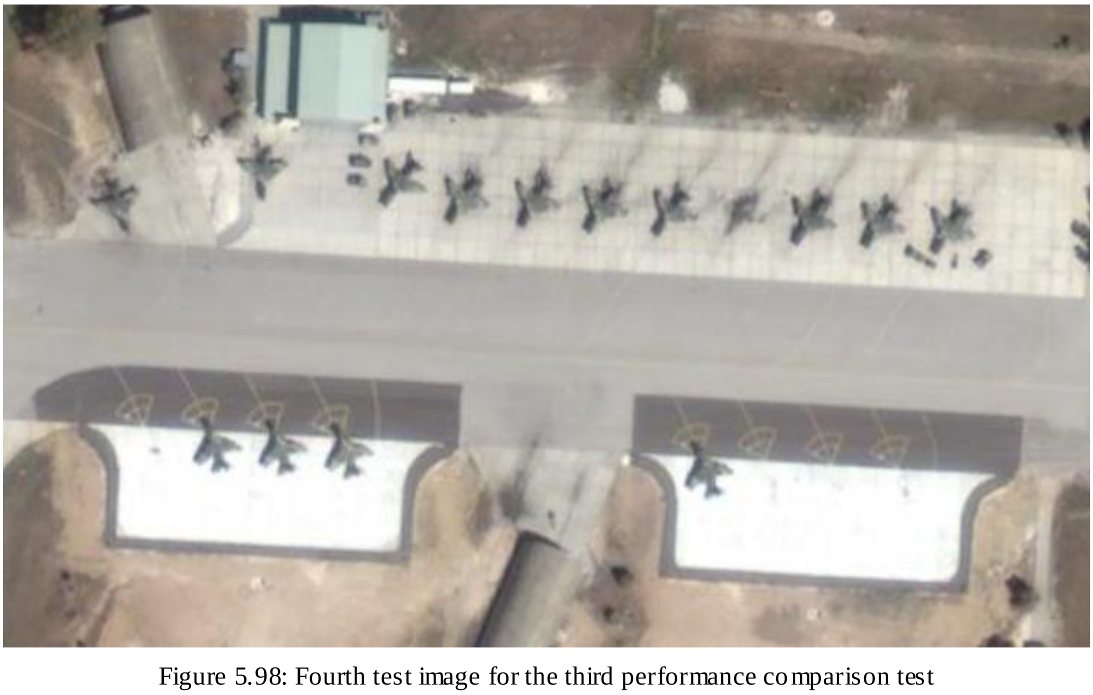


Figure 5.98: Fourth test image for the third performance comparison test

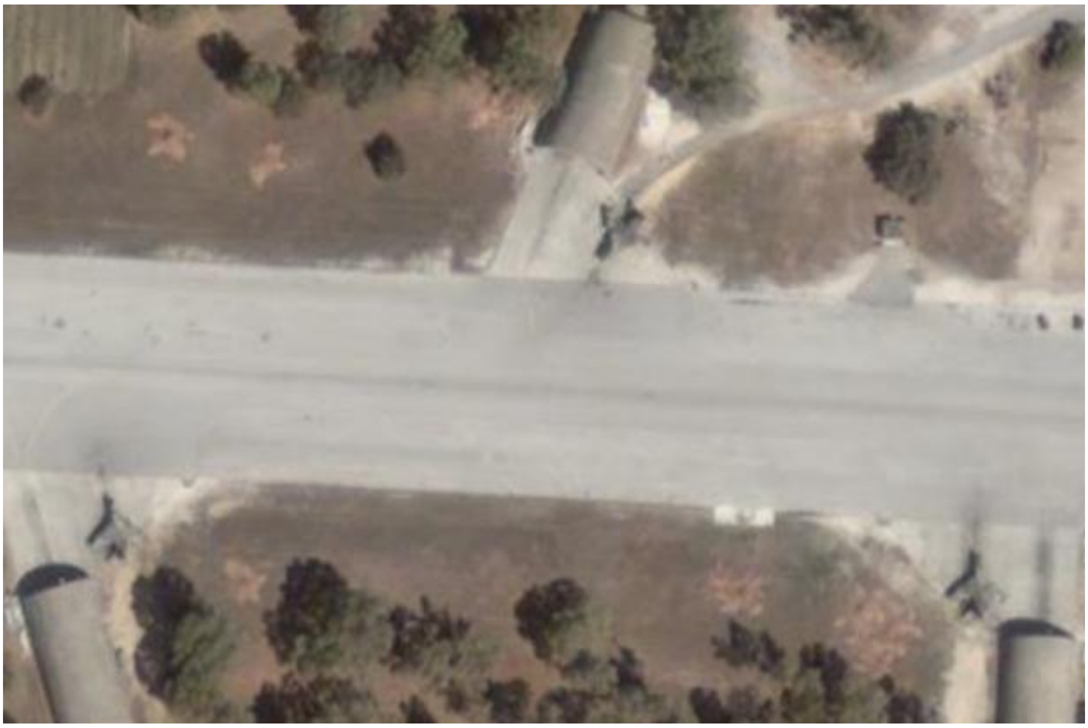


Figure 5.99: Fifth test image for the third performance comparison test

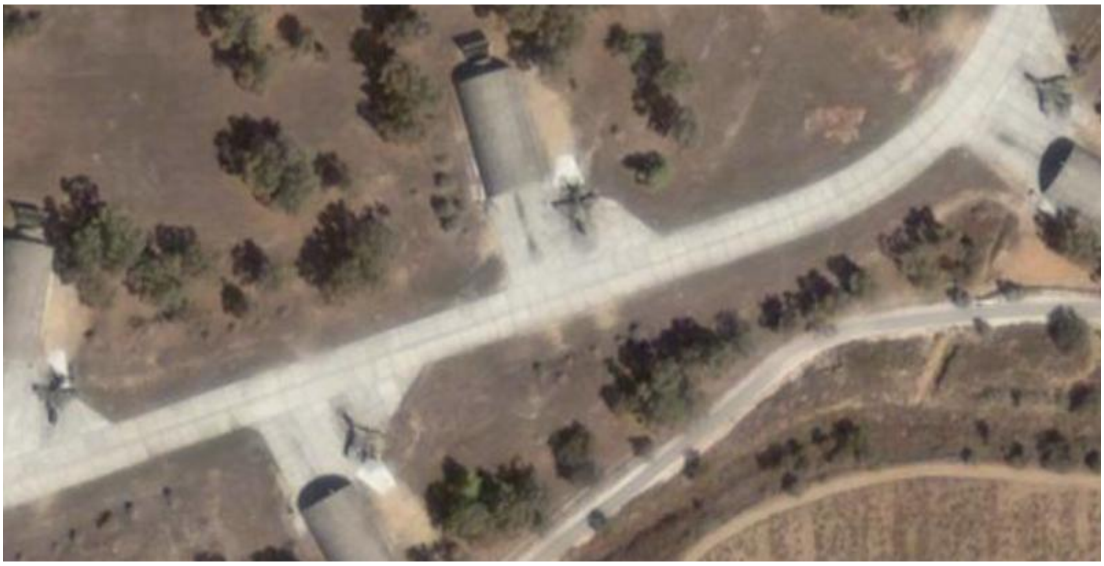


Figure 5.100: Sixth test image for the third performance comparison test

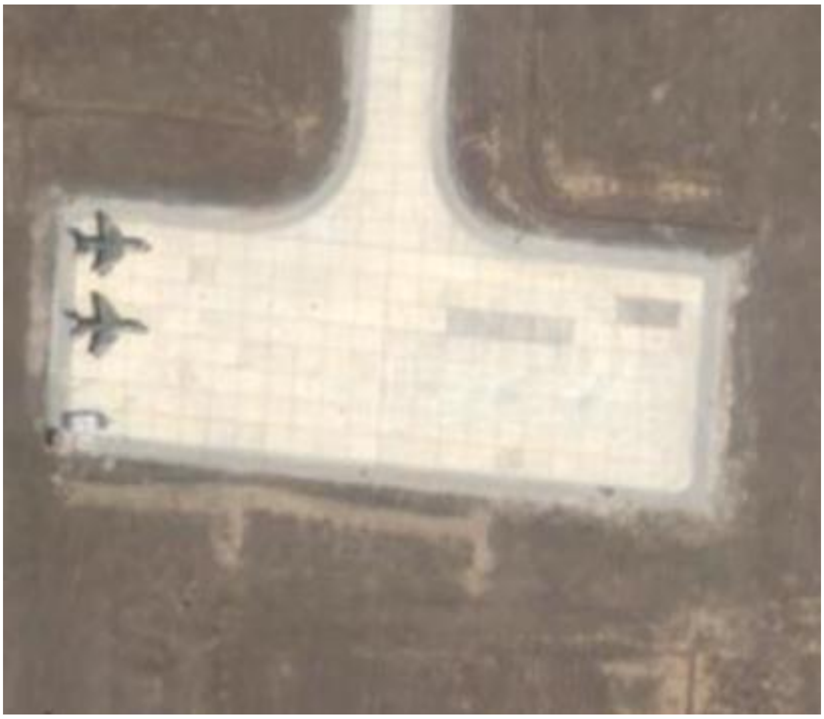


Figure 5.101: Seventh test image for the third performance comparison test

5.4.4.2 Recall vs. Precision Graphs for the Third Performance Test

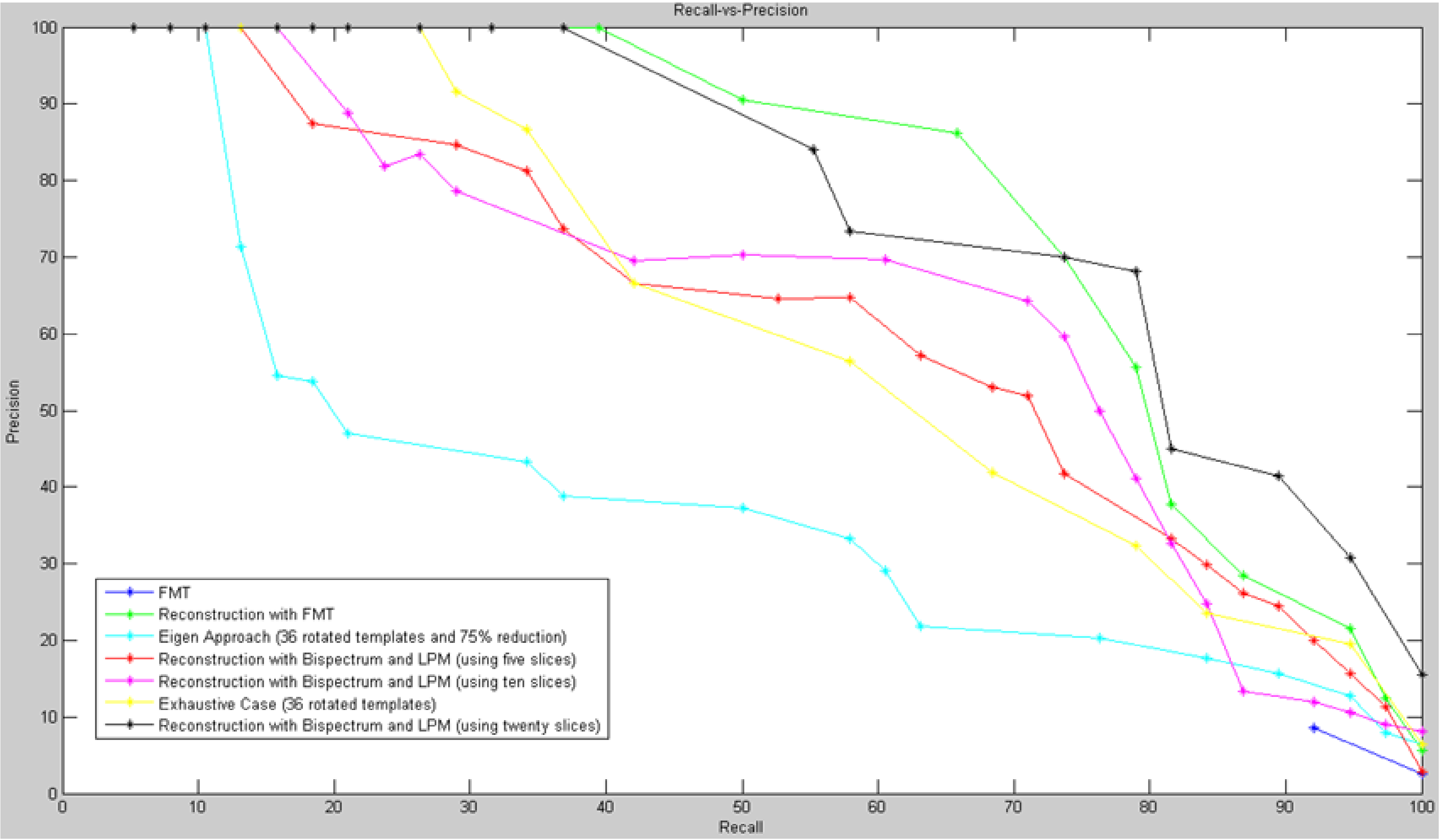


Figure 5.102: Results for the third performance comparison test

5.4.4.3 Results of the Third Performance Comparison Test

As can be seen from Figure 5.102, good performance rates were obtained even though the test group consists of targets that are hard to detect. The reason for this is that even though the targets themselves are hard to detect, the backgrounds of all the test images are similar to the background of the template image.

The method using FMT provided a very limited performance curve which is stuck at high recall and low precision rates.

Eigen approach, using 36 rotated template images and an eigenvalue reduction of 75%, resulted in the next lowest recall-vs.-precision rates.

The exhaustive case using 36 rotated templates provided a better curve than the eigen approach, as expected, but overall was short of the curves of the reconstruction-base methods.

The reconstruction-based methods provided better performance curves and good rates for both recall and precision. The reconstruction based method using FMT provided better results than the reconstruction based method using bispectrum and LPM, both with five slices and ten slices. Reconstruction based method using bispectrum with twenty slices and LPM reaches the performance of the reconstruction based method using FMT, which tells us that choosing the best slice, in a more efficient way, would result in the best performance for this test group.

5.4.5 Performance Test on Using Extra Templates

In this section, the improvement that is caused by using extra template images will be presented and studied. Using extra templates, in this context, means that for one target type, more than one template image is used. The extra template should not simply be a rotated version of the template image as in the case of exhaustive approach or eigen approach, but a different image of the same target type.

5.4.5.1 Test Data for the Performance Test on Using Extra Templates

The data used for the performance test on using extra templates is almost the same as the data for the third performance comparison test, i.e. Section 5.4.4, with one difference, which is the extra template image. This template image is presented below in Figure 5.103.

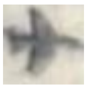


Figure 5.103: Second template image for the performance comparison test for multiple templates

5.4.5.2 Recall vs. Precision Graph for the Performance Test on Using Extra Templates

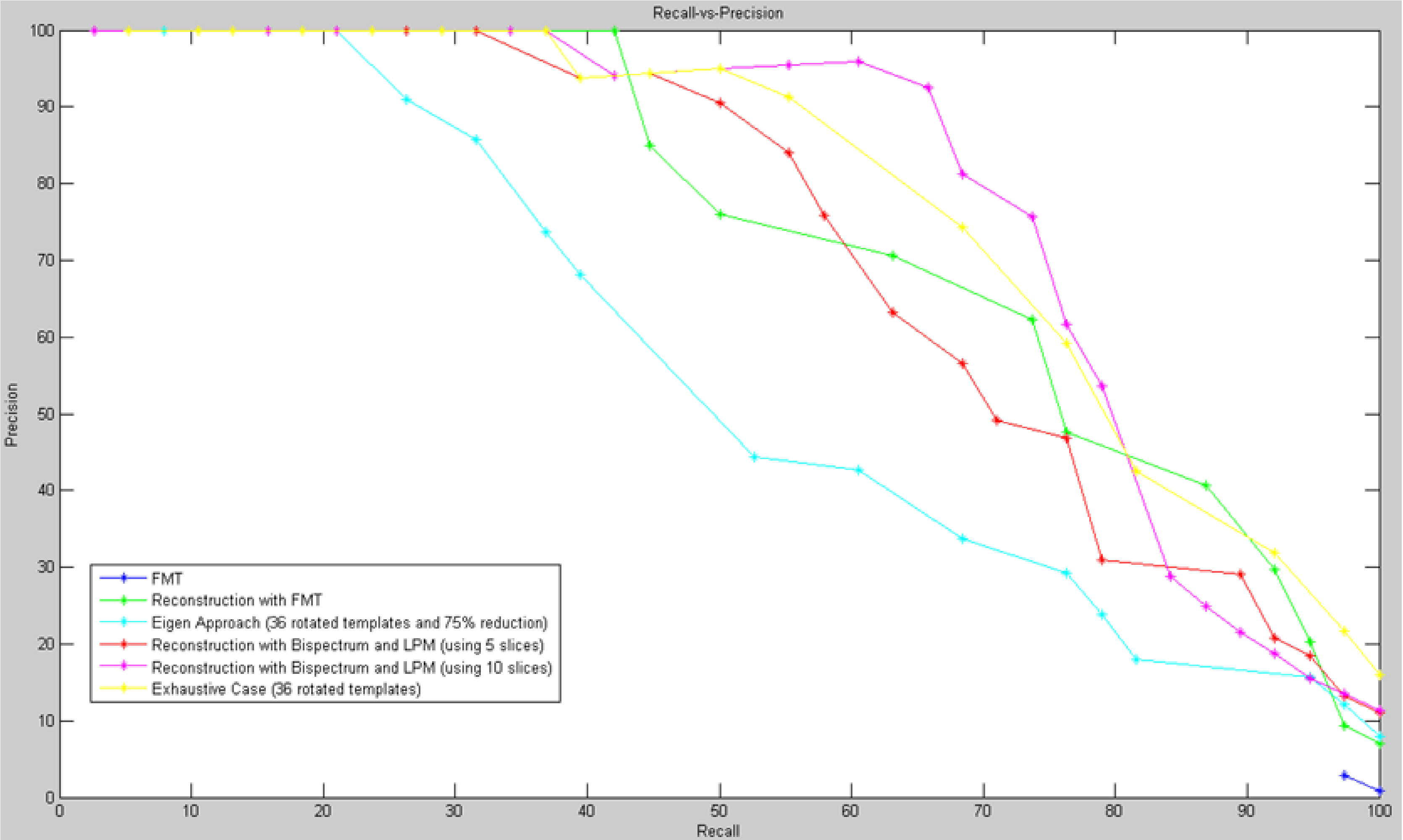


Figure 5.104: Results for the performance test on using extra templates

5.4.4.3 Results of the Performance Test on Using Extra Templates

Comparing Figure 5.104 with Figure 5.102, it can be seen that as expected, using an extra template image provided better performance curves in general, for almost every method used in this study.

It can also be observed that only the performance of the reconstruction-based method using FMT worsens with the use of the extra template image in Figure 5.103. This decrease in recall vs. precision rates is hard to explain, but would probably not exist in a test made with a larger amount of data.

Overall, it can be said that the so-called exhaustive case and the reconstruction-based method using bispectrum with ten slices and LPM provided the best performance curves for middle rates of recall; and the exhaustive case and reconstruction-based method using FMT provided the best results for high recall rates.

5.4.6 Performance Test on using Different Numbers of Rotated Templates

As mentioned earlier, and can be observed in the recall vs. precision graphs presented up-to this point, both the eigen approach and the so-called exhaustive case in this study use a total number of 36 rotated templates for each template. Larger numbers are not used for feasibility. In this section, the performance rates for the so-called exhaustive case will be presented and analyzed for different numbers of rotated templates. It is expected that a number of 36 rotated templates will provide a reasonable performance curve.

5.4.6.1 Test Data for the Performance Test on using Different Numbers of Rotated Templates

The test data used for this performance test is the same as the test data in Section 5.4.4.1.

5.4.6.2 Recall vs. Precision Graphs for the Performance Test on using Different Numbers of Rotated Templates

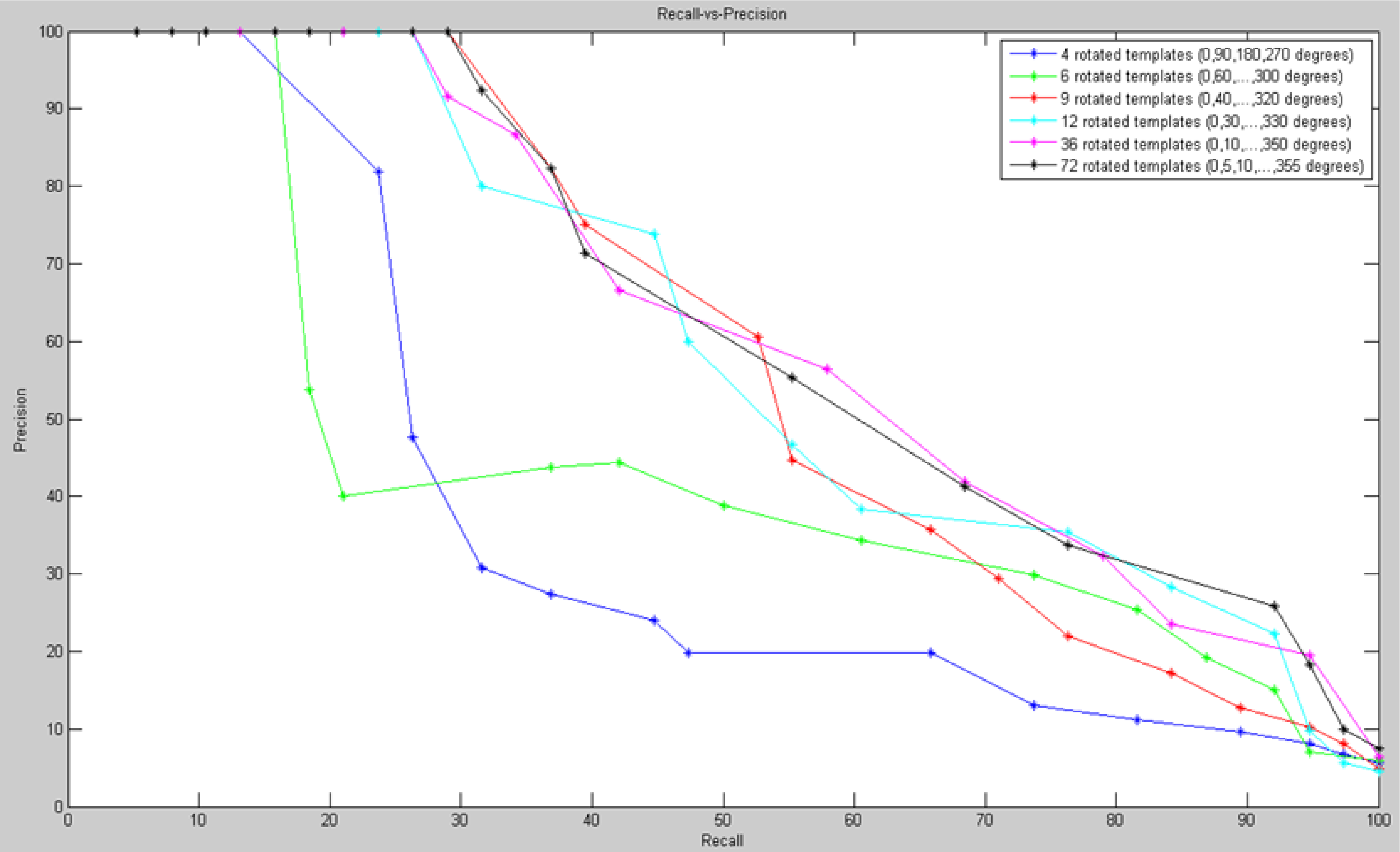


Figure 5.105: Results for the performance test on rotation

5.4.6.3 Results for the Performance Test on using Different Numbers of Rotated Templates

As expected, there is a general tendency to reach higher recall vs. precision rates as the number of rotated templates is increased. The increase in recall vs. precision rates is more pronounced between lower numbers of rotated templates, but gets harder to observe as the number of rotated templates is increased. 36 rotated templates, which is used in this study in performance tests, is a good choice both for retaining at least some feasibility and, as can also be seen from Figure 5.105, for performance.

5.4.7 Performance Test on Airports

As we have focused mostly on targets that are aircrafts, a valid question would be whether the same methods can be used to detect airports in a much lower resolution, and hence develop a pyramidal approach, in which airports are detected first, and then the aircrafts are searched within the high resolution version of the localized airport.

Unfortunately, the answer is mostly no. The main reason for this, which will be explained in Section 5.4.7.3, can be better understood after a performance test on airport detection and localization is conducted.

5.4.7.1 Test Data for the Performance Test on Airports

Given below in Figure 5.106 is a template image for the performance test on airports. The template image is followed by the test images in Figure 5.107 to Figure 5.113.

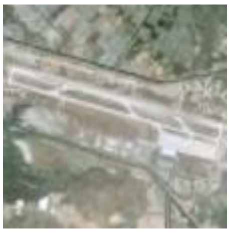


Figure 5.106: Template image for the performance test on airports

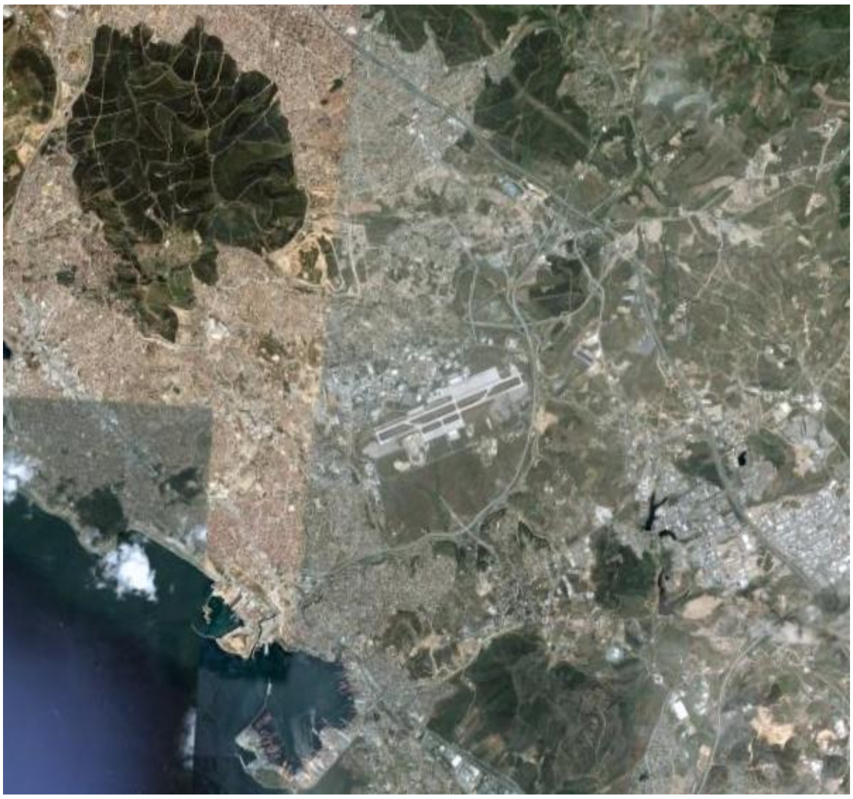
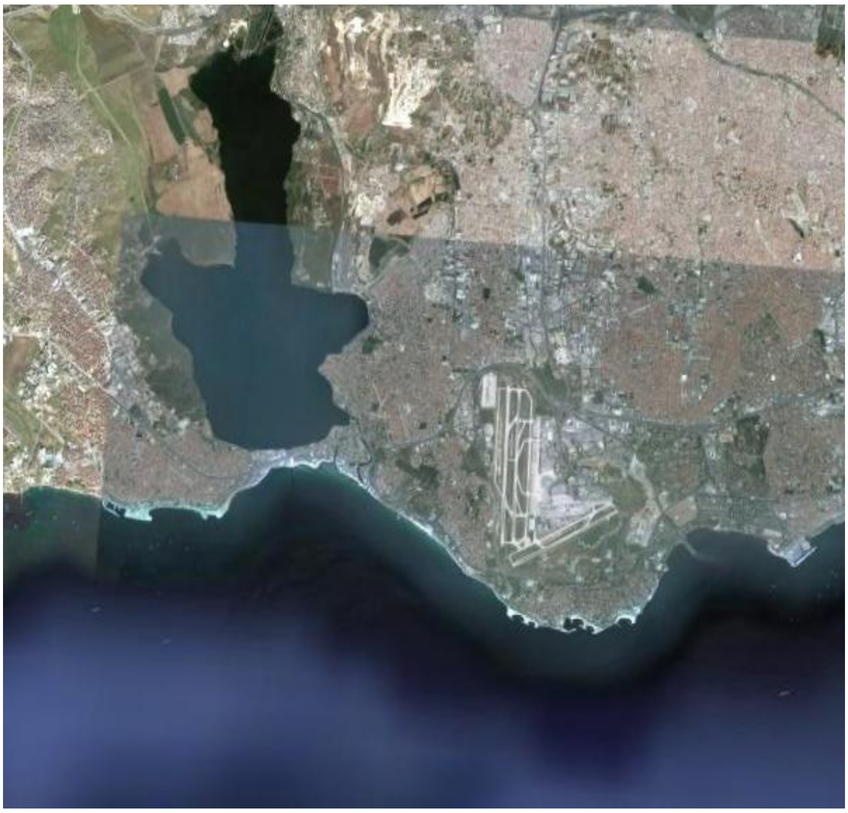


Figure 5.107: First test image for the performance test on airports



,

Figure 5.108: Second test image for the performance test on airports

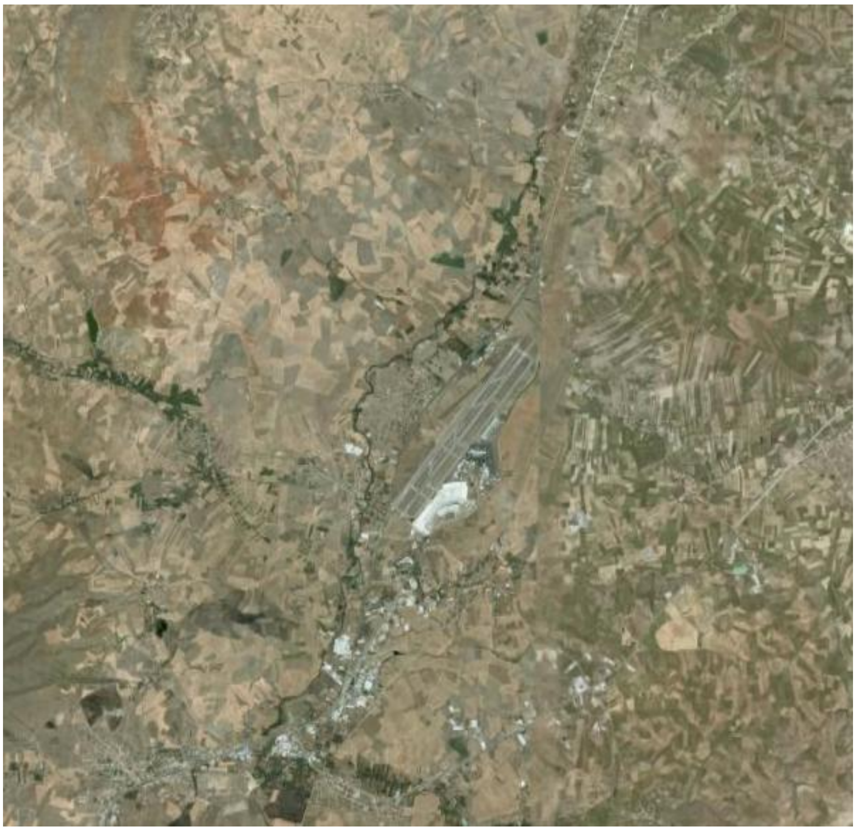


Figure 5.109: Third test image for the performance test on airports

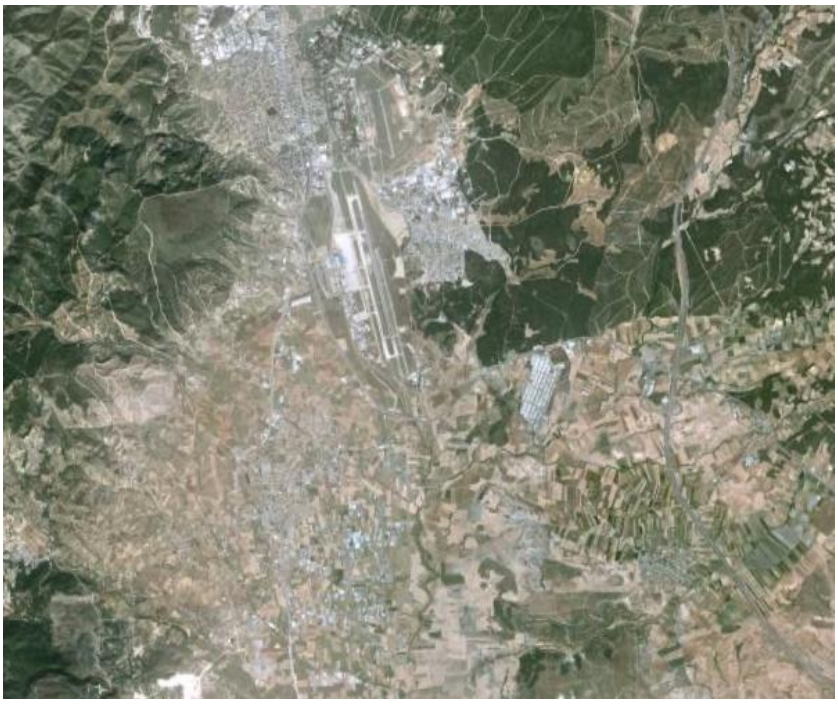


Figure 5.110: Fourth test image for the performance test on airports

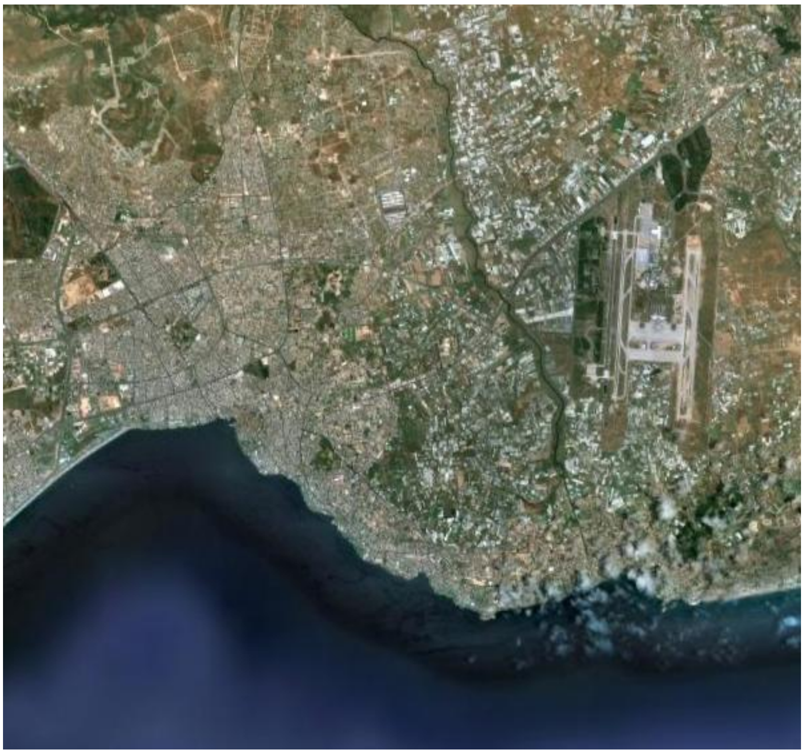


Figure 5.111: Fifth test image for the performance test on airports

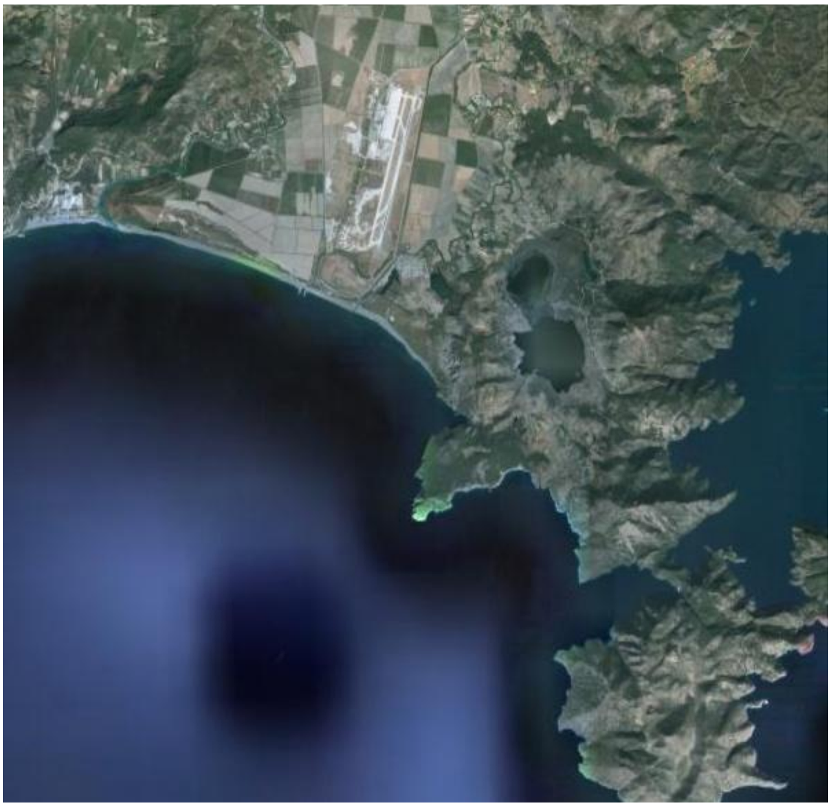


Figure 5.112: Sixth test image for the performance test on airports

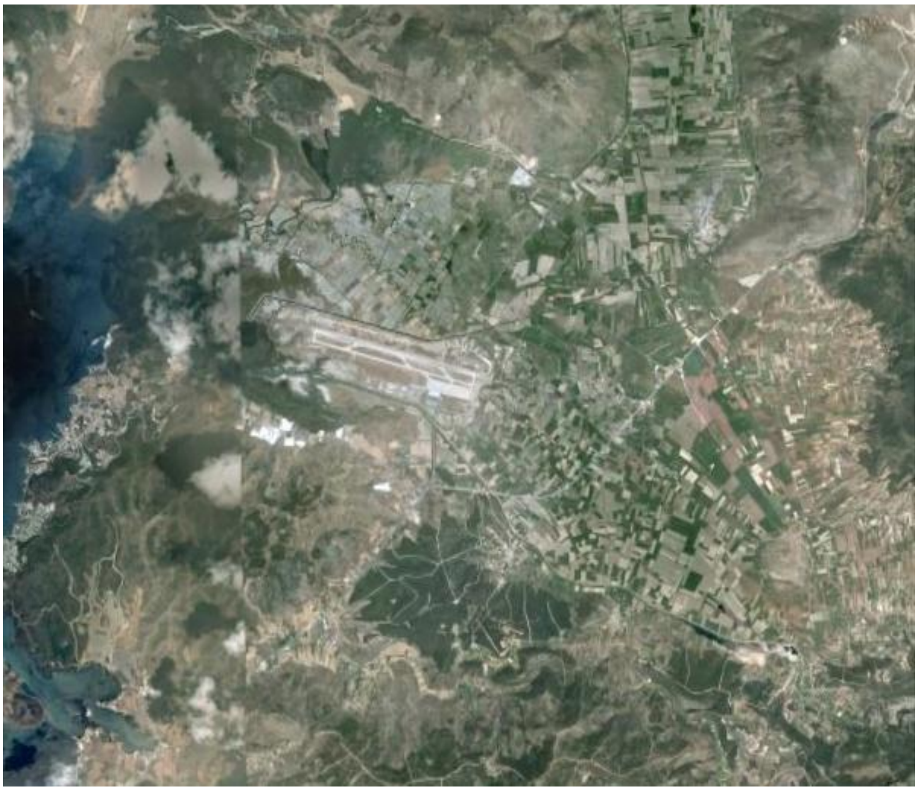


Figure 5.113: Seventh test image for the performance test on airports

5.4.7.2 Recall vs. Precision Graphs for the Performance Test on Airports

Given below in Figure 5.114 are the performance curves of the methods used in this study in terms of recall versus precision, and for the test data given in section 5.4.7.1.

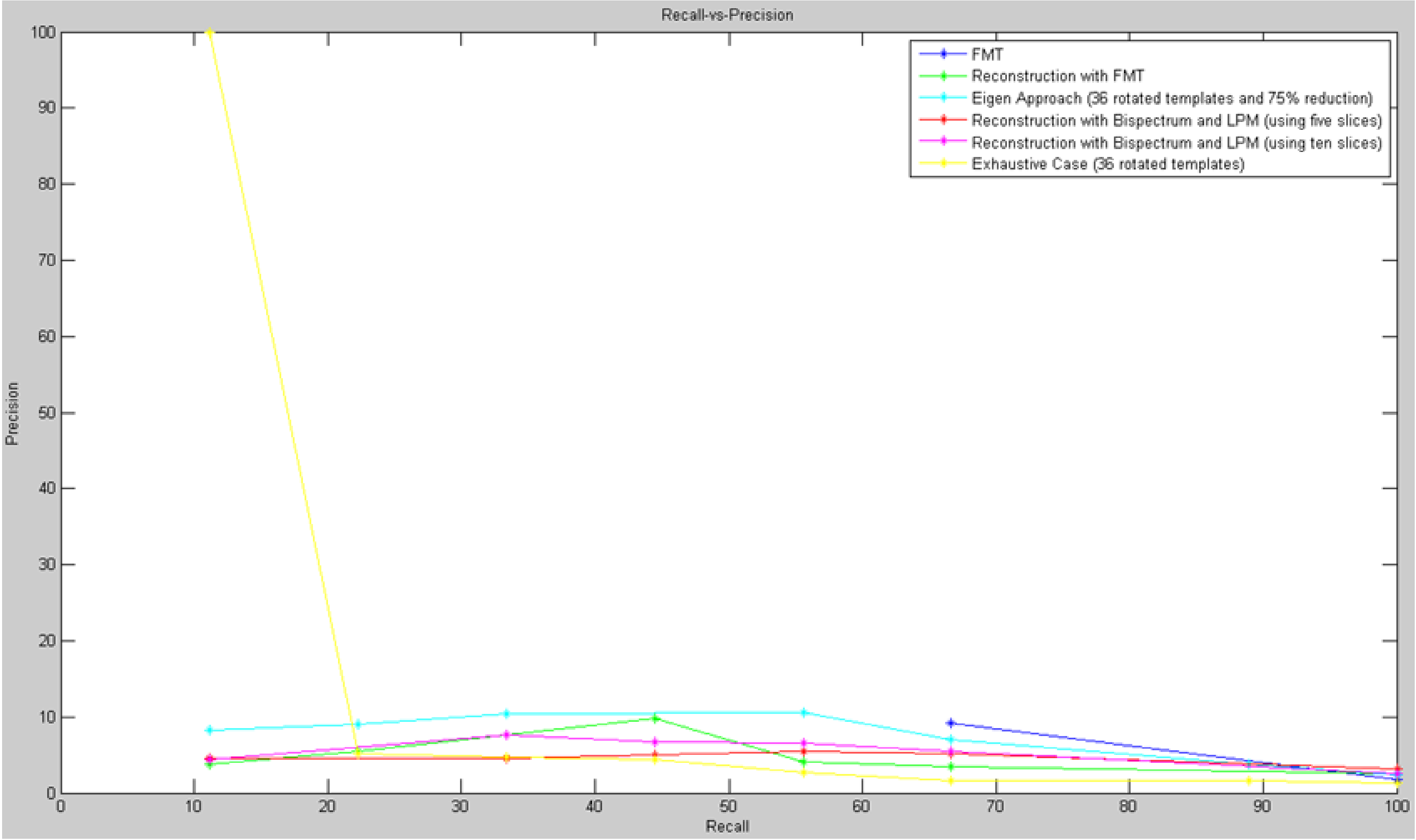


Figure 5.114: Results for the performance test on airports

5.4.7.3 Results for the Performance Test on Airports

As can be seen from Figure 5.114, all the methods used in this study performed poorly on the selected airport data. This result is not caused by the test data per se, but rather is a cause of the overall structures of airports.

Although each and every airport has a structure that is similar, namely contains long stretches of a single color which are the runways with intersections along; they are greatly varying from the perspective of template matching. The runways and their intersections are all distinct from each other in each airport. This can also be observed from Figures 5.107 to 5.113. These variances makes template matching a poor technique for airport detection compared to some other approaches such as detection by shape or texture.

5.4.8 Performance Test on Helipads

In this section, the methods used in this study are used to detect and localize a different type of target. Helipads are chosen for their importance and universality. Also, helipads are not as greatly varying from the perspective of template matching as airports are. There are three general types of helipads. The most commonly known “H” symbol, the “+” symbol used in the helipads of the hospitals, and a circular area, which can be observed in Figure 5.116. In this performance test, the helipads containing the “H” symbol will be used as test data.

5.4.8.1 Test Data for the Performance Test on Helipads

Given below is the test data for the performance test on helipads. One template image is used to detect twelve targets in nine images.

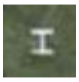


Figure 5.115: Template image for the performance test on helipads

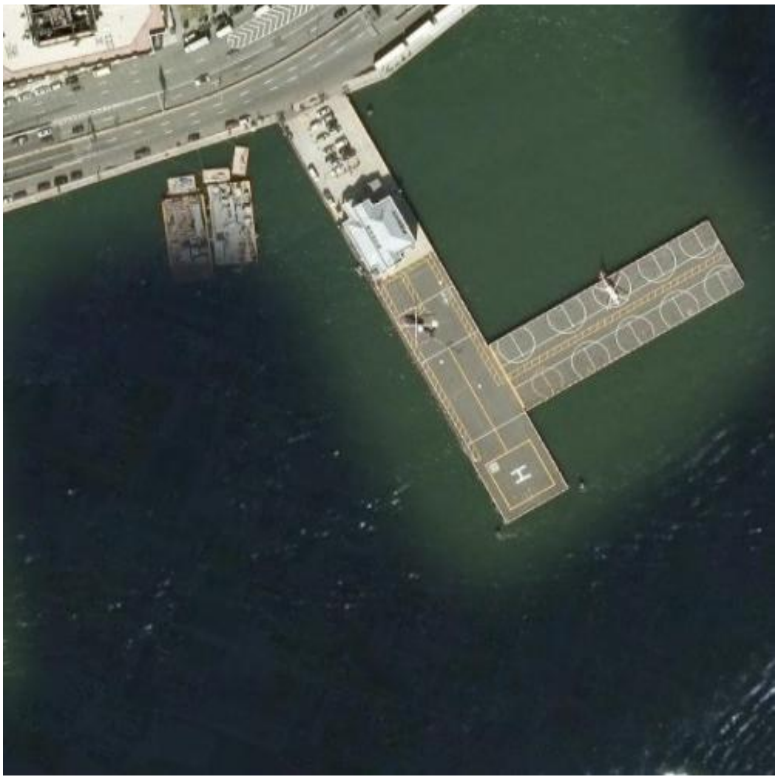


Figure 5.116: First test image for the performance test on helipads

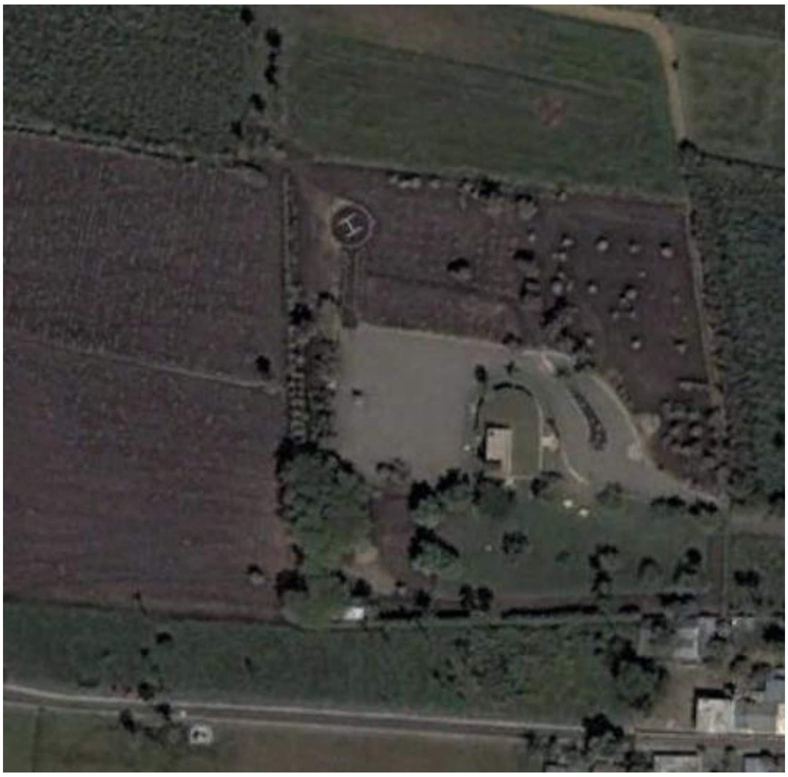


Figure 5.117: Second test image for the performance test on helipads

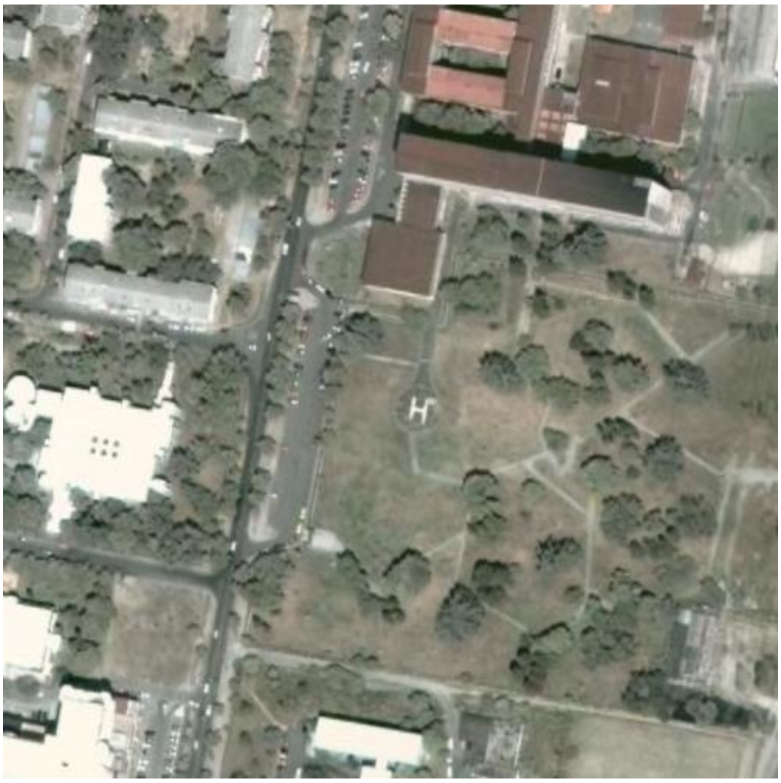


Figure 5.118: Three test image for the performance test on helipads

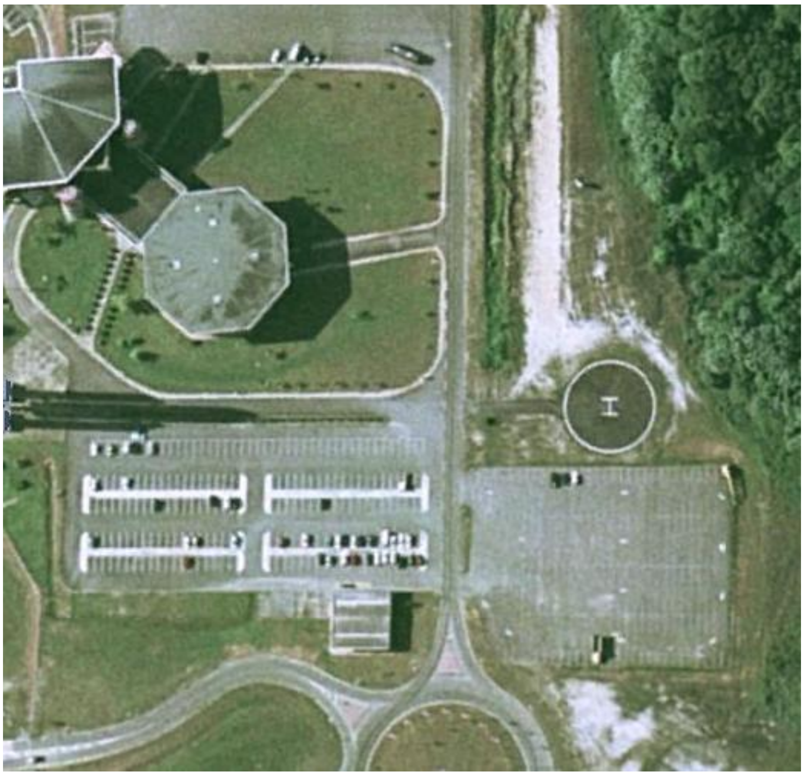


Figure 5.119: Fourth test image for the performance test on helipads

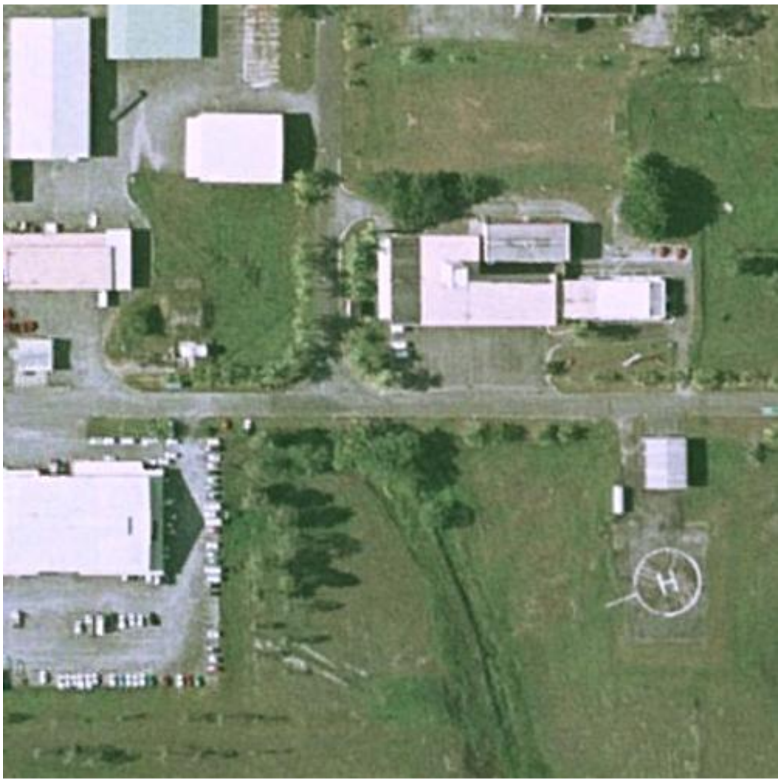


Figure 5.120: Fifth test image for the performance test on helipads

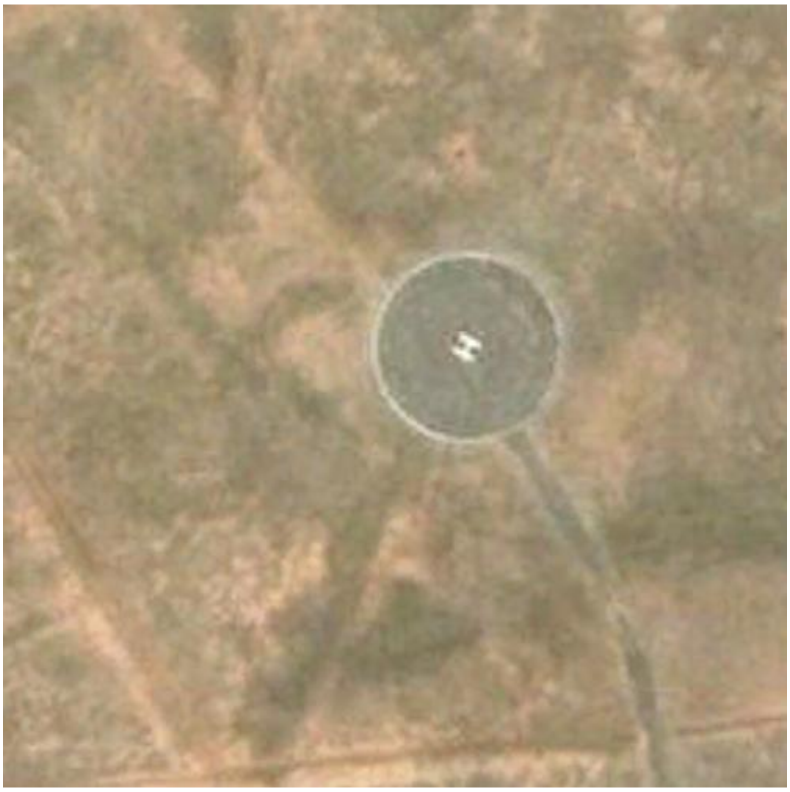


Figure 5.121: Sixth test image for the performance test on helipads

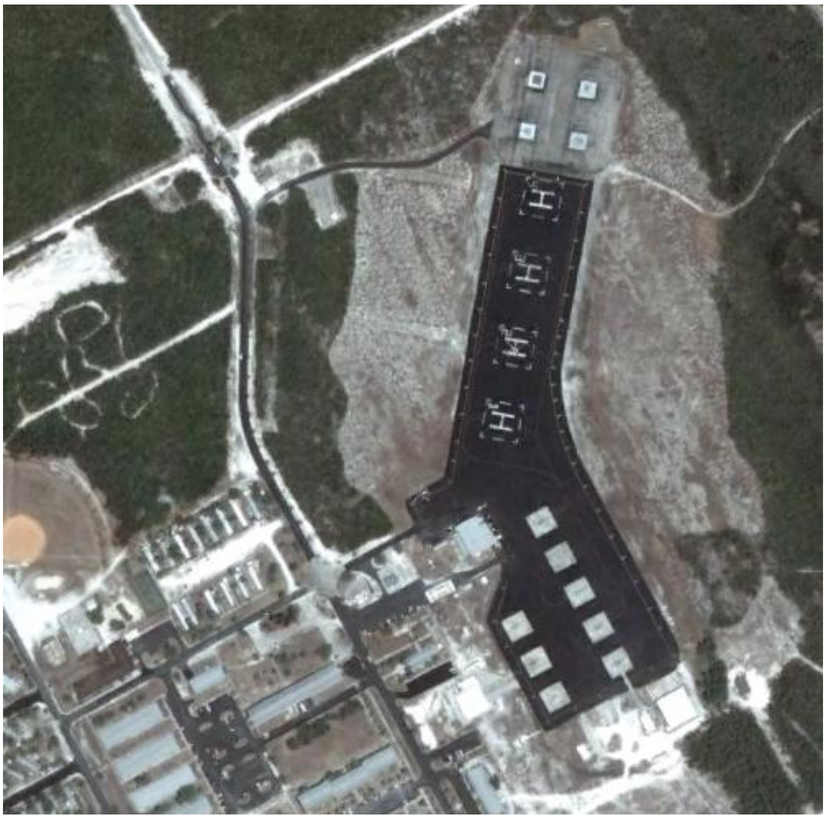


Figure 5.122: Seventh test image for the performance test on helipads

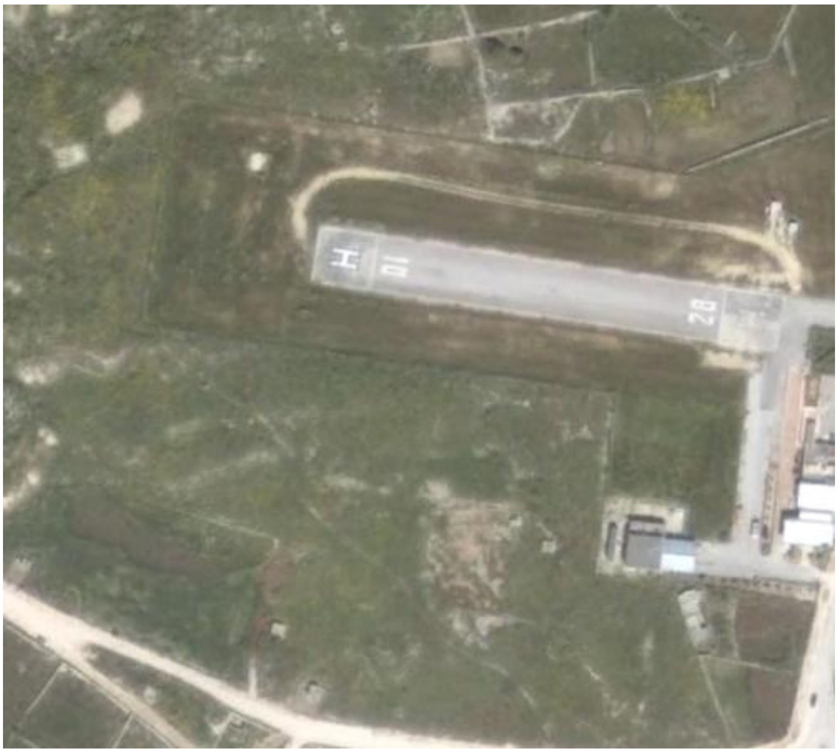


Figure 5.123: Eighth test image for the performance test on helipads

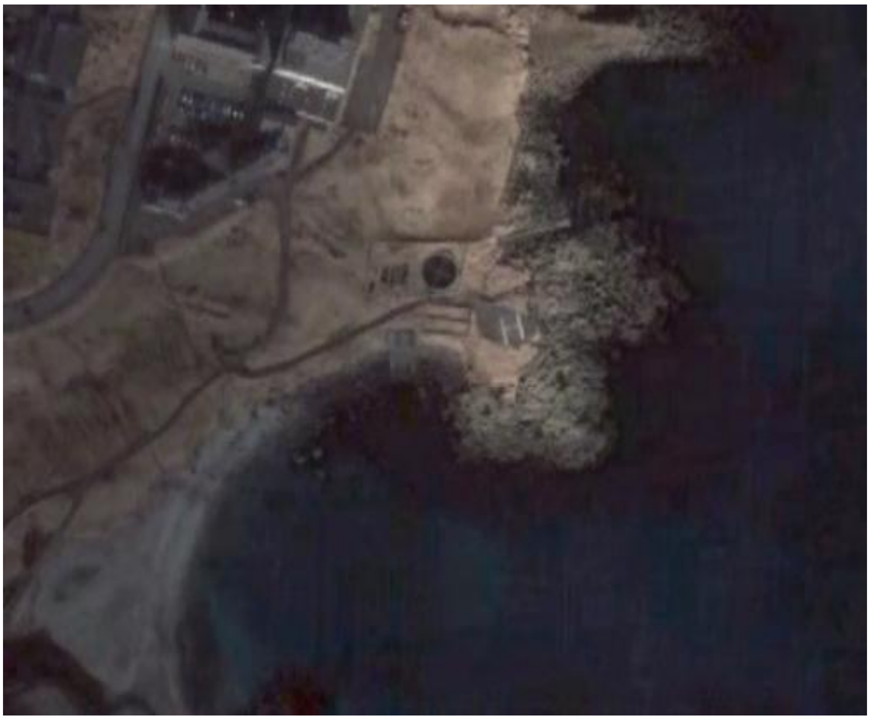


Figure 5.124: Ninth test image for the performance test on helipads

5.4.8.2 Recall vs. Precision Graphs for the Performance Test on Helipads

Given below in Figure 5.125 are the performance curves of the methods used in this study for the test data of helipads.

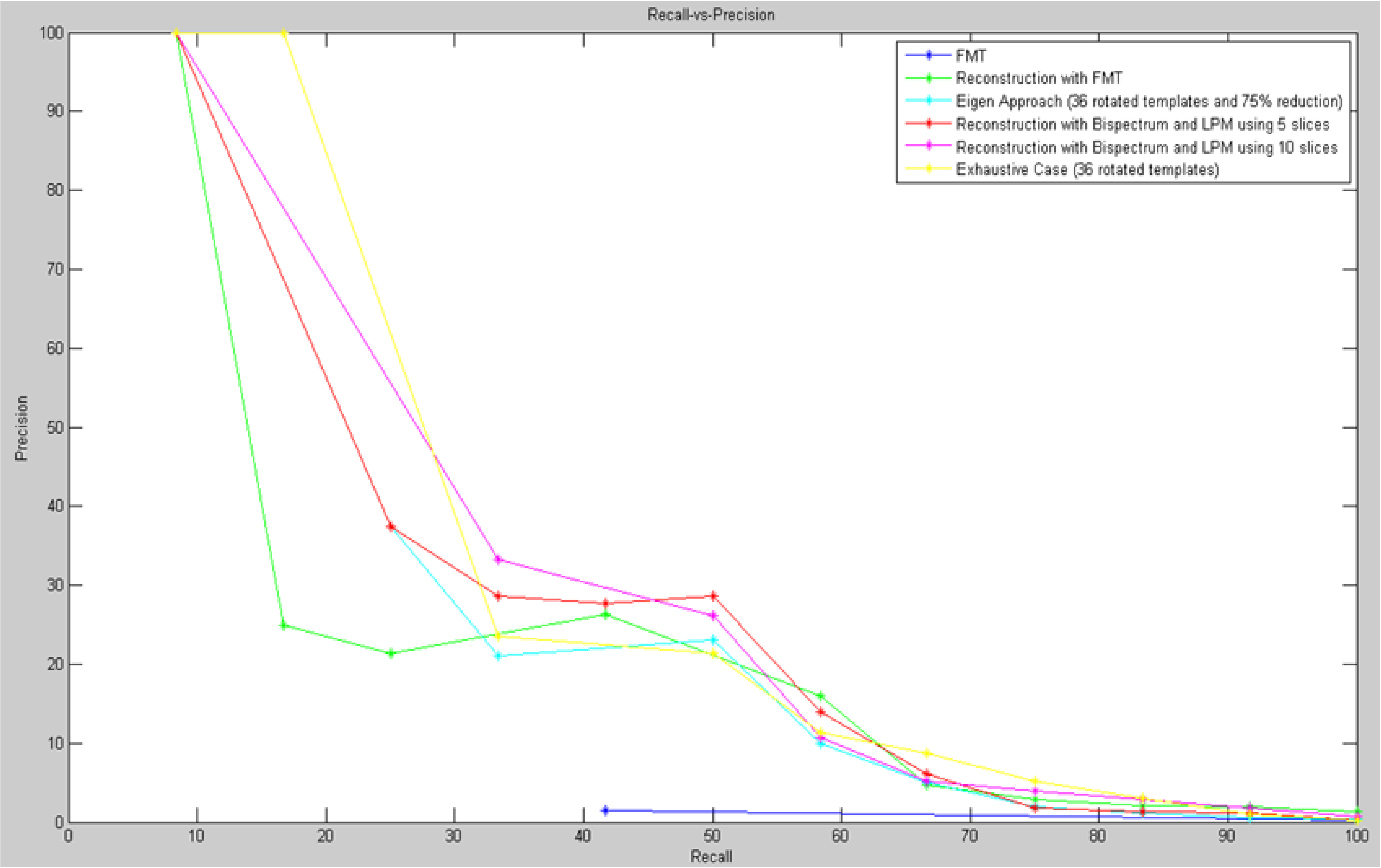


Figure 5.125: Results for the performance test on helipads

5.4.8.3 Results for the Performance Test on Helipads

As can be observed from Figure 5.125 the recall vs. precision rates are somewhat low for this test group. Especially for high recall rates, very low precision rates were obtained, which means that the false alarm rate gets very high. This is caused by the test data used, which was presented in Section 5.4.7.1, and which is particularly hard to detect and localize.

The method using FMT has provided a performance curve which is stuck at low precision rates.

All the other methods provided recall vs. precision rates which were similar. The exhaustive case with the translation fix and the RST-invariant reconstruction-based method using bispectrum and LPM provided the better results, but even these performances are poor.

5.4.9 Performance Test on Buildings

A performance test on building detection and localization will be performed in this section. Building detection is an important topic for remote sensing and object detection and localization, and it has a large number of civilian applications ranging from map updating to land use analysis and city modeling.

5.4.9.1 Test Data for the Performance Test on Buildings

A building with a square shape is chosen as the template image and two test images which contain the target building in the template image, with empty spaces between, were chosen to ensure that more than one target is not taken as detected as the template image is rotated, since all the RST invariant methods contain LPM, which is a circular operator. Hence, more realistic performance curves can be obtained from such a selection instead of choosing a rectangular building as the template image and test images that contain crammed clusters of targets.

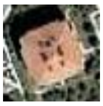


Figure 5.126: Template image for the performance test on buildings

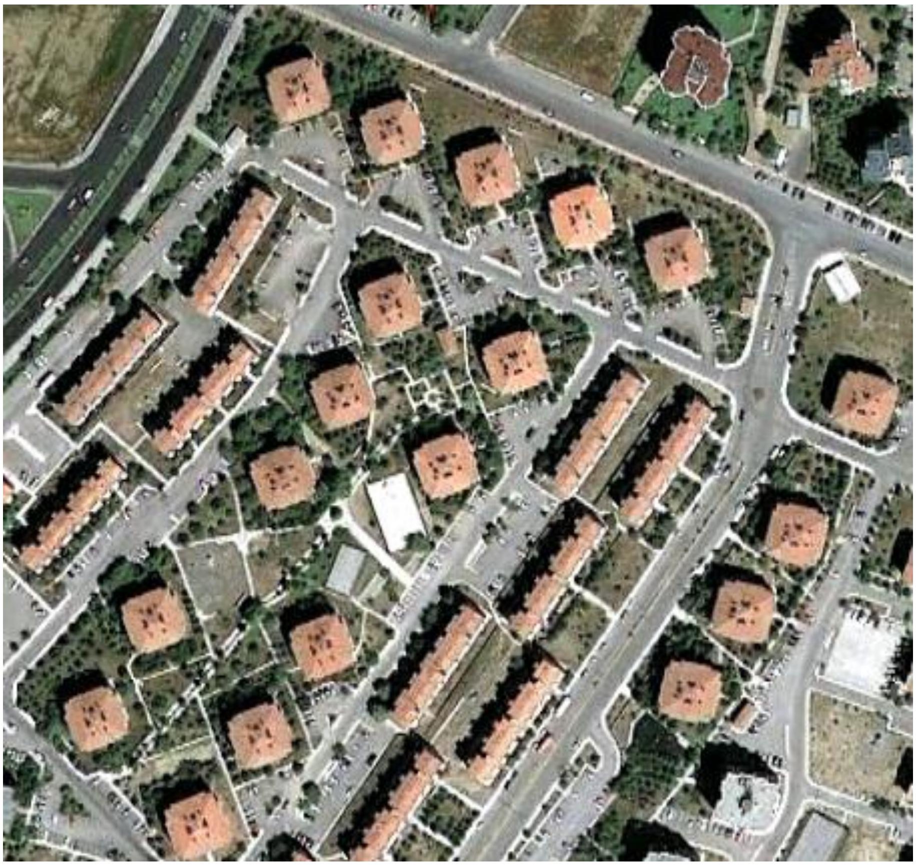


Figure 5.127: First test image for the performance test on buildings (size is reduced to fit in page)

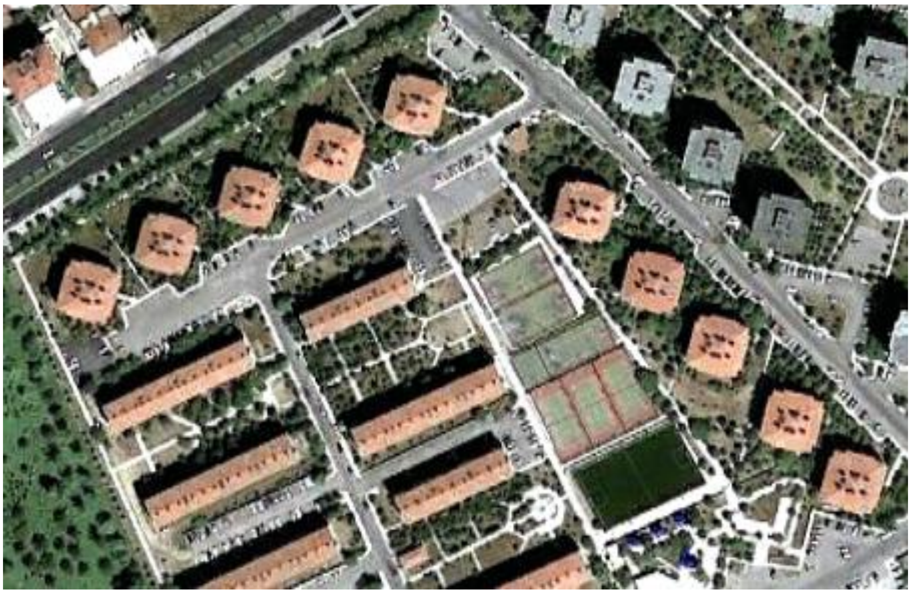


Figure 5.128: Second test image for the performance test on buildings (size is reduced to fit in page)

5.4.9.2 Recall vs. Precision Graphs for the Performance Test on Buildings

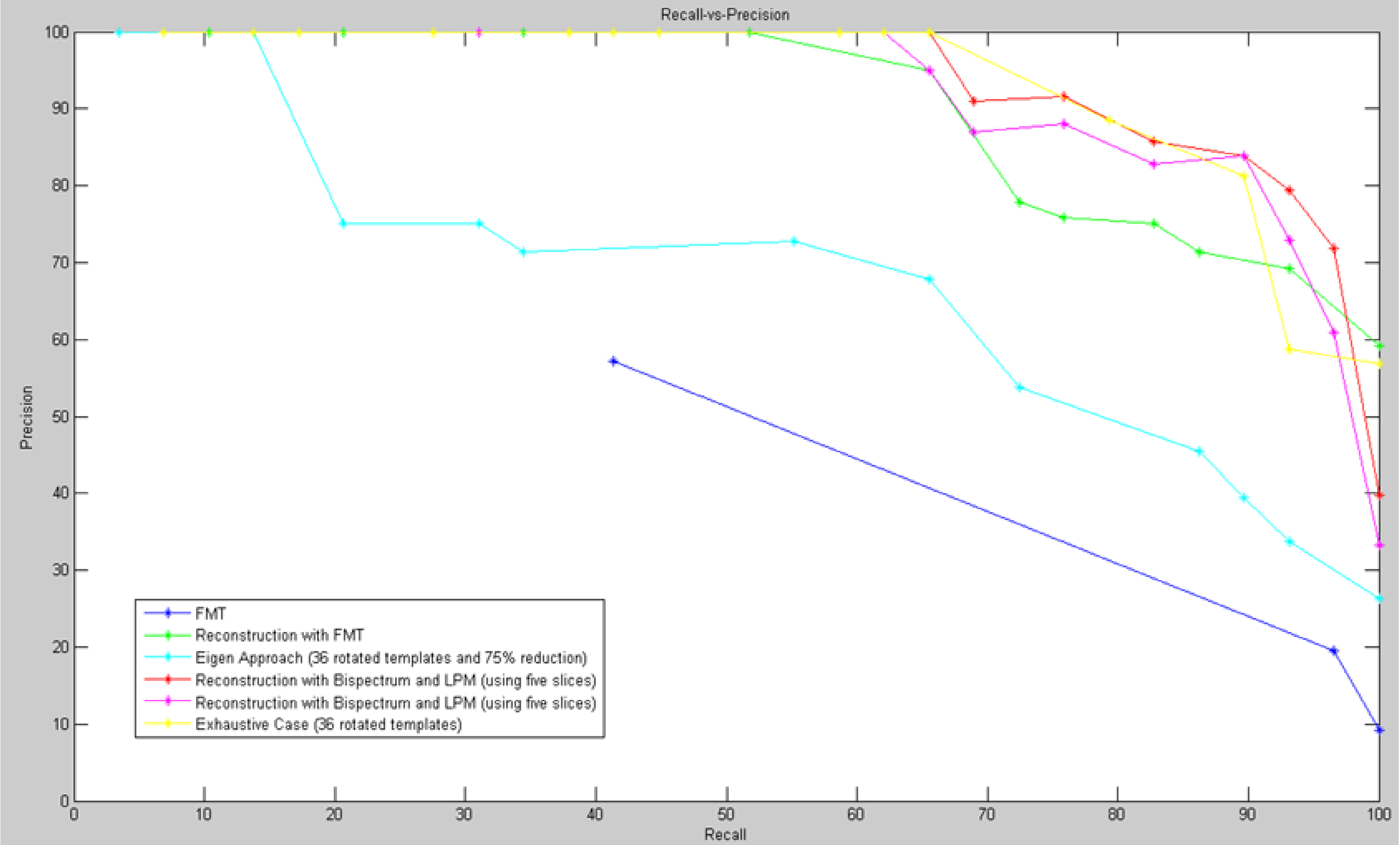


Figure 5.129: Results for the performance test on buildings

5.4.9.3 Results for the Performance Test on Buildings

As can be observed from Figure 5.129, the method involving transforming to FMT domain provided the worst performance curve for the given test data.

Eigen approach using 75% eigenvalue reduction provided the next worse performance curve in terms of recall vs. precision.

Reconstruction based RST invariant methods and the so-called exhaustive case in this study performed the best results for this test data.

While the exhaustive case has resulted in the highest precision for the 100% recall rate, overall, the performance curve of the reconstruction-based method using bispectrum and LPM competes with the exhaustive case to be the leading method in performance for this test group.

5.4.10 Performance Test on Cars

A performance on test on another type of target will be conducted in this section. The target selected was a car. One template image containing a specific car with specific coloring is used. There are four test images containing a total number of 163 cars, 11 of which are the same as the target in the template image.

5.4.10.1 Test Data for the Performance Test on Cars

Given below are the template image and the four test images for the performance test on cars.

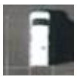


Figure 5.130: Template image for the performance test on cars

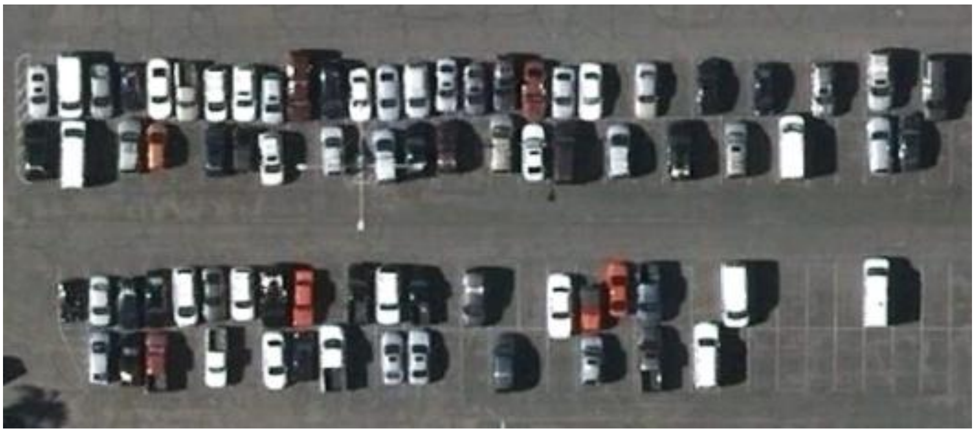


Figure 5.131: First test image for the performance test on cars

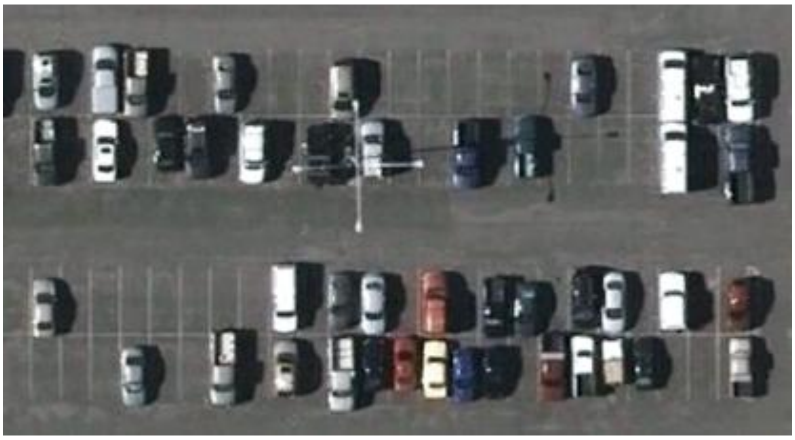


Figure 5.132: Second test image for the performance test on cars

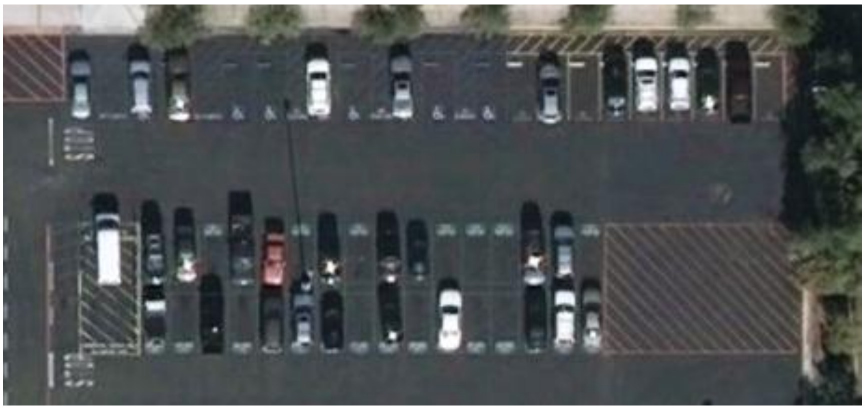


Figure 5.133: Third test image for the performance test on cars

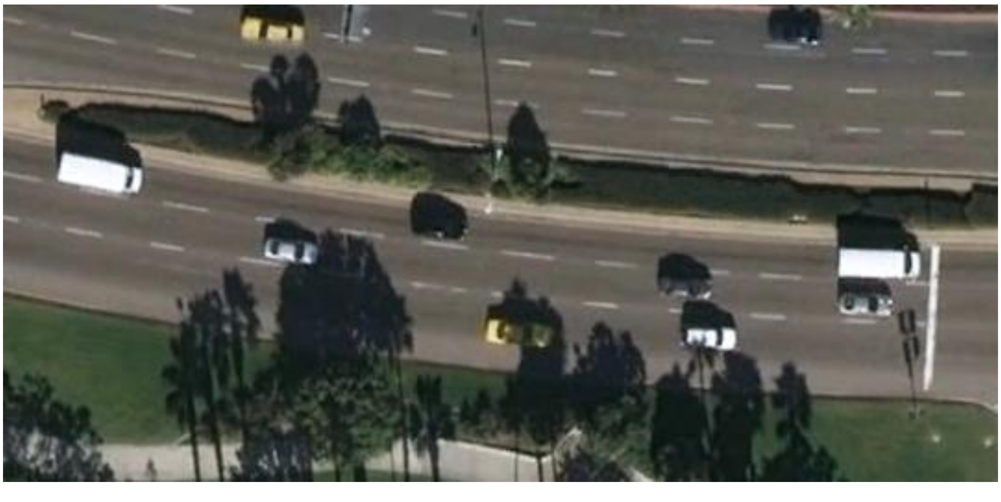


Figure 5.134: Fourth test image for the performance test on cars

5.4.10.2 Recall vs. Precision Graphs for the Performance Test on Cars

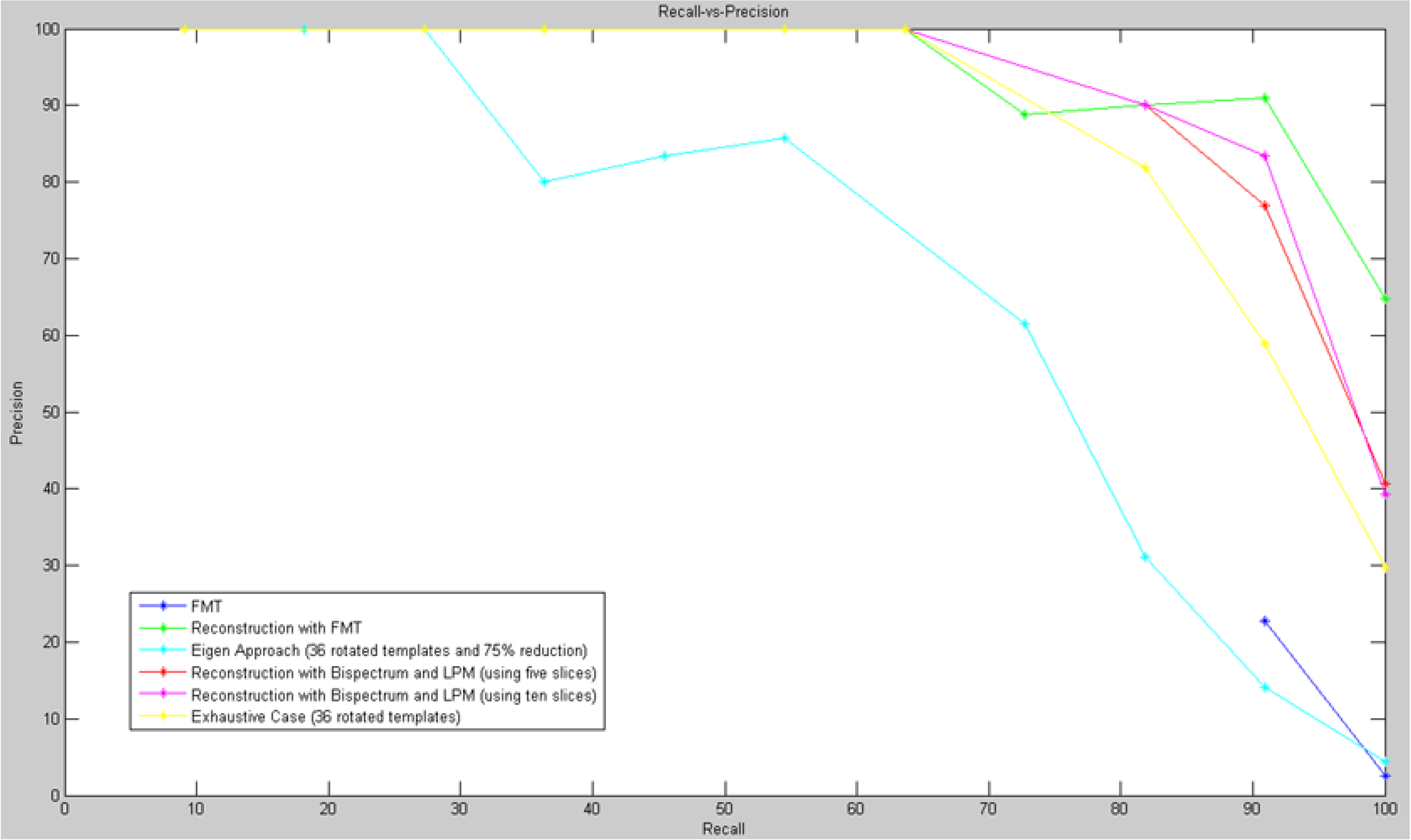


Figure 5.135: Results for the performance test on cars

5.4.10.3 Results for the Performance Test on Cars

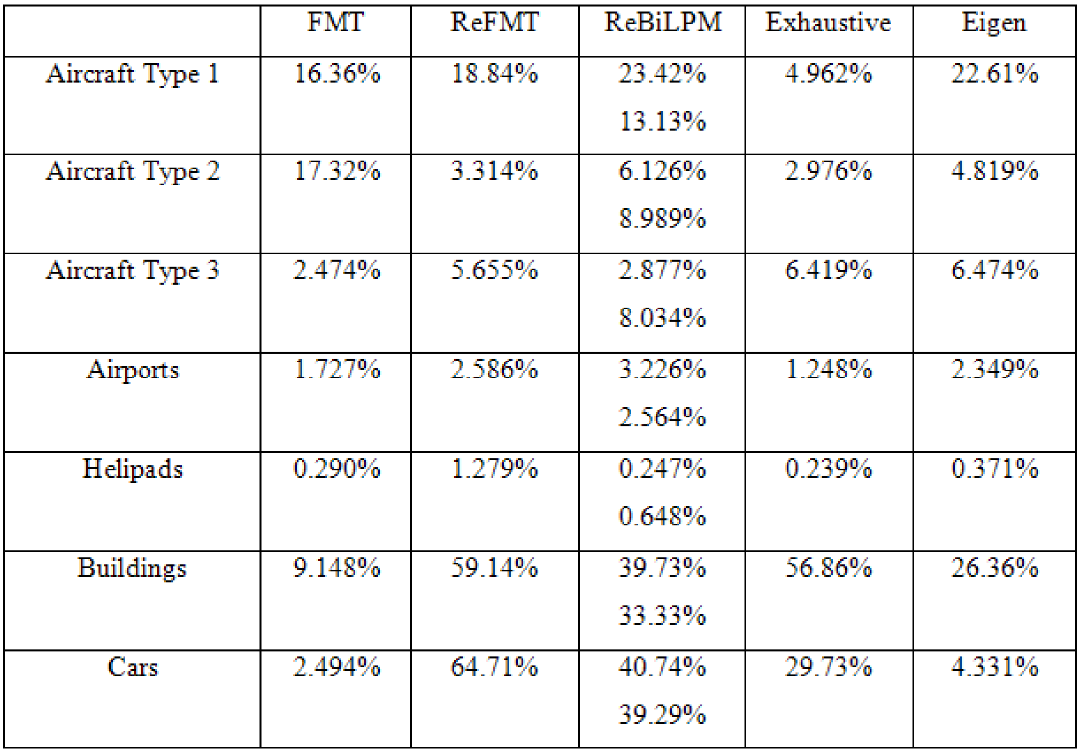
As can be observed from Figure 5.135, the method using FMT and the eigen approach using 36 rotated templates and 75% reduction provided the worst performance curves for this test group, over all the methods used in this study.

The exhaustive case and the reconstruction-based RST invariant methods provided much better performance curves, with the reconstruction-based method using FMT providing the highest precision rate for 100% recall rate.

The obtained recall and precision rates are high for this test data, because the backgrounds are all are roads or car parking areas, which are similar to each other in terms of texture and coloring.

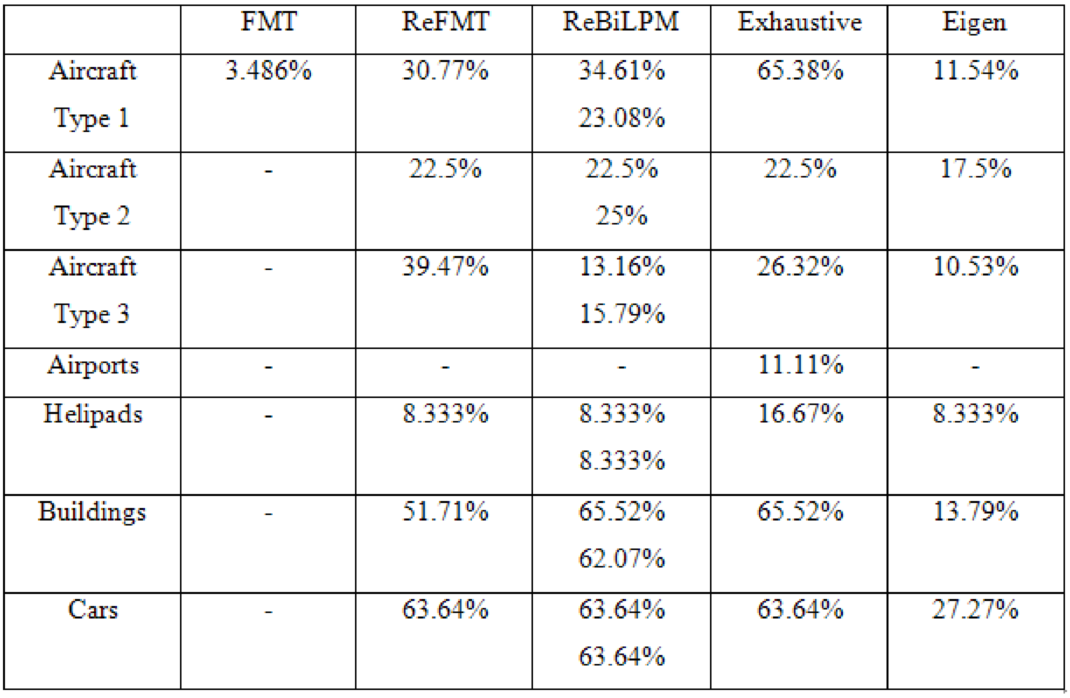
5.4.11 Overall Analysis of the Performance Tests

Table 5.1: Precision rates of the methods at 100% recall rate



	FMT	ReFMT	ReBiLPM	Exhaustive	Eigen
Aircraft Type 1	16.36%	18.84%	23.42% 13.13%	4.962%	22.61%
Aircraft Type 2	17.32%	3.314%	6.126% 8.989%	2.976%	4.819%
Aircraft Type 3	2.474%	5.655%	2.877% 8.034%	6.419%	6.474%
Airports	1.727%	2.586%	3.226% 2.564%	1.248%	2.349%
Helipads	0.290%	1.279%	0.247% 0.648%	0.239%	0.371%
Buildings	9.148%	59.14%	39.73% 33.33%	56.86%	26.36%
Cars	2.494%	64.71%	40.74% 39.29%	29.73%	4.331%

Table 5.2: Recall rates of the methods at 100% precision rate



	FMT	ReFMT	ReBiLPM	Exhaustive	Eigen
Aircraft Type 1	3.486%	30.77%	34.61% 23.08%	65.38%	11.54%
Aircraft Type 2	-	22.5%	22.5% 25%	22.5%	17.5%
Aircraft Type 3	-	39.47%	13.16% 15.79%	26.32%	10.53%
Airports	-	-	-	11.11%	-
Helipads	-	8.333%	8.333% 8.333%	16.67%	8.333%
Buildings	-	51.71%	65.52% 62.07%	65.52%	13.79%
Cars	-	63.64%	63.64% 63.64%	63.64%	27.27%

Through the performance tests conducted in this study, and as can be observed from Table 5.1 and Table 5.2 above, several important conclusions can be drawn by the analysis of the obtained recall vs. precision graphs.

As expected, Fourier-Mellin Transform, by itself, was not sufficient to obtain good performance rates. Mostly, only low precision rates could be obtained with this method. This is direct consequence of drawbacks of this approach, which were explained in Section 4.2.2.

Although the exhaustive case provided good performance rates for most the test groups, obtaining 36 images of a target with 10° rotations is generally unfeasible, unless these images are generated artificially.

From the performance rates of the eigen approach, it is deduced that either no or a lower rate eigenvalue reduction must be used to obtain good recall and precision rates unless the translation invariance vs. loss of phase information drawback can be overcome.

The reconstruction-based RST invariant methods provided the best performance curves, considering that only a single template image was used for each target type. While the method denoted as reconstruction with FMT provided a relatively good performance curve, it was observed that the method denoted as reconstruction with bispectrum and LPM did not always provide better recall vs. precision rates than this method, as expected. This is caused by the effect of the bispectrum slices. Increasing the number of bispectrum slices However, it should be noted that increasing the number of bispectrum slices does not necessarily always result in obtaining better performances, while it always results in increasing the computation time.

As mentioned earlier, and will also be mentioned in Section 6.3, future studies, a better method for choosing bispectrum slices would solve the high computation time drawback of the method using bispectrum, while enabling higher recall vs. precision rates, and hence better performances than the ones obtained in this study.

Table 5.3: The effect of using extra templates on recall rates at 100% precision rate

Recall rate at 100% precision rate	FMT	ReFMT	ReBiLPM	Exhaustive	Eigen
Aircraft Type 3	-	39.47%	13.16% 15.79%	26.32%	10.53%
Extra Template	-	42.11%	31.58% 36.84%	36.84 %	21.05%

Table 5.4: The effect of using extra templates on precision rates at 100% recall rate

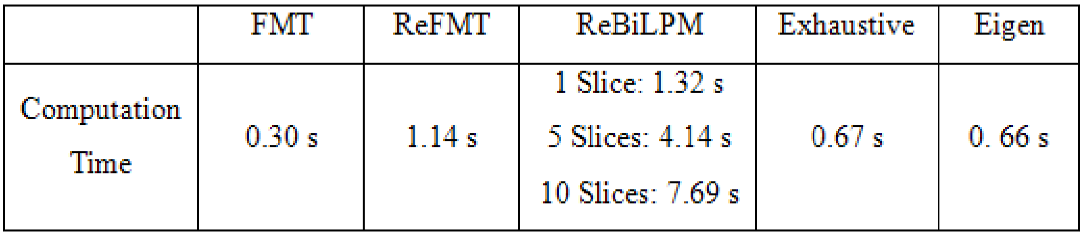
Precision rate at 100% recall rate	FMT	ReFMT	ReBiLPM	Exhaustive	Eigen
Aircraft Type 3	2.474%	5.655%	2.877% 8.034%	6.419%	6.474%
Extra Template	0.872%	6.985%	11.08% 11.34%	15.97%	8.017%

As can be observed from the tables above, using extra number of templates increases the both the recall and the precision rates for all of the methods implemented in this study, as expected.

5.5 Computation Time Analysis

Given below in Table 5.5 are the computation times for the methods implemented in this study, for a template image and a sub-image of size 60x60 pixels. The computation times are calculated in MATLAB® R2007a and on a computer with Intel® Core™2 Quad CPU @ 2.4GHz, 3.50 GB of RAM.

Table 5.5: Computation times of the methods



	FMT	ReFMT	ReBiLPM	Exhaustive	Eigen
Computation Time	0.30 s	1.14 s	1 Slice: 1.32 s 5 Slices: 4.14 s 10 Slices: 7.69 s	0.67 s	0. 66 s

As can be observed from Table 5.5, FMT is the method with the shortest computation time. This is followed by the exhaustive and the eigen approaches. Note that although the eigen approach is devised to lessen the computation time of the exhaustive approach, the improvement is negligible. Considering the decrease in recall vs. precision rates, eigen approach seems as an unnecessary method at its form in this study. The RST-invariant methods based on reconstruction have higher computation times, due to the increased number of operations and the multiple numbers of peaks used. As can be seen, using multiple numbers of slice constants results in an increase in computation time, that is not directly proportional to the number, since not all of the operations are multiplied.

5.6 Template Selection and Extraction

Template selection is an important part of this study and affects the performance of any method using template matching, drastically. Ideally, the template should be exactly the same in the scene to be recognized, but this cannot be satisfied in object detection and recognition due to background and illumination (including the angle of the sun) variations.

Although using a large number of templates is a straightforward solution to this problem, this is generally not feasible.

Then, all that can be done is to choose the templates as good as possible. Through this study, an understanding of the template selection has been gained. Some of the guidelines to keep in mind while choosing or extracting templates are as follows:

1) Keeping the object at the center of the template image:

This is mandatory in applications involving or requiring RST invariance, since the rotation of the template image and the object in the template image should be the same. Also, keeping the object at the center ensures that the object is affected minimally by transitions. In application without RST invariance, this is still good advice, since combined with the guideline two below, this ensures that the object stays in the template image completely while the image contains as little background as possible.

2) Keeping the background as small as possible in the template image:

Although it is the object in the template that is wished to be matched, the overall template is matched. The rest of the template is named loosely as background. In applications like object detection, this background varies for each matching even though the object stays pretty much the same and hence decreases the matching performance. Hence, obviously, keeping the template as small as possible while retaining the object fully, i.e. keeping the background as small as possible, results in better performance.

3) Selecting the template with the relatively simpler background:

Even though the background is kept as small as possible, the properties of the background are also very important. Ideally, the background should be as similar to the backgrounds of the targets in the test images. This is in many, if not all, cases not feasible. However, the fact that the background should not be irrelevant or far-out can easily be deduced from here.

Failing taking similar backgrounds, the background of the chosen template should be as simple as possible. In this context, simple means little or no other objects in the template image, other than the object that is being tried to be matched. The reason for this is that these objects almost always vary and a relatively simpler background can be matched more successfully.

4) Selecting the template with smaller shadows and well illuminated object:

This guideline is in parts trivial and in parts a consequence of the previous guideline. A well illuminated template is trivially better than a template with weak illumination since the information stored in the pixels are affected by the illumination and a weak illumination results in loss of information. If the angle of the light –or in our case, sun- is perpendicular to the object, the illumination distribution will be even in both sides of a symmetrical object. However, if this angle is oblique, this distribution will vary, and the more oblique the angle, the greater the variance, and the greater the shadows.

As explained previously, and will be again mentioned later on, the shadows – both self and cast- and the illumination angle are important drawbacks of using template matching for object detection and recognition. However, selecting a template with nearly perpendicular illumination and little shadow, results in better performance, as expected, since no shadow or a distinct variation of illumination coefficients are tried to be matched in such a case.

CHAPTER 6

CONCLUSIONS

6.1 Summary

In this thesis, template matching was used with RST invariance methods to achieve automatic target recognition from satellite imagery.

Template matching was studied in Chapter 2. A literature survey on template matching was presented. Similarity metrics were also studied in Chapter 2, with special emphasis on normalized cross-correlation and phase correlation, which are used in this study in great extent.

RST invariance, which has been the primary focus of this study, was studied in Chapter 3. After the literature survey on RST invariance, moment-based approaches, which were studied and discarded because of their many drawbacks, were explained. Then, the translation, rotation and scaling properties of Fourier transform were presented, followed by log-polar mapping. Fourier-Mellin transform, which is one of the methods implemented in this study for RST invariance, and is a combination of Fourier transforms and log-polar mapping, was presented next. Bispectrum and its properties of rotation, scaling and translation were presented last.

In Chapter 4 the methods used in this study were presented and studied in detail. These methods were, in order of study, and hence in order of appearance in this thesis, Fourier-Mellin transform, a reconstruction-based method using Fourier-Mellin transform and a reconstruction-based method using bispectrum and log-polar mapping. The drawbacks of the RST invariant approach was explained after the common points that should be heeded. A method involving eigenvalue reduction, named in this study as the eigen approach, was presented next, since it was used in this study for performance comparison purposes.

In Chapter 5, the tests on variations and parameters was presented and studied, followed by the tests of performance. The tests on whether to use overlapping or non-overlapping blocks, and what is the best overlapping ratio was presented first, which were concluded for using overlapping blocks, with the overlap ratio set at 3/4. These tests were followed by the windowing tests, in which using Hanning windowing were concluded with the window size set as proportional to the template image size. The tests on using artificial or real background were concluded in not using an artificial background. The slice analysis of the method involving bispectrum was the last of the tests on variations and parameters. The affect of slice choice on the performance was shown and an approach using multiple slices and for each sub-image, choosing the slice with the highest correlation result among these was developed.

Also in Chapter 5, some performance tests were presented in terms of recall versus precision. The performances of the methods were compared and analyzed for each group of test data, followed by the overall analysis of performance.

6.2 Conclusions

Target detection, recognition and localization from satellite and air imagery have been and will continue to be an important topic of research. ATR has many military and civilian applications. These applications range from GPS-enabled firing systems to homeland security and surveillance, from land use analysis to map updating and city planning. RST invariance is a must for most of the targets to be recognized in many of these applications, since using a template image for each scale and rotation angle is not feasible. However, the conclusive research on RST invariant target recognition and localization is still limited due to many drawbacks faced.

RST invariant automatic target recognition is realized in this thesis using template matching. A variety of RST invariance methods was investigated, implemented and tested for performance for a variety of target types. Important and conclusive deductions have been made through the extensive performance tests conducted.

It was realized that a simple method of transforming the template images and the sub-images to the Fourier-Mellin domain generally doesn’t result in a good performance and is limited to low precision rates. The reason for this is that all the images are transformed into a similar form, due to the losses of phase information, and the dynamic range is very small. Reconstruction-based method, which is devised to overcome this drawback, and which detects the RST parameters and uses these to reconstruct the sub-image to a form in which the target in the sub-image will have the same scale and rotational alignment as the target in the template image, resulted in better performance curves. Yet the phase information was still lost to some extent. For this reason, to reach better performance rates, this method was replaced by another. This new method which works similarly but uses bispectrum instead of FT provided better results, but another problem, namely the bispectrum slice choice, arose. It was realized that the computationally costly approach of using a multiple number of slices for each sub-image limited the potential of this method. A novel approach to solve this drawback would probably make this method reach the highest performance rates among the methods studied and implemented in this study.

Despite the faced drawbacks, RST-invariant ATR using template matching was achieved in this study and tested for performance on different types of targets. Performance rates reaching or suppressing the exhaustive case using 36 rotated templates could be obtained in nearly all of these performance tests. Of particular interest are the precision rates when a recall rate of 100% is reached. This recall rate corresponds to the threshold that results in each and every target being recognized, in other words when there are no missed targets. This is especially important for military applications. In most of the performance tests conducted in this study, precision rates of at least 10% or 5% could be reached for a recall rate of 100%. This means that checking just 10 times or 20 times more sub-images than the number of targets will be sufficient for a human operator to have all the targets recognized without having any false alarms. This is a rate that has not been obtained in the literature up to now and would be acceptable, if not desirable, for many applications.

An understanding on the nature of template matching was also obtained through this study. Template selection and extraction is a fundamental and important part of each method applying template matching. As can be seen from the performance tests, some target types are more suitable for template matching, while others, such as airports, play into the drawbacks of such an approach.

RST-invariant ATR using template matching is a research topic that is still open to improvement. Studies that are built on this thesis can achieve greater performance rates, and higher real-life applicability, by building on the methods used in this study and solving the few drawbacks that are brought into light by it.

6.3 Future Studies

The basic drawback of the approach of this study was that the target does not appear the same in the test images as in the templates. One of the important factors that cause this deviation from the template images is illumination. Both cast shadow and self shadow cause changes in the illumination distribution of the target. A possible future work would be to use a shadow detection and compensation system with the methods in this study to make the methods not only RST invariant but also, in a way, shadow invariant.

In a very recent study, Matungka, Zheng and Ewing [62] have solved the two important problems of log-polar mapping, oversampling and bias, which were also mentioned in this thesis. Their novel method, named as adaptive polar transform, outperformed the conventional LPM and provided both unbiased matching and lesser computation time. A possible future study would be to incorporate adaptive polar transform into the methods used in this study, in the place of LPM.

As observed and explained through the presented tests in this study, the choice of the bispectrum slice affects the results of the last method, the reconstruction-based method using bispectrum and LPM. Instead of the straightforward and computationally highly inefficient solution applied in this study, namely using a multiple number of bispectrum slices for each sub-image and taking the highest result, a novel method to solve this drawback in a computationally efficient way can be developed to obtain better results in much less time.

Instead of the sliding-window approach used in this study, which has a high computation cost, a more efficient sub-window search can be adopted. Lampert *et al* [63] have developed such a method which approaches the search as an optimization problem that can be applied to any quality function, and uses a global branch-and-bound search. They have achieved fast object localization while retaining global optimality in the search.

REFERENCES

[1] B. Bhanu, D. E. Dudgeon, E. G. Zelnio, A. Rosenfeld, D. Casasent, and I. S. Reed, "Guest editorial introduction to the special issue on automatic target detection and recognition*", IEEE Transactions on Image Processing*, Vol. 6, Issue 1, Jan. 1997, pp. 1-6

[2] M. Khosravi and R. W. Schafer**,** "Template matching based on a grayscale hit-or-miss transform", *IEEE Transactions on Image Processing*, Vol. 5, Issue 6, June 1996, pp. 1060-1066

[3] L. D. Stefano, S. Mattoccia, "Fast template matching using bounded partial correlation", *Machine Vision and Applications,* Vol. 13, Issue 4, Feb. 2003, pp. 213-221

[4] Bo Tao, and B. Dickinson, "Template-based image retrieval", *IEEE International Conference on Image Processing (ICIP) 1996*, Vol. 3, Sept. 1996, pp. 871 - 874

[5] Y. Bentoutou, N. Taleb, K. Kpalma, and J. Ronsin, "An automatic image registration for applications in remote sensing", *IEEE Transactions on Geoscience and Remote Sensing,* Vol. 43, No. 9, Sept. 2005, pp. 2127-2137

[6] H. Peng, F. Long, and Z. Chi, "Document Image Recognition based on template matching of component block projections", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 25, Issue 9, Sept. 2003, pp. 1188-1192

[7] R. M. Dufour, E. L. Miller, and N. P. Galatsanos, "Template matching based object recognition with unknown geometric parameters", *IEEE Transactions on Image Processing,* Vol. 11, Issue 12, Dec. 2002, pp. 1385-2396

[8] H. D. Tagare, "Deformable 2-D template matching using orthogonal curves", *IEEE Transactions on Medical Imaging,* Vol. 16, No. 1, Feb. 1997, pp. 108 - 117

[9] R. Brunelli, T. Poggio, "Face recognition: features versus templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 15, Issue 10, Oct. 1993, pp. 1042-1052

[10] A. K. Sao, and B. Yegnanarayana, "Face verification using template matching", *IEEE Transactions on Information Forensics and Security*, Vol. 2, Issue 3, Part 2, Sep. 2007, pp. 636-641

[11] G. J. Vanderbrug, and A. Rosenfeld, "Two-stage template matching", *IEEE Transactions on Computers,* Vol. C-26, No. 4, April 1977

[12] M. Gharavi-Alkhansari, "A fast globally optimal algorithm for template matching using low-resolution pruning", *IEEE Transactions on Image Processing,* Vol. 10, Issue 4, April 2001, pp. 526-533

[13] M. Uenohara, and T. Kanade, "Use of Fourier and Karhunen-Loeve decomposition for fast pattern matching with a large set of templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 19, Issue 8, Aug. 1997, pp. 891-898

[14] H. Schweitzer, J. W. Bell, and F. Wu, "Very fast template matching", *Proceedings of the 7th European Conference on Computer Vision-Part* IV, 2002, pp. 358–372

[15] S. Omachi, and M. Omachi, "Fast template matching with polynomials", *IEEE Transactions on Image Processing,* Vol. 16, No. 8, Aug. 2007, pp. 2139-2149

[16] Z. Lin, L. S. Davis, D. Doermann, and D. DeMenthon, "Hierarchical part-template matching for human detection and segmentation", *IEEE International Conference on Computer Vision (ICCV),* Oct. 2007, pp. 1 - 8

[17] D. M. Gavrila, "Multi-feature hierarchical template matching using distance transforms", *IEEE International Conference on Pattern Recognition (ICPR),* Vol. 1, Aug. 1998, pp. 439 - 444

[18] H. van der Werff, F. van Ruitenbeek, M. van der Meijde, F. van der Meer, S. de Jong, and S. Kalubandara, "Rotation-Variant Template Matching for Supervised Hyperspectral Boundary Detection", *IEEE Geoscience and Remote Sensing Letters,* Vol. 4, No. 1, Jan. 2007, pp. 70-74

[19] D. W. Paglieroni, G. E. Ford, and E. M. Tsujimoto, “The position-orientation masking approach to parametric search for template matching”, *IEEE Transactions on Pattern Analysis and Machine Analysis,* Vol. 16**,** No. 7**,** July 1994, pp. 740-747

[20] J. P. Lewis, “Fast normalized cross-correlation”, *Vision Interface,* 1995, pp. 120-123

[21] L. D. Stefano, S. Mattoccia, and M. Mola, “An efficient algorithm for exhaustive template matching based on normalized cross-correlation”, *IEEE International Conference on Image Analysis and Processing (ICIAP),* Sept. 2003, pp. 322-327

[22] Shou-Der Wei and Shang-Hong Lai, “Fast template matching based on normalized cross correlation with adaptive multilevel winner update”, *IEEE Transactions on Image Processing*, Vol. 17, No. 11, Nov. 2008, pp. 2227 - 2235

[23] H. Foroosh, J. B. Zerubia, and M. Berthod, “Extension of phase correlation to subpixel registration”, *IEEE Transactions on Image Processing,* Vol. 11, No. 3, Mar. 2002, pp. 188–200

[24] S. Ertürk, “Digital image stabilization with sub-image phase correlation based global motion estimation”, *IEEE Transactions on Consumer Electronics,* Vol. 49, No. 4, Nov. 2003, pp. 1320-1325

[25] A. Ertürk, S. Ertürk, “Unsupervised segmentation of hyperspectral images using modified phase correlation”, *IEEE Geoscience and Remote Sensing Letters,* Vol. 3, No. 4, Oct. 2006, pp. 527-531

[26] D. Zheng, Y. Liu, and J. Zhao, “A survey of RST invariant image watermarking algorithms”, *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE),* May 2006, pp. 2086-2089

[27] C.-Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, “Rotation, scale and translation resilient watermarking for images”, *IEEE Transactions on Image Processing*, Vol. 10, No. 5, May 2001, pp. 767-782

[28] D. Casasent, and D. Psaltis, “New optical transforms for pattern recognition”, *Proceedings of the IEEE*, Vol. 65, No. 1, Jan. 1977, pp. 77-84

[29] Ming-Kuei Hu, “Visual pattern recognition by moment invariants*”, *IRE Transactions on Information Theory,* Vol. 8, No. 2, Feb. 1962, pp. 179-187

[30] A.Khotanzad, and Y. H. Hong, “Invariant image recognition by Zernike moments”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 5, May 1990, pp. 489-497

[31] J. Ricard, D. Coeurjolly, and A. Baskurt, “Generalization of angular radial transform”, *IEEE International Conference on Image Processing (ICIP)*, Vol. 4, Oct. 2004, pp. 2211-2214

[32] Sun-Kyoo Hwang, and Whoi-Yul Kim, “Fast and efficient method for computing ART”, *IEEE Transactions on Image Processing*, Vol. 15, No. 1, Jan. 2006, pp. 112-117

[33] Qin-Sheng Chen, M. Defrise, and F. Deconinck, “Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 12, Dec. 1994, pp. 1156-1168

[34] Y. S. Abu-Mostafa, and D. Psaltis, “Recognitive aspects of moment invariants”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 6, Nov. 1984, pp. 698-706

[35] J.J.K. O’Ruanaidh, and T. Pun, “Rotation, scale and translation invariant digital image watermarking”, *IEEE International Conference on Image Processing (ICIP)*, Oct. 1997, pp. 536-539

[36] D. Zheng, J. Zhao, and A. El Saddik, “RST-invariant digital image watermarking based on log-polar mapping and phase correlation”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 8, Aug. 2003, pp. 753-765

[37] F. Motallebi, and H. Aghaeinia, “RST invariant wavelet-based image watermarking using template matching techniques”, *Congress on Image and Signal Processing (CISP)*, Vol. 5, 2008, pp. 720-724

[38] S. Zokai, and G. Wolberg, “Image registration using log-polar mappings for recovery of large-scale similarity and projective transformations”, *IEEE Transactions on Image Processing*, Vol. 14, No. 10, Oct. 2005, pp. 1422-1434

[39] Chi-Man Pun, and Moon-Chuen Lee, “Log-polar wavelet energy signatures for rotation and scale invariant texture classification”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, May 2003, pp. 590-603

[40] J. Zhang, Z. Ou, and H. Wei, "Fingerprint matching using phase-only correlation and Fourier-Mellin transforms", *IEEE International Conference on Intelligent Systems Design and Applications (ISDA),* Vol. 2, Oct. 2006, pp. 379-383

[41] D. Schonberg, and D. Kirovski, "EyeCerts", *IEEE Transactions on Information Forensics and Security,* Vol.1, No. 2, June 2006, pp. 144-153

[42] K. Hotta, T. Kurita, and T. Mishima, "Scale invariant face detection method using higher-order local autocorrelation features extracted from log-polar image", *IEEE International Conference on Automatic Face and Gesture Recognition*, Apr. 1998, pp. 70-75

[43] M. Gueham, A. Bouridane, D. Crookes, and O. Nibouche, "Automatic recognition of shoeprints using Fourier-Mellin transform", *NASA/ESA Conference on Adaptive Hardware and Systems (AHS),* June 2008, pp. 487-491

[44] O. K. Al-Shaykh, and J. F. Doherty, "Invariant image analysis based on Radon transform and SVD", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* Vol. 43, No. 2, Feb. 1993, pp. 123-133

[45] J. You, W. Lu, G. Gindi, and Z. Liang, "Image matching for translation, rotation and uniform scaling by the Radon transform", *IEEE International Conference on Image Processing,* Vol. 1, Oct. 1998, pp. 847-851

[46] C. Lian, and D. Sidan, "Rotation, scale and translation invariant image watermarking using Radon transform and Fourier transform", *IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication,* Vol. 1, 2004, pp. 281-284

[47] N. D. Thuc, and B. D. Khanh, "Determining of translation and rotation of an object from its Radon transform via neural network", *IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 5, Oct. 2001, pp. 3281-3284

[48] Y. Liu, and J. Zhao, "Rotation, scale, translation invariant image watermarking based on Radon transform", *IEEE First Canadian Conference on Computer and Robot Vision,* May 2004, pp. 225-232

[49] K. Jafari-Khouzani, and H. Soltanian-Zadeh, "Radon transform orientation estimation for rotation invariant texture analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 6, June 2005, pp. 1004-1008

[50] D. Simitopoulos, D. Koutsonanos, and M. G. Strintzis, "Image watermarking resistant to geometric attacks using generalized Radon transformations", *IEEE 14th International Conference on Digital Signal Processing*, Vol. 1, 2002, pp. 85-88

[51] V. Chandran, B. Carswell, and B. Boashash, "Pattern recognition using invariants defined from higher order spectra: 2-D image inputs", *IEEE Transactions on Image Processing*, Vol. 6, No. 5, May 1997, pp. 703-712

[52] J. Heikkila, "Image scale and rotation from the phase-only bispectrum", *IEEE International Conference on Image Processing (ICIP)*, Vol. 3, Oct. 2004, pp. 1783-1786

[53] H. Kim, Y. Baek, and H. K. Lee, "Rotation, scale and translation invariant image watermark using higher order spectra", *Optica Engineering*, Vol. 42, No. 2, 2003, pp. 340-349

[54] V. Ojansivu, and J. Heikkila, "A method for blur and similarity transform invariant object recognition", *IEEE 14th International Conference on Image Analysis and Processing (ICIAP)*, Sept. 2007, pp. 583-588

[55] S. Odo, and K. Hoshino, "Real time object tracking based on retino-cortical mapping", *IEEE International Workshop on Robot and Human Interactive Communication*, Sept. 2000, pp. 47-51

[56] R. Matungka, Y. F. Zheng, and R. L. Ewing, "Object recognition using log-polar wavelet mapping", *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov. 2008, Vol. 2, pp. 559-563

[57] Fang-Fang He, Ji-Yin Sun, Wen-Pu Guo, and Li-Bo Sun, "Forward-looking scene matching with RST invariant conformation based on Fourier-Mellin transform", *IEEE International Conference on Wavelet Analysis and Pattern Recognition*, Nov. 2007, pp. 157-162

[58] B. M. Sadler, "Shift and rotation invariant object reconstruction using the bispectrum", *Workshop on Higher-Order Spectral Analysis*, June 1989, pp. 106-111

[59] A. P. Petropulu, and C. L. Nikias, "Signal reconstruction from the phase of the bispectrum", *IEEE Transactions on Signal Processing*, Vol. 40, No. 3, March 1992, pp. 601-610

[60] A. P. Petropulu, and H. Pozidis, "Phase reconstruction from bispectrum slices", *IEEE Transactions on Signal Processing*, Vol. 46, No. 2, Feb. 1998, pp. 527-530

[61] E. Trucco, and A. Verri, “Introductory techniques for 3-D computer vision”, Prentice Hall, 1998

[62] R. Matungka, Y. F. Zheng, and R. L. Ewing, “Image registration using adaptive polar transform”, *IEEE Transactions on Image Processing*, Vol. 18, No. 10, Oct. 2009, pp. 2340-2354

[63] C. H. Lampert, M. B. Blaschko, and T. Hofmann, “Efficient subwindow search: a branch and bound framework for object localization”, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 31, No. 12, Dec. 2009, pp. 2129-2142