

ROBUST AFFINE INVARIANT SHAPE DESCRIPTORS

by

Ye Mei

M.Sc.(University of Kiel, Germany) 2004

B.Sc.(Shanghai Jiaotong University, P.R.China) 1998

A dissertation

presented to Ryerson University

in partial fulfillment of the

requirement for the degree of

Doctor of Philosophy

in the Program of

Electrical and Computer Engineering.

Toronto, Ontario, Canada, 2010

© Ye Mei 2010

Author's Declaration

I hereby declare that I am the sole author of this dissertation.

I authorize Ryerson University to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

Ye Mei

I further authorize Ryerson University to reproduce this dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Ye Mei

Abstract

Robust Affine Invariant Shape Descriptors

Ye Mei

Doctor of Philosophy

Electrical and Computer Engineering

Ryerson University, Toronto, 2010

With the increasing number of available digital images, there is an urgent need of image content description to facilitate content based image retrieval (CBIR). Besides colour and texture, shape is an important low level feature in describing image content. An object can be photographed from different distances and angles. However, we often want to classify the images of the same object into one class, despite the change of perspective. So, it is desired to extract shape features that are invariant to the change of perspective. The shape of an object from one viewpoint to another can be linked through an affine transformation, if it is viewed from a much larger distance than its size along the line of sight. Those invariant shape features are known as affine invariant shape representations. Because of the change of perspective, it is more difficult to develop affine invariant shape representations than normal ones. The goal of this work is to develop robust affine invariant shape descriptors. Through shape retrieval experiments, we find that the performance of the existing affine invariant shape representations are not satisfactory. Especially, when the shape boundary is corrupted by noise, their performance degrades quickly. In this work, two new affine invariant contour-based shape descriptors, the ICA Fourier shape descriptor (ICAFSD) and

the whitening Fourier shape descriptor (WFSD) have been developed. They perform better than most of the existing affine invariant shape representations, while having compact feature size and low computational time requirement. Four region-based affine-invariant shape descriptors, the ICA Zernike moment shape descriptor (ICAZMSD), the whitening Zernike moment shape descriptor (WZMSD), the ICA orthogonal Fourier Mellin moment shape descriptor (ICAOFMMSD), and the whitening orthogonal Fourier Mellin moment shape descriptor (WOFMMSD), are also proposed, in this work. They can be applied to both simple and complex shapes, and have close to perfect performance in retrieval experiments. The advantage of those newly proposed shape descriptors is even more apparent in experiments on shapes with added boundary noise: Their performance does not deteriorate as much as the existing ones.

Acknowledgement

I would like to thank my supervisor, Prof. Dimitrios Androutsos, for his guidance and supervision through my Ph.D. studies. It is my great pleasure to have worked with such a diligent, dedicated, humble and knowledgeable professor. I also want to thank him for teaching me how to write clearly and guiding me through the daunting task of writing a Ph.D. thesis.

I would like to thank my colleagues, Richard Rzeszutek and Raymond Phan, and many other research students who have shared our lab, for the friendly environment they provided and the help they gave to me. Special thanks to Richard Rzeszutek, for kindly reading over the thesis and providing many important comments.

Many thanks to our system support staff, Bruce Derwin, who always solves computer and network problems quickly, so that those important tools needed in my research always functioned well.

Great thanks to my parents for their immeasurable support and understanding. Ph.D. studies took a long time. I thank them for their patience.

I would like to thank Prof. Alagan Anpalagan, Prof. Sri Krishnan, Prof. Soosan Beheshti, Prof. Alireza Sadeghian, and Prof. Kostas Plataniotis for taking the time out of their very busy schedules to be on my exam committee.

To Changlin and Yiwen

Contents

Abstract	iii
Acknowledgement	v
1 Introduction	1
1.1 Motivation and background	1
1.2 Objective	3
1.3 Contributions	3
1.3.1 Proposal of new contour-based affine-invariant shape descriptors . .	4
1.3.2 Proposal of new region-based affine-invariant shape descriptors . .	4
1.4 Outline	6
2 Review of Shape Representation and Their Extractions	8
2.1 Introduction	8
2.2 Contour-based shape representations and their extractions	9
2.2.1 1D function shape signatures	10
2.2.1.1 Centroid distance signature	10
2.2.1.2 Tangent angle signature	11
2.2.1.3 Complex coordinates signature	11
2.2.2 Chain code and chain code histogram	11
2.2.3 Boundary moments	12
2.2.4 Simple shape descriptors	13
2.2.4.1 Convexity	13
2.2.4.2 Ratio of principle axes (Eccentricity)	13

2.2.4.3	Compactness	14
2.2.4.4	Circular variance	15
2.2.4.5	Elliptic variance	15
2.2.5	Curvature scale space (CSS) representation	17
2.2.5.1	CSS representation	17
2.2.5.2	CSS distance measure	22
2.3	Region-based shape representation and their extractions	22
2.3.1	Fourier-wavelet descriptor	23
2.3.2	Generic Fourier descriptor	24
2.3.3	R-transform representation	25
2.3.4	Geometric moment descriptor	26
2.3.5	Complex moment descriptor	32
2.4	Summary	32

3 Review and Performance Study of Previous Affine-Invariant Shape Representations 33

3.1	Introduction	33
3.2	Affine transformation	34
3.3	Affine-invariant parameters	35
3.3.1	Affine arc length	35
3.3.2	Enclosed area parameter	35
3.4	Affine-invariant Fourier shape descriptor	36
3.5	Affine-invariant wavelet-based shape representation	40
3.6	Affine-invariant curvature scale space shape descriptor	43
3.7	Affine moment invariants by Taubin and Cooper	46
3.8	Affine moment invariants by Flusser and Suk	48
3.9	Experimental results	49
3.9.1	Test database	49
3.9.2	Distance measure	51
3.9.3	Retrieval accuracy	55
3.9.3.1	Comparison of the AMI-TC and the AMI-FS	56

3.9.3.2	Comparison of the AIFSD, the AICSSSD, and the AIWSR	61
3.9.3.3	Comparison of the AIFSD, the AICSSSD, and the AMI-FS	65
3.9.4	Comparison of extraction time, distance calculation time, and compactness	69
3.10	Summary	72
4	Independent Component Analysis	73
4.1	Introduction	73
4.2	ICA, its ambiguities, its relationship with whitening, and its estimation criteria	74
4.2.1	The ICA model	74
4.2.2	Ambiguities of ICA	75
4.2.3	Whitening as a preprocessing step in ICA	75
4.2.3.1	Independence	76
4.2.3.2	Uncorrelatedness	76
4.2.3.3	Whiteness	76
4.2.3.4	Whitening simplifies the ICA estimation	77
4.2.4	ICA by maximization of non-gaussianity	78
4.2.4.1	Measuring non-gaussianity by kurtosis	78
4.2.4.2	Measuring non-gaussianity by negentropy	79
4.2.5	ICA by minimization of mutual information	80
4.3	Illustration of ICA	81
4.3.1	With subgaussian data	81
4.3.2	With supergaussian data	84
4.4	Summary	87
5	Novel Contour-Based Affine Invariant Shape Descriptors	89
5.1	Introduction	89
5.2	Canonicalization of shape contour by ICA	89
5.3	ICA Fourier shape descriptor	91
5.3.1	FT and DFT	94
5.3.2	DFT on ICA-canonicalized shape contour	95

5.4	Whitening Fourier shape descriptor	99
5.5	Experimental results	101
5.5.1	Experimental data	102
5.5.2	Retrieval accuracy	102
5.5.3	Comparison of extraction time, distance calculation time, and compactness	106
5.6	Summary	109
6	Novel Region-Based Affine Invariant Shape Descriptors	110
6.1	Introduction	110
6.2	Canonicalization of shape by ICA	111
6.3	ICA Zernike moment shape descriptor	116
6.3.1	Zernike moments	117
6.3.2	The invariant properties of the Zernike moments	122
6.3.2.1	Rotation invariant	122
6.3.2.2	Reflection invariant	123
6.3.3	Zernike moments extraction from the ICA-canonicalized shape . . .	124
6.4	ICA orthogonal Fourier-Mellin moment shape descriptor	126
6.4.1	Orthogonal Fourier-Mellin moments	126
6.4.2	The invariant properties of the orthogonal Fourier-Mellin moments	129
6.4.2.1	Rotation invariant	129
6.4.2.2	Reflection invariant	131
6.4.3	Orthogonal Fourier-Mellin moments extraction from the ICA-canonicalized shape	132
6.5	Whitening Zernike moment shape descriptor and whitening orthogonal Fourier-Mellin moment shape descriptor	135
6.6	Experimental results	136
6.6.1	Experimental data	136
6.6.2	Retrieval accuracy	136
6.6.3	Comparison of extraction time, distance calculation time, and compactness	144

6.7	Application in traffic sign retrieval	146
6.8	Summary	147
7	Conclusions and Future Work	150
7.1	Conclusions	150
7.2	Future work	152
A	Plots of Zernike Radial Polynomials	156
B	Lists of Zernike Moment Bases	160
C	Lists of Orthogonal Fourier-Mellin Moment Bases	165
	Bibliography	170

List of Figures

1.1	Images of a flag, taken from two different perspectives	2
1.2	Images of a stop sign, taken from two different perspectives	3
2.1	Simple shapes	9
2.2	Complex shapes	9
2.3	Chain code direction (a) Chain code in eight directions ; (b) chain code in four directions	12
2.4	Shape contours and convex hulls	13
2.5	Principle and secondary axes of the shape contours	14
2.6	Shape contour and the circle with equal amount of area enclosed	15
2.7	Shrinkage and smoothing of the contour curve, as σ increases	19
2.8	The CSS images and the maxima (the maxima are indicated using red squares)	21
2.9	Complex shape image and its corresponding polar image in Cartesian co- ordinates	24
2.10	Simple shape image and its corresponding polar image in Cartesian coor- dinates	24
2.11	Radon transform, projection along line l	25
2.12	Shape image, its corresponding Radon transform and R-transform	27
2.13	Simple shape image, its corresponding Radon transform and R-transform	28
2.14	Simple shape image, its corresponding Radon transform and R-transform	29
2.15	Complex shape image, its corresponding Radon transform and R-transform	30

3.1	All areas are changed in the same ratio under an affine mapping (Figure from [1])	36
3.2	Shape image and its corresponding AIFSD	38
3.3	Shape image and its corresponding AIFSD	39
3.4	Shape image and its corresponding AIFSD	39
3.5	Shape image and its corresponding AIFSD	40
3.6	Shape image and its corresponding AIWSR	41
3.7	Shape image and its corresponding AIWSR	42
3.8	Shape image and its corresponding AIWSR	42
3.9	Shape image and its corresponding AIWSR	43
3.10	Shape image and its corresponding AICSSSD	45
3.11	Shape image and its corresponding AICSSSD	45
3.12	Shape image and its corresponding AICSSSD	46
3.13	Shape image and its corresponding AICSSSD	46
3.14	70 Benchmark shape images from MPEG-7 Shape B	52
3.15	50 Benchmark shape images	53
3.16	Sample shape images with zero-mean Gaussian noise added to the shape boundaries	54
3.17	Calculation of precision rate and recall rate.	56
3.18	Average precision-recall graphs for 4000 objects using different affine-invariant region-based shape descriptors on the 4000 shape complex shape database.	57
3.19	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database without noise.	57
3.20	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=30\text{dB}$	58
3.21	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$	58

3.22	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.0206\text{dB}$	59
3.23	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=24.7712\text{dB}$	59
3.24	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=23.0103\text{dB}$	60
3.25	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$	60
3.26	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database without noise.	61
3.27	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=30\text{dB}$	62
3.28	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$	62
3.29	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.0206\text{dB}$	63
3.30	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=24.7712\text{dB}$	63
3.31	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=23.0103\text{dB}$	64

3.32	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$	64
3.33	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database without noise.	65
3.34	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=30\text{dB}$	66
3.35	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$	66
3.36	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.0206\text{dB}$	67
3.37	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=24.7712\text{dB}$	67
3.38	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=23.0103\text{dB}$	68
3.39	Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$	68
3.40	Comparison of feature extraction time (seconds)	70
3.41	Comparison of feature distance calculation time (milliseconds)	71
3.42	Comparison of the size of the shape representations	71
4.1	(a)The joint distribution of the independent components s_1 and s_2 with uniform distributions; (b)Histogram of s_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c)Histogram of s_2 , in comparison with gaussian distribution with unit variance (shown in red colour).	82

4.2	(a)The joint distribution of the mixtures x_1 and x_2 ; (b)Histogram of x_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c)Histogram of x_2 , in comparison with gaussian distribution with unit variance (shown in red colour).	83
4.3	(a)The joint distribution of the mixtures z_1 and z_2 ; (b)Histogram of z_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c)Histogram of z_2 , in comparison with gaussian distribution with unit variance (shown in red colour).	83
4.4	(a)The joint distribution of the estimated independent components y_1 and y_2 with uniform distributions; (b)Histogram of y_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c)Histogram of y_2 , in comparison with gaussian distribution with unit variance (shown in red colour).	84
4.5	(a)The joint distribution of the independent components s_1 and s_2 with Laplacian distributions; (b)Histogram of s_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c)Histogram of s_2 , in comparison with gaussian distribution with unit variance (shown in red colour).	85
4.6	(a)The joint distribution of the mixtures x_1 and x_2 ; (b)Histogram of x_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c)Histogram of x_2 , in comparison with gaussian distribution with unit variance (shown in red colour).	86
4.7	(a)The joint distribution of the mixtures z_1 and z_2 ; (b)Histogram of z_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c)Histogram of z_2 , in comparison with gaussian distribution with unit variance (shown in red colour).	86
4.8	(a)The joint distribution of the estimated independent components y_1 and y_2 with Laplacian distributions; (b)Histogram of y_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c)Histogram of y_2 , in comparison with gaussian distribution with unit variance (shown in red colour).	87

5.1	Contours of two affine-related shapes (helicopter)	91
5.2	ICA canonicalized shapes corresponding to the shape in Figure 5.1 (a) . . .	92
5.3	ICA canonicalized shapes corresponding to the shape in Figure 5.1 (b) . . .	93
5.4	(a) Shape contour of a horse shoe, and (b) its ICA-canonicalized shape contour	94
5.5	Centroid distances, corresponding to the ICA canonicalized shapes in Fig- ure 5.2	96
5.6	Centroid distances, corresponding to the ICA canonicalized shapes in Fig- ure 5.3	97
5.7	ICAFSDs of the two affine-related shapes (helicopter) shown in Figure 5.1	98
5.8	ICAFSD of the horseshoe shape shown in Figure 5.4 (a)	99
5.9	Comparison of the ICAFSDs	99
5.10	Diagram of ICA-Fourier shape descriptor extraction	100
5.11	Diagram of whitening-Fourier shape descriptor extraction	100
5.12	(a) shape contour, (b) whitened shape contour, (c) centroid distance and (d) WFSD	101
5.13	Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database without noise. .	102
5.14	Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=30\text{dB}$	103
5.15	Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$	103
5.16	Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.0206\text{dB}$	104
5.17	Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=24.7712\text{dB}$	104

5.18	Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=23.0103\text{dB}$	105
5.19	Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$	105
5.20	Overlapping of the average precision-recall curve of the ICAFSD and that of the WFSD (on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$).	106
5.21	Comparison of feature extraction time (seconds)	107
5.22	Comparison of feature distance calculation time (milliseconds)	108
5.23	Comparison of the size of the shape representations	109
6.1	Two affine-related shapes (eagle)	113
6.2	ICA canonicalized shapes corresponding to the shape in Figure 6.1 (a) . . .	114
6.3	ICA canonicalized shapes corresponding to the shape in Figure 6.1 (b) . . .	115
6.4	(a) A pepper shape, and (b) its ICA-canonicalized shape	116
6.5	Zernike radial polynomials $R_{nm}(r)$, ($n = 0, 1, 2, 3, 4$)	120
6.6	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 0, m = 0$)	120
6.7	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 1, m = 1$)	120
6.8	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 2, m = 0$)	121
6.9	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 2, m = 2$)	121
6.10	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 3, m = 1$)	121
6.11	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 3, m = 3$)	122
6.12	Diagram of ICAZMSD extraction	124
6.13	ICAZMSDs of the two affine related eagle shapes shown in Figures 6.1 . . .	125
6.14	ICAZMSD of the pepper shape in Figure 6.4	125
6.15	Comparison of the ICAZMSDs	126
6.16	Orthogonal Mellin radial functions, $n=1-6$	127
6.17	Real and imaginary parts of OFMM base, ($n=1, m=1$)	129
6.18	Real and imaginary parts of OFMM base, ($n=2, m=1$)	129

6.19	Real and imaginary parts of OFMM base, (n=2,m=2)	129
6.20	Real and imaginary parts of OFMM base, (n=3,m=1)	130
6.21	Real and imaginary parts of OFMM base, (n=3,m=2)	130
6.22	Real and imaginary parts of OFMM base, (n=3,m=3)	130
6.23	Diagram of ICAOFMMSD extraction	133
6.24	ICAOFMMSDs of the two affine related eagle shapes shown in Figures 6.1	133
6.25	ICAOFMMSD of the pepper shape in Figure 6.4	134
6.26	Comparison of the ICAOFMMSDs	134
6.27	Diagram of WZMSD extraction	135
6.28	Diagram of WOFMMSD extraction	136
6.29	(a)(b):WZMSD and WOFMMSD of the eagle shape in Figure 6.1 (a); (c)(d):WZMSD and WOFMMSD of the eagle shape in Figure 6.1 (b); (c)(d):WZMSD and WOFMMSD of the pepper shape in Figure 6.4.	137
6.30	Average precision-recall graphs of 4000 retrievals using different affine in-variant shape descriptors on complex shape database.	138
6.31	Average precision-recall graphs of 5600 retrievals using different affine in-variant shape descriptors on simple shape database with no noise.	139
6.32	Average precision-recall graphs of 5600 retrievals using different affine in-variant shape descriptors on simple shape database with Gaussian noise at $SNR_{dB}=30dB$.	139
6.33	Average precision-recall graphs of 5600 retrievals using different affine in-variant shape descriptors on simple shape database with Gaussian noise at $SNR_{dB}=26.9897dB$.	140
6.34	Average precision-recall graphs of 5600 retrievals using different affine in-variant shape descriptors on simple shape database with Gaussian noise at $SNR_{dB}=26.0206dB$.	140
6.35	Average precision-recall graphs of 5600 retrievals using different affine in-variant shape descriptors on simple shape database with Gaussian noise at $SNR_{dB}=24.7712dB$.	141

6.36	Average precision-recall graphs of 5600 retrievals using different affine invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=23.0103\text{dB}$	141
6.37	Average precision-recall graphs of 5600 retrievals using different affine invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$	142
6.38	Overlapping of the average precision-recall curve of the ICAZMSD and that of the WZMSD (on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$).	143
6.39	Overlapping of the average precision-recall curve of the ICAOFMMSD and that of the WOFMMSD (on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$).	143
6.40	Comparison of feature extraction time (seconds)	144
6.41	Comparison of feature distance calculation time (milliseconds)	145
6.42	Comparison of the size of the shape representations	146
6.43	(a)Traffic signs, (b)their shape images, and (c)their ICAZMSDs	148
6.44	Retrieved traffic sign pictures, from the most related to the least related (left to right, top to bottom). The five pictures with 'Stop' signs are all in the top five matches.	149
7.1	Images of a colour texture under different illumination condition (Image source: Outex colour texture database [2])	153
7.2	The magnitude, phase angle, real, and imaginary parts of one of the spherical harmonics ($L = 5, M = 3$), where L and M are the indexes.	154
7.3	The magnitude, phase angle, real, and imaginary parts of one of the OSH-MMs ($L = 5, M = 3$) at radius, $r=0.3$ and $r=0.9$	155
A.1	Zernike Radial Polynomials $R_{nm}(r), (n = 0, 1, \dots, 4)$	156
A.2	Zernike Radial Polynomials $R_{nm}(r), (n = 5)$	157
A.3	Zernike Radial Polynomials $R_{nm}(r), (n = 6)$	157
A.4	Zernike Radial Polynomials $R_{nm}(r), (n = 7)$	158
A.5	Zernike Radial Polynomials $R_{nm}(r), (n = 8)$	158

A.6	Zernike Radial Polynomials $R_{nm}(r)$, ($n = 9$)	159
A.7	Zernike Radial Polynomials $R_{nm}(r)$, ($n = 10$)	159
B.1	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 0, m = 0$)	160
B.2	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 1, m = 1$)	160
B.3	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 2, m = 0$)	161
B.4	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 2, m = 2$)	161
B.5	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 3, m = 1$)	161
B.6	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 3, m = 3$)	162
B.7	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 4, m = 0$)	162
B.8	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 4, m = 2$)	162
B.9	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 4, m = 4$)	163
B.10	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 5, m = 1$)	163
B.11	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 5, m = 3$)	163
B.12	Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 5, m = 5$)	164
C.1	Real and imaginary parts of OFMM base, ($n=1, m=1$)	165
C.2	Real and imaginary parts of OFMM base, ($n=2, m=1$)	165
C.3	Real and imaginary parts of OFMM base, ($n=2, m=2$)	166
C.4	Real and imaginary parts of OFMM base, ($n=3, m=1$)	166
C.5	Real and imaginary parts of OFMM base, ($n=3, m=2$)	166
C.6	Real and imaginary parts of OFMM base, ($n=3, m=3$)	166
C.7	Real and imaginary parts of OFMM base, ($n=4, m=1$)	167
C.8	Real and imaginary parts of OFMM base, ($n=4, m=2$)	167
C.9	Real and imaginary parts of OFMM base, ($n=4, m=3$)	167
C.10	Real and imaginary parts of OFMM base, ($n=4, m=4$)	167
C.11	Real and imaginary parts of OFMM base, ($n=5, m=1$)	168
C.12	Real and imaginary parts of OFMM base, ($n=5, m=2$)	168
C.13	Real and imaginary parts of OFMM base, ($n=5, m=3$)	168
C.14	Real and imaginary parts of OFMM base, ($n=5, m=4$)	168
C.15	Real and imaginary parts of OFMM base, ($n=5, m=5$)	169

List of Tables

2.1	Shapes and their simple shape descriptor values	16
3.1	Feature extraction time (seconds)	69
3.2	Feature distance calculation time (milliseconds)	70
3.3	Size of the shape representations	71
5.1	Feature extraction time(seconds)	107
5.2	Feature distance calculation time(milliseconds)	108
5.3	Size of the shape representations	108
6.1	Zernike radial polynomials $R_{nm}(r)$	119
6.2	The polynomial set of $Q_n(r)$	128
6.3	Feature extraction time(seconds)	144
6.4	Feature distance calculation time(milliseconds)	145
6.5	Size of the shape representations	146

Nomenclature

(c_x, c_y) Centroid

(x, y) Cartesian coordinates

$\ddot{x}(u)$ The 2nd derivatives of $x(u)$

$\ddot{y}(u)$ The 2nd derivatives of $y(u)$

$\dot{x}(u)$ The 1st derivatives of $x(u)$

$\dot{y}(u)$ The 1st derivatives of $y(u)$

η Enclosed area parameter

Γ Curve of the shape contour

κ Curvature of a plane curve

κ_a Curvature of the contour of an affine transformed shape

λ Eigenvalues

A Affine transform matrix

C Covariance matrix

D Diagonal matrix

M Mixing matrix

\mathbf{M}_a	Mixing matrix after affine transform
\mathbf{m}_i	Column vector of the Mixing matrix \mathbf{M}
\mathbf{P}	Permutation matrix
\mathbf{s}	Independent component sources
\mathbf{T}	Translation vector
\mathbf{V}	Whitening matrix
\mathbf{W}	Demixing matrix
\mathbf{x}	Pixel coordinate of the shape (or shape contour)
\mathbf{x}_a	Pixel coordinate of the shape (or shape contour) after affine transform
Φ_{nm}	Orthogonal Fourier-Mellin moments
σ	Standard deviation
τ	Affine arc length
θ	Tangent angle signature
A_{cd}	Root mean square amplitude of the centroid distance function
$cvar$	Circular variance
$evar$	Elliptic variance
g	The gaussian function
g_{uu}	The 2nd derivatives of the gaussian function
g_u	The 1st derivatives of the gaussian function
H	Differential entropy
I	Mutual information

J	Negentropy
$kurt$	Kurtosis
m_k	The k th moment
P	Perimeter
p	Probability density function
P_{cd}	Average power of the centroid distance function
P_{nm}	Orthogonal Fourier-Mellin polynomials
P_{noise}	Average power of the noise
Q_n	Orthogonal Mellin radial functions
$R_{ab}(l)$	Normalized cross-correlation coefficient
R_{nm}	Zernike radial functions
R_{prax}	Ratio of principle axes
V_{nm}	Zernike polynomials
Z_{nm}	Zernike moments
Γ_σ	Curve of the smoothed shape contour
μ_k	The k th central moment

List of Acronyms

AICSSD Affine-Invariant Curvature Scale Space Shape Descriptor

AIFSD Affine-Invariant Fourier Shape Descriptor

AIWSR Affine-Invariant Wavelet-based Shape Representation

CBIR Content Based Image Retrieval

CCH Chain Code Histogram

CMD Complex Moment Descriptor

CSS Curvature Scale-Space

CSSSD Curvature Scale Space Shape Descriptor

DFT Discrete Fourier Transform

ECG Electrocardiogram

EEG Electroencephalography

EGG Electrogastrogram

EUC Euclidean Distance

FWD Fourier-Wavelet Descriptor

GFD Generic Fourier Descriptor

GMD Geometric Moment Descriptor

ICA Independent Component Analysis

ICAFSD ICA Fourier Shape Descriptor

ICAOFMMSD ICA Orthogonal Fourier Mellin Moment Shape Descriptor

ICASS ICA Shape Signature

ICAZMSD ICA Zernike Moment Shape Descriptor

MEG Magnetoencephalography

OFMM Orthogonal Fourier Mellin Moment

OSHMM Orthogonal Spherical Harmonics Mellin Moment

PCA Principal Component Analysis

PDF Probability Density Function

SAD Sum of Absolute Differences

SNR Signal-to-Noise Ratio

WFSD Whitening Fourier shape descriptor

WOFMMSD Whitening Orthogonal Fourier Mellin Moment Shape Descriptor

WZMSD Whitening Zernike Moment Shape Descriptor

ZM Zernike Moment

Chapter 1

Introduction

1.1 Motivation and background

A picture is worth a thousand words. Images often contain quite a lot of information. Nowadays, with the advent and development of digital photography, capturing images is easier than ever before. They are widely used in different areas, such as retail catalogs, news media, medical diagnosis, remote sensing, and etc.. With the decreasing cost of digital storage media, the number of digital images stored around the world is also increasing rapidly. However, the huge amount of images, makes the efficient search of those images a challenging work.

One common way of searching images is by textual indexing. For example, Google image search [3] is done through textual indexing. When the word 'flower' is used as the searching keyword, Google images will show the images that are linked with the word 'flower'. For example, most of the search results will return image files that have the word 'flower' in their filenames or in the surrounding text. That works fine, sometimes. The problem is that images with a flower in them may not be retrieved, if the word 'flower' is neither in the filename nor in the surrounding text. Also, the amount of work required to annotate all the images correctly is daunting. In fact, most of the images around the world are not annotated.

Thus, it is desirable to have a technology that can retrieve images based on their visual content. Such technology is commonly referred to as content-based image retrieval

(CBIR). The term CBIR was most likely coined by T.Kato in 1992 [4], when describing automatic retrieval of images from a database based on colour and shape. Since then, CBIR has attracted the attention of many researchers in the fields of computer vision, image processing and pattern recognition and thus many related research works have been published [5, 6, 7, 8, 9, 10, 11, 12, 13].

Recently, Google is trying to extend from textual-based search to visual content-based search, such as the 'Google Goggles', which allows one to use pictures to search the web [14]. When the user captures an image, Google extracts features from it, compares those features against every item in its image database and returns the results to the user, ordered by rank [15].

In CBIR, essential features are extracted from images to represent image content. Low level features such as colour [16, 17, 18, 19, 20, 21], texture [22, 23, 24, 25], shape [26, 27, 28] or a combination of them [29, 30, 31, 32, 33], can be extracted directly from images and have been found effective in representing image content. Shape, in particular, is an important and effective low level feature. For example, the shape of a horse and the shape of a flower is easily distinguishable. In real life, objects in images are not always photographed from the same position, distance or angle. Therefore, the shapes of an object in two photos taken from different perspectives are related, but not exactly the same (Figure 1.1 and Figure 1.2). However, it is desirable to classify those related images of the same object into one class, even though the images were taken from different perspectives.



(a)



(b)

Figure 1.1: Images of a flag, taken from two different perspectives



(a)



(b)

Figure 1.2: Images of a stop sign, taken from two different perspectives

1.2 Objective

As we can see from the background review, shape is an important and effective feature in representing image content in CBIR. Objects are often photographed from different perspectives, yet we still want to classify the same objects into one class, despite the change in perspective. Therefore, it is desirable to extract shape features that are invariant to the change of perspective. The shape of an object from one viewpoint to another can be linked through an affine transformation, if it is viewed from a much larger distance than its size along the line of sight [34]. Those invariant shape features are known as affine-invariant shape representations. The objective of this research is to develop effective affine-invariant shape representations that have strong discrimination power, compact size, and low computational requirement. Since current image segmentation techniques are not perfect, there is often noise on shape boundaries. Therefore, the newly developed affine-invariant shape representations should also be robust against shape boundary noise.

1.3 Contributions

The contributions of this dissertation are listed below:

1.3.1 Proposal of new contour-based affine-invariant shape descriptors

Two contour-based affine-invariant shape descriptors are presented:

1. *ICA Fourier shape descriptor (ICAFSD)* [35]: The independent component analysis (ICA) [36] and the discrete Fourier transform (DFT) [37] are both utilized in developing the proposed ICAFSD. ICA is used to transform a shape contour into one of eight possible canonical shape contours. Those eight possible canonical shape contours are different only by a rotation of 90, 180, or 270 degrees, and/or a reflection. The DFT is then applied on the centroid distance of the canonical shape contour. Since a rotation of the canonical shape contour will only cause phase changes in the frequency domain, but no magnitude changes, the magnitudes of the DFT coefficients are used as the newly proposed ICAFSD. The ICAFSD has a compact size, and low computational time requirement, and out performs most of the existing affine-invariant shape descriptors in retrieval experiments.
2. *Whitening Fourier shape descriptor (WFSD)* [35]: The studies on ICA show that whitened data and the ICA-ed data are different only by a rotation and/or a reflection, which the ICA algorithm needs to estimate. The newly proposed WFSD also first transforms a shape contour into its canonical one, but doesn't further estimate that rotation and/or the reflection of the shape contour, since the DFT will anyway be used later to extract rotation and reflection invariant features. Because of that, the WFSD has even lower computational time requirement than the ICAFSD, but maintains the same compactness and retrieval accuracy as the ICAFSD.

1.3.2 Proposal of new region-based affine-invariant shape descriptors

Four region-based shape descriptors are presented:

1. *ICA Zernike moment shape descriptor (ICAZMSD)* [38, 39]: Comparing with affine-invariant contour-based shape descriptors, which can only be applied to simple shapes, the region-based ICAZMSD can be applied to both simple and complex

shapes. ICA is applied to the coordinates of the shape pixels, instead of the coordinates of the shape contour, to transform the shape into a canonical shape. There are eight possible positions of the canonical shape, which are different by a 90, 180, or 270 degree rotation, and/or a reflection. Zernike moments (ZMs) are then extracted from the canonical shape, and the magnitudes of the ZMs are the newly proposed ICAZMSD. Because the magnitudes of the ZMs are rotation and reflection invariant, the value of the ICAZMSD will be the same, regardless which of the eight possible positions the canonical shape takes. The proposed ICAZMSD has a compact size, and acceptable computational time requirement. It performs far better than existing affine-invariant shape descriptors in retrieval experiments. And it is very robust under noisy condition.

2. *Whitening Zernike moment shape descriptor (WZMSD)* [39]: The WZMSD uses whitening to transform a shape into its canonical shape and extracts the magnitudes of the ZMs from the canonical shape, as the affine-invariant shape descriptor. As whitening is used, instead of ICA, the WZMSD has lower computational time requirement than the ICAZMSD, but maintains the same compactness and high retrieval accuracy as the ICAZMSD.
3. *ICA orthogonal Fourier Mellin moment shape descriptor (ICAOFMMSD)*: The orthogonal Fourier Mellin moments are relatively new type of moments. It has been found that the OFMMs have better performance than the ZMs in image reconstruction. In this work, the OFMMs are used together with ICA to develop the ICAOFMMSD. ICA is also used to transform shapes into their canonical form. The magnitudes of the OFMMs are then extracted from the canonical shape as the proposed ICAOFMMSD. The proposed ICAOFMMSD, can also be applied to both simple shape and complex shapes. It has a compact size and acceptable computational time requirement. Its retrieval perform is far better than those of the existing affine-invariant shape descriptors.
4. *Whitening orthogonal Fourier Mellin moment shape descriptor (WOFMMSD)*: It has similar feature extraction steps as the ICAOFMMSD. But, whitening, instead of ICA, is used to transform shapes into their canonical shapes. The WOFMMSD requires

less computational time than the ICAOFMMSD, while still having the same compactness and high retrieval accuracy as the ICAOFMMSD.

1.4 Outline

In *Chapter 1*, the background, objective and contributions of this research were presented.

Chapter 2, provides a general picture of what shape representations are and how they are extracted. It reviews many of the existing *regular* shape representations. The reviewed shape representations include: the 1D function shape signatures, the chain codes, the chain code histogram, the boundary moments, the simple shape descriptors, the curvature scale-space representation, the Fourier-wavelet descriptor, the generic Fourier descriptor, the R-transform, the geometric moment descriptor and the complex moment descriptor. The chapter also explains the difference between simple shape and complex shape, and the difference between contour-based shape representations and region-based shape representations.

Chapter 3, gives the definition of the *affine transformation*. Several of the existing *affine-invariant* shape representations are then reviewed and studied in detail in the chapter. Those include: the affine-invariant Fourier shape descriptor, the affine-invariant wavelet-based shape representation, the affine-invariant curvature scale space shape descriptor, the affine moment invariants by Taubin and Cooper and the affine moment invariants by Flusser and Suk. These affine-invariant shape representations are tested and compared in shape image retrieval experiments.

Chapter 4, studies an important statistical tool, ICA, which is used to transform shape contours or shapes into their canonical forms. The estimation principles of ICA, and the relationship between ICA and whitening, are discussed. Two examples of ICA on random data are used to illustrate the principles and the relationship.

Chapter 5, introduces the two proposed contour-based affine-invariant shape descriptors, the ICAFSD and the WFSD. How ICA is able to transform shape contours into their canonical forms, and how the DFT is able to extract invariant shape descriptors from the centroid distances of the canonical forms, are explained. Given the relationship between

whitening and ICA, the WFSD is further proposed to reduce the computational time requirement of the ICAFSD. The computational time, compactness, and retrieval accuracy of the two proposed affine-invariant shape descriptors are compared with those of the existing ones. Shape retrieval experiments show that the proposed affine-invariant shape descriptors perform better than most of the existing ones, while having low computational requirements and compact sizes.

Chapter 6, introduces the four proposed region-based affine-invariant shape descriptors the ICAZMSD, the WZMSD, the ICAOFMMSD and the WOFMMSD. How ICA is able to transform shapes into their canonical forms, and how ZMs and OFMMs can be used to extract invariant region-based shape descriptors, are explained. The WZMSD and the WOFMMSD are further proposed to reduce the computational time requirements of the ICAZMSD and the ICAOFMMSD, respectively. The computational time, compactness, and retrieval accuracy of the four proposed affine-invariant shape descriptors are compared with those of the existing ones. Shape retrieval experiments show that the proposed affine-invariant shape descriptors perform far better than the existing ones, while having acceptable computational time requirements and compact descriptor sizes.

Chapter 7, concludes the thesis and recommends future research directions.

Chapter 2

Review of Shape Representation and Their Extractions

2.1 Introduction

Shape representations can be generally categorized into two classes, i.e., contour-based and region-based. Contour-based shape representations, which utilize only the contour information of the shape, have their limitations, i.e., they can describe only simple shapes with a single connected region as those shown in Figure 2.1, but not complex shapes with holes in the shape or consisting of several disjoint regions, as those in Figure 2.2. By contrast, region-based shape representations, which utilize all the pixels of a shape, have no such limitations. They can describe both simple and complex shapes. Although, the goal of this work is to develop robust affine invariant shape representations and retrieval methods, we need to first review those regular shape representations that are not affine invariant, since some of the affine invariant ones are inspired by, or are related to them. Such a review will also give a general picture of shape representations.

A shape representation can usually be a vector or a 1D function. Those vector representations are usually called “shape descriptors”, while those function ones are usually called “shape signatures”. One kind of shape representation can use different similarity measures. For example, shape descriptors can be compared using one of many commonly used distance measures, such as the sum of absolute difference and the sum of Euclidean

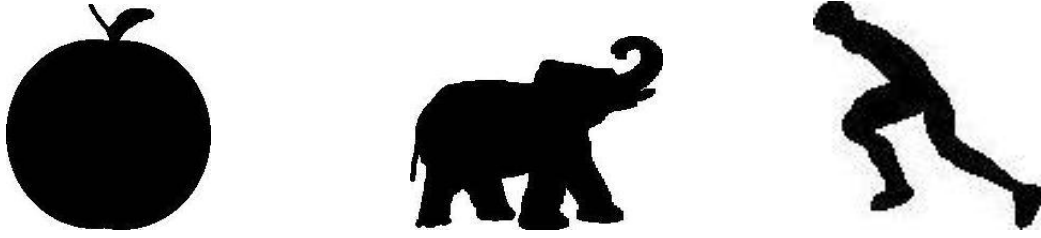


Figure 2.1: Simple shapes



Figure 2.2: Complex shapes

distance. The review in this chapter will focus on shape feature extraction. In Section 2.2, the contour-based shape representation will be reviewed; In Section 2.3, the region-based shape representations will be reviewed; Section 2.4 summarizes this chapter.

2.2 Contour-based shape representations and their extractions

Contour-based shape representations only require shape boundary information in feature extraction. In this section, the following commonly used contour-based shape representations will be reviewed:

1. *1D function shape signatures* [40]: These capture the perceptual feature of the shape and turn 2D shape contours into 1D functions. Centroid distance signature, tangent angle signature and complex coordinates signature are commonly used 1D function shape signatures.
2. *Chain codes* [41, 42]: These are a sequence of numbers that represent the directions of each steps of movements along the shape contour.

3. *Chain code histogram* [43]: It is a histogram calculated from the chain code and reflects the probabilities of different directions present in a shape contour.
4. *Boundary moments* [44]: These are 1D moments extracted from complex coordinates signature to give a more concise representation.
5. *Simple shape descriptors* [45]: There are the simple geometric features that can be used to discriminate shapes with large differences.
6. *Curvature scale-space representation* [46]: This is a shape descriptor calculated from the curvature of an evolving shape, using scale space theory.

Besides these aforementioned contour-based shape representations, the Fourier shape descriptor [47] is also a contour-based shape representation, which is also used in developing the affine invariant ICA Fourier shape descriptor and the affine invariant whitening Fourier shape descriptor [35]. It will be discussed in Chapter 5.

2.2.1 1D function shape signatures

A one-dimensional function that is extracted from a shape contour to represent a shape, is usually called a shape signature. Commonly used shape signatures are centroid distance signature, tangent angle signature and complex coordinates signature, which we will discuss briefly here.

2.2.1.1 Centroid distance signature

The centroid distance signature $d(n)$ is calculated by the distance of the shape boundary points to the shape centroid (c_x, c_y) [40]:

$$d(n) = \left[(x(n) - c_x)^2 + (y(n) - c_y)^2 \right]. \quad (2.1)$$

The subtraction of the centroid makes the signature translation invariant. The centroid (c_x, c_y) is calculated as:

$$c_x = \frac{1}{N} \sum_{n=1}^N x(n), \quad (2.2)$$

and

$$c_y = \frac{1}{N} \sum_{n=1}^N y(n), \quad (2.3)$$

where N is the number of sampled points on the shape boundary.

2.2.1.2 Tangent angle signature

The tangent angle signature $\theta(n)$ at a point is defined as [40]:

$$\theta(n) = \theta_n = \arctan \frac{y(n) - y(n-w)}{x(n) - x(n-w)}, \quad (2.4)$$

where $(x(n), y(n))$ is the position of the point and w is a step of selected length. The tangent angle signature represents changes in the shape boundary directions, which are important to the human visual system.

2.2.1.3 Complex coordinates signature

The complex coordinates signature transform coordinates of boundary points, $P_n(x(n), y(n))$ into a complex function [40]:

$$z(n) = [x(n) - c_x] + i[y(n) - c_y]. \quad (2.5)$$

The complex coordinates signature is also translation invariant, because of the subtraction of the centroid.

2.2.2 Chain code and chain code histogram

The chain code representation [41] is a sequence of numbers that represents the directions required to trace the contour of the shape. The most common chain code representations are 4-directional or 8-directional. As shown in Figure 2.3, the direction of each movement along the shape contour is encoded by the numbering scheme, where $i = 0, 1, 2, 3$ or $i = 0, 1, \dots, 7$ represent the directions of a counter-clockwise angle $i \cdot \frac{\pi}{2}$ or $i \cdot \frac{\pi}{4}$ regarding the

positive x -axis. A more general chain code representation, which is N -directional ($N > 8$ and $N = 2^k$), has also been proposed in [42, 48].

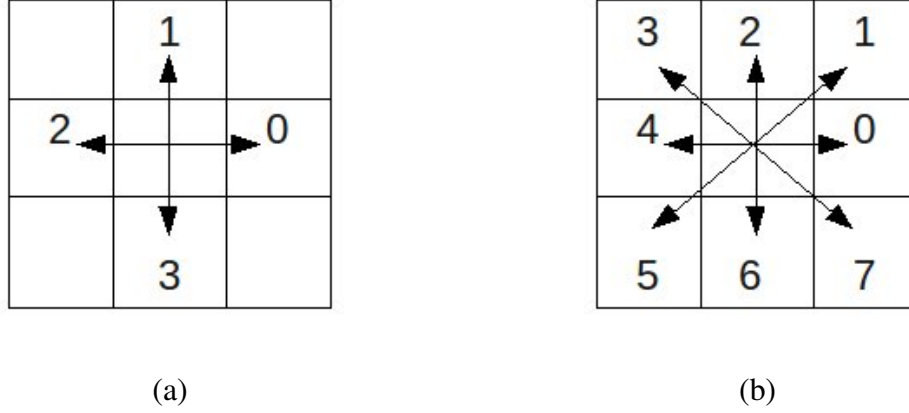


Figure 2.3: Chain code direction (a) Chain code in eight directions ; (b) chain code in four directions

The chain code histogram (CCH) [43] is a shape representation closely related to the chain code. In calculating CCH, an n -directional chain code is transformed into an n -dimensional histogram to reflect the probabilities of different directions present in a shape contour. Similarity measures such as, the Euclidean distance and the sum of absolute values can be used for comparing CCH shape descriptors.

2.2.3 Boundary moments

Moments contain important information about a function. The boundary moments are moments of the complex signature $z(n) = x + iy$, where x and y are the coordinates of the shape contour. The k th moment m_k and central moment μ_k are [44]:

$$m_k = \frac{1}{N} \sum_{i=1}^N [z(i)]^k \quad (2.6)$$

and

$$\mu_k = \frac{1}{N} \sum_{i=1}^N [z(i) - m_1]^k. \quad (2.7)$$

2.2.4 Simple shape descriptors

Five simple shape descriptors have been introduced in [45]. They are convexity, ratio of principal axes, compactness, circular variance, and elliptic variance.

2.2.4.1 Convexity

The minimal convex covering of an object is called a convex hull, as shown in Figure 2.4. Convexity is defined as the ratio of perimeters of the convex hull $P_{convexhull}$ over that of the original contour P [45]:

$$Convexity = \frac{P_{convexhull}}{P}. \quad (2.8)$$

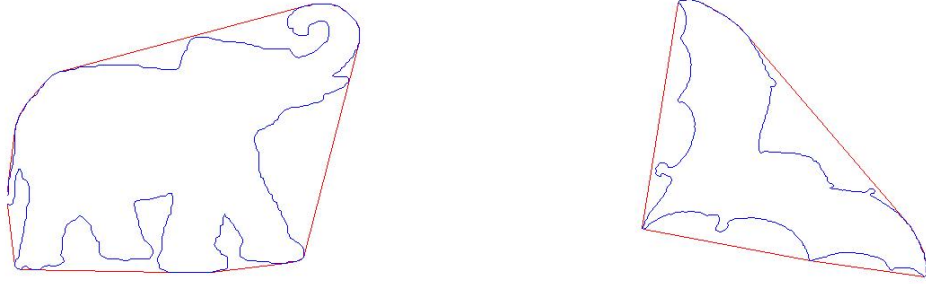


Figure 2.4: Shape contours and convex hulls

2.2.4.2 Ratio of principle axes (Eccentricity)

When the sampled points $\mathbf{x}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$ of the shape contour are treated as statistical data, the principle component direction and the secondary principle component direction of the data can be found, as shown in the samples in Figure 2.5. The covariance matrix C of those statistical data can also be calculated. The ratio of the principle axes or eccentricity, R_{pr_ax} is defined as the ratio between the lengths of the principle axes or the ratio between the two eigenvalues $\lambda_{1,2}$ of the covariance matrix C . If we denote the elements of the covariance matrix C as [45]:

$$C = \begin{pmatrix} c_{xx} & c_{xy} \\ c_{yx} & c_{yy} \end{pmatrix}, \quad (2.9)$$

the ratio of principle axes can also be calculated as[45]:

$$R_{pr_ax} = \frac{c_{yy} + c_{xx} - \sqrt{(c_{yy} + c_{xx})^2 - 4(c_{xx}c_{yy} - c_{xy}^2)}}{c_{yy} + c_{xx} + \sqrt{(c_{yy} + c_{xx})^2 - 4(c_{xx}c_{yy} - c_{xy}^2)}}. \quad (2.10)$$

This makes it possible to avoid the computation of eigenvalues.

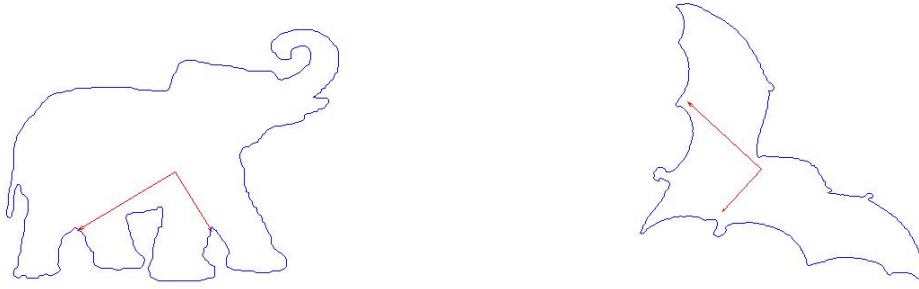


Figure 2.5: Principle and secondary axes of the shape contours

2.2.4.3 Compactness

The compactness of a contour is defined as the ratio of the squared perimeter of the contour and the area enclosed by the contour [45]. When the contour is a circle, the contour will have the smallest compactness value $\frac{(2\pi r)^2}{\pi r^2} = 4\pi$. Figure 2.6 shows shape contours and the circle with equal amounts of area enclosed.



Figure 2.6: Shape contour and the circle with equal amount of area enclosed

2.2.4.4 Circular variance

The circular variance is a measure of the proportional mean-squared error with respect to solid circle. It is defined as [45]:

$$cvar = \frac{1}{N\mu_r^2} \sum_i (\|\mathbf{p}_i - \mu\| - \mu_r)^2, \quad (2.11)$$

where N is the number of sample points, $\mu = \frac{1}{N} \sum_i \mathbf{p}_i$ is the centroid, and $\mu_r = \frac{1}{N} \sum_i \|\mathbf{p}_i - \mu\|$ is the mean variance. The circular variance of a circle is zero.

2.2.4.5 Elliptic variance

Elliptic variance is a measure of how much the shape fits in an ellipse with the same covariance matrix as that of the shape, i.e., $C = C_{ellipse}$. It is defined as:

$$evar = \frac{1}{N\mu_{rc}} \sum_i \left(\sqrt{(\mathbf{p}_i - \mu)^T C^{-1} (\mathbf{p}_i - \mu)} - \mu_{rc} \right)^2, \quad (2.12)$$

where

$$\mu_{rc} = \frac{1}{N} \sum_i \sqrt{(\mathbf{p}_i - \mu)^T C^{-1} (\mathbf{p}_i - \mu)}. \quad (2.13)$$

Table 2.1, show shapes and their simple shape descriptor values. As we can see from the table, simple shape descriptors have their limitations: different shapes could have very

similar simple shape descriptor values. Simple shape descriptors can only be used to discriminate shapes with large differences. However, as they are simple and easy to extract, they are suitable to be used as prefilters, and to be used together with other shape descriptors.


























	Convexity	Eccentricity	Compactness	Circular Variance	Elliptic Variance
	0.8375	0.6957	18.0863	0.0248	0.0398
	0.8074	0.3342	38.4434	0.1683	0.1697
	0.4499	0.4271	96.6635	0.1665	0.2017
	0.8833	0.6436	21.4946	0.0351	0.0375
	0.7275	0.5099	34.2587	0.0999	0.1047
	0.8827	0.0472	31.6433	0.2423	0.2641
	0.9693	0.0963	28.8950	0.1811	0.0459
	0.9038	0.1791	24.7461	0.1186	0.0327
	0.6925	0.4609	42.5661	0.1965	0.1967
	0.6091	0.3782	36.4780	0.0770	0.0843
	0.8884	0.2532	22.9290	0.0951	0.0294
	0.6721	0.1776	38.7590	0.2212	0.1702
	0.6975	0.2761	30.0804	0.1078	0.1079
	0.9525	0.1942	21.9372	0.1088	0.0297
	0.6411	0.6001	34.0558	0.0956	0.1274
	0.8999	0.1161	24.3948	0.1543	0.0309
	0.7207	0.2277	37.1684	0.1270	0.0747
	0.8322	0.1945	26.3032	0.1216	0.0498
	0.9691	0.2367	20.9168	0.0971	0.0535
	0.5892	0.5135	38.9036	0.1229	0.1712
	0.7059	0.6821	24.0706	0.0805	0.1100
	0.4528	0.3269	56.4532	0.1012	0.1133
	0.6887	0.9882	53.5000	0.1144	0.1531
	0.8141	0.9888	26.7089	0.0249	0.0348
	0.4826	0.9578	98.9508	0.0918	0.1240

Table 2.1: Shapes and their simple shape descriptor values

2.2.5 Curvature scale space (CSS) representation

The curvature scale space shape representation [46] was introduced by Abbasi et al. Based on the scale space theory, the contour of a shape is convoluted with 1D Gaussian functions with increasing variances. The curvatures of the evolving shape contour are calculated to form the CSS image. The maxima of the CSS image is the CSS shape representation.

2.2.5.1 CSS representation

The curvature of a plane curve is defined as

$$\kappa(s) = \lim_{h \rightarrow 0} \frac{\phi}{h}, \quad (2.14)$$

where ϕ is the angle between $\mathbf{t}(s)$ and $\mathbf{t}(s+h)$. \mathbf{t} represents the tangent vector and s is the arc length parameter. Curvature-zero crossings of a curve are points where $\kappa = 0$.

When the curve of the shape contour is represented as:

$$\Gamma(u) = (x(u), y(u)), \quad (2.15)$$

where u is an arbitrary parameter. The curvature function is calculated using the following formula:

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{\left(\dot{x}^2(u) + \dot{y}^2(u)\right)^{\frac{3}{2}}}, \quad (2.16)$$

where $\dot{x}(u)$, $\dot{y}(u)$ are the first and $\ddot{x}(u)$, $\ddot{y}(u)$ are the second derivatives of the components $x(u)$ and $y(u)$ with respect to the parameter u .

In extracting the CSS representation, curve smoothing is performed prior to curvature measurement to reduce the effects of noise. Each coordinate of the curve is convoluted with a 1-D Gaussian function $g(u, \sigma)$. The resulting smoothed contour is $\Gamma_\sigma(u) = (X(u, \sigma), Y(u, \sigma))$, where

$$X(u, \sigma) = x(u) * g(u, \sigma) \quad (2.17)$$

and

$$Y(u, \sigma) = y(u) * g(u, \sigma). \quad (2.18)$$

According to the properties of convolution, the first derivative of the components, $X_u(u, \sigma)$ and $Y_u(u, \sigma)$, can be calculated as:

$$X_u(u, \sigma) = x(u) * g_u(u, \sigma) \quad (2.19)$$

and

$$Y_u(u, \sigma) = y(u) * g_u(u, \sigma), \quad (2.20)$$

where $g_u(u, \sigma)$ is the first derivative of the Gaussian function. Similarly, the second derivative of the components, $X_{uu}(u, \sigma)$ and $Y_{uu}(u, \sigma)$, can be calculated as:

$$X_{uu}(u, \sigma) = x(u) * g_{uu}(u, \sigma) \quad (2.21)$$

and

$$Y_{uu}(u, \sigma) = y(u) * g_{uu}(u, \sigma), \quad (2.22)$$

where $g_{uu}(u, \sigma)$ is the second derivative of the Gaussian function. The curvature of the smoothed curve can be calculated as

$$\kappa(u, \sigma) = \frac{X_u(u, \sigma)Y_{uu}(u, \sigma) - X_{uu}(u, \sigma)Y_u(u, \sigma)}{\left(X_u(u, \sigma)^2 + Y_u(u, \sigma)^2\right)^{\frac{3}{2}}}. \quad (2.23)$$

Figure 2.7 shows the curve Γ_σ of the contours, corresponding to the two duck shapes. These two duck shapes are actually the rotated version of each other. The curve Γ_σ are smoothed after being convolved with Gaussian functions of increasing width σ . As the value of σ increases the curve Γ shrinks and becomes smoother. That process is called the evolution of Γ .

The CSS image of the curve is generated by displaying the locations of curvature zero-crossings of every Γ_σ in the (u, σ) plane, where u is the normalized arc length and σ is

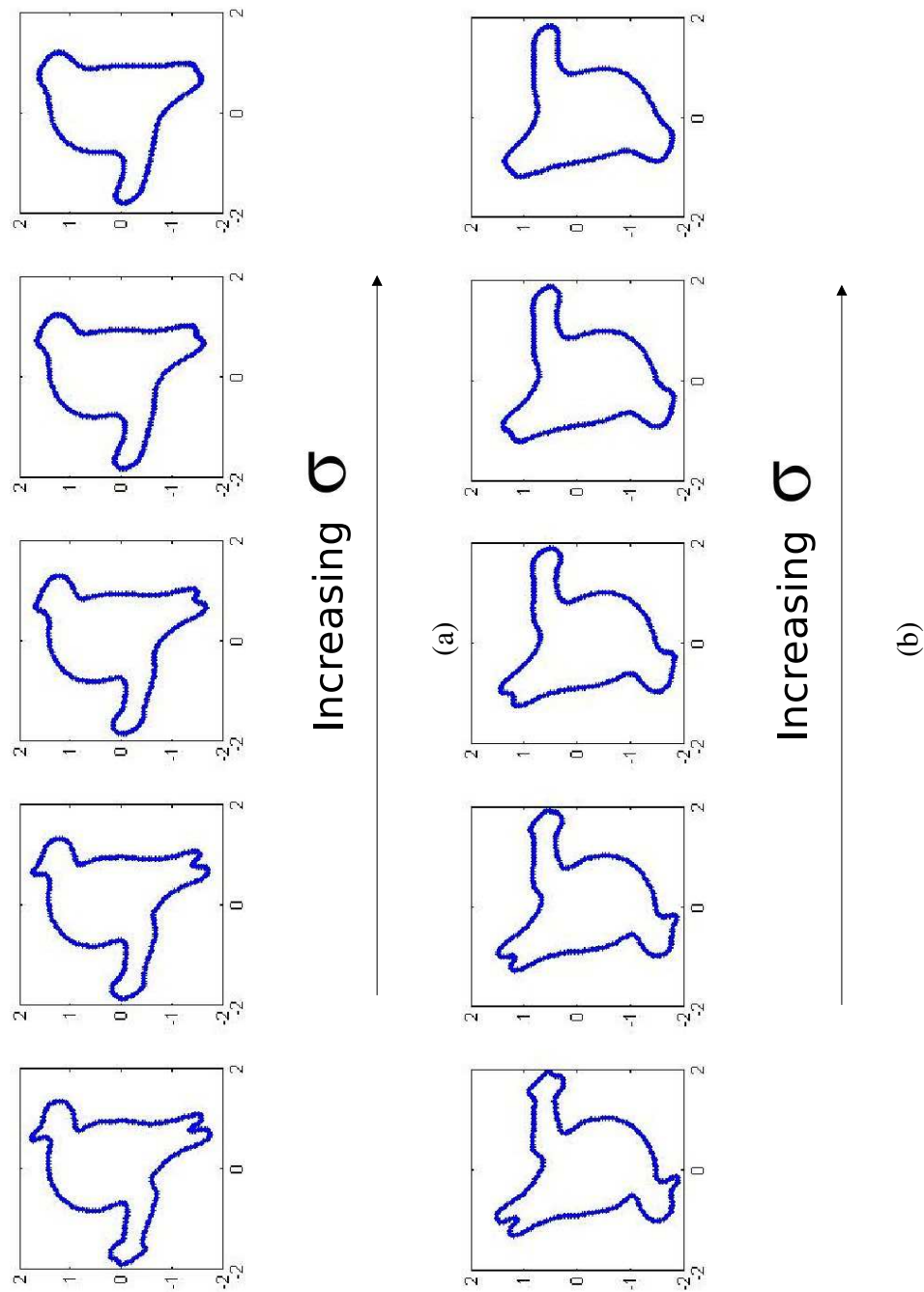
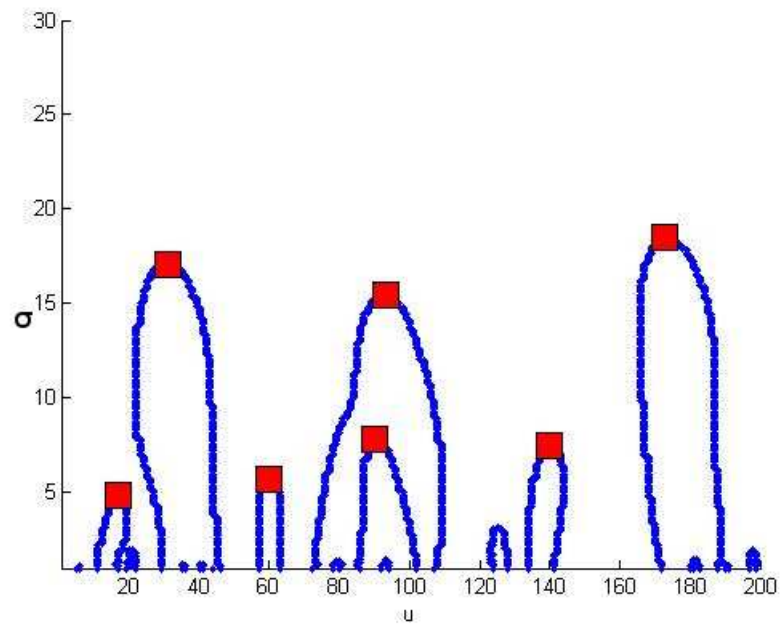
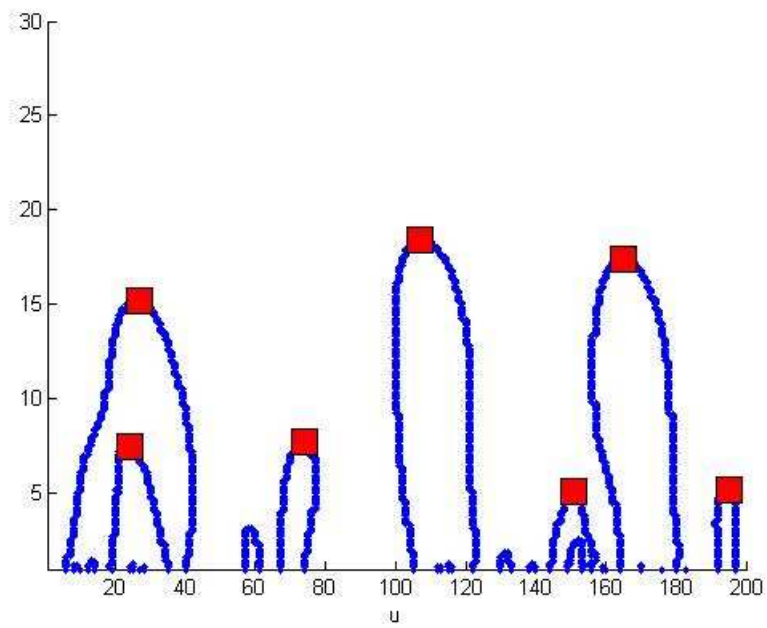


Figure 2.7: Shrinkage and smoothing of the contour curve, as σ increases

the width of the Gaussian function. In a CSS image, the intersection of a horizontal line with the contour indicates the location of the curvature zero-crossings on the corresponding evolved curve. Figure 2.8 shows two CSS images, corresponding to the two duck shapes used in Figure 2.7. On each of the contours in a CSS images, there are usually two curvature zero-crossing points. As σ increases, the two curvature zero-crossing points of a contour merge to one point at the top of the contour. That point is called the CSS maximum. The maxima in a CSS image are used to represent the corresponding shape. Those maxima less than one sixth of the largest maximum of the same CSS image, are excluded in the CSS shape representation, as they are usually related to noise or small ripples of the curve. Each maximum will need two integer numbers to represent the height and the location. The CSS shape representation usually consists of several pairs of integer numbers. The examples in Figure 2.8 will need seven pairs of integer numbers. The size of the CSS representation is relatively small in comparison with other representations. If a shape is rotated, its corresponding CSS representation will shift circularly, as shown in the examples in Figure 2.8. Because of the arc length normalization and the same number of sampling points are used, scaling will not affect the CSS representation.



(a)



(b)

Figure 2.8: The CSS images and the maxima (the maxima are indicated using red squares)

2.2.5.2 CSS distance measure

In calculating the distance between two CSS shape representations, the two sets of CSS maxima are horizontally circular shifted to make the best possible match. The summation of the Euclidean distances between the relevant pairs of maxima is the matching value. The effect of mirroring is also considered, by flipping one of the CSS image horizontally and calculating the CSS image matching value again. The smaller one of the two CSS image matching values is chosen as the final CSS descriptor distance value.

2.3 Region-based shape representation and their extractions

Contour-based shape representation and retrieval methods can be used for simple shape image retrieval, but not for complex shape image retrieval. Region-based methods, however, do not have such limitations. In this section, the following commonly used region-based shape representations will be reviewed:

1. *The Fourier-wavelet descriptor (FWD)* [49]: It plots the polar image of a shape into Cartesian coordinates, and treats the radial and angular axes as the new x and y axes. In order to extract the shape descriptor, FWD applies the 1-D Fourier transform and the 1-D wavelet transform along the new *angular*-axis and along the new *radial*-axis, respectively.
2. *The generic Fourier descriptor (GFD)* [50]: It also plots the polar image of a shape into Cartesian coordinates, and treats the radial and angular axes as the new x and y axes. Different from the FWD, the GFD applies 2-D Fourier transform on the new image to extract the shape descriptor.
3. *The R-transform* [51] shape representation: It is based on the 2D Radon transform [52], and is the integral of the square of the the 2D Radon transform of a shape image.
4. *The geometric moment descriptor (GMD)* [53]: It is extracted by projecting shape image as a 2D function onto real polynomial bases

5. *The complex moment descriptor (CMD)* [54]: It is extracted by projecting a shape image onto complex polynomial bases, instead of onto real polynomial bases as in extracting GMD.

The Zernike moment shape descriptor [55, 56], is a special case of complex moment descriptor, and has shown robust performance as a shape descriptor. As it is also used in developing the affine invariant ICA Zernike moment shape descriptor [38, 39], it will be discussed in Chapter 6.

2.3.1 Fourier-wavelet descriptor

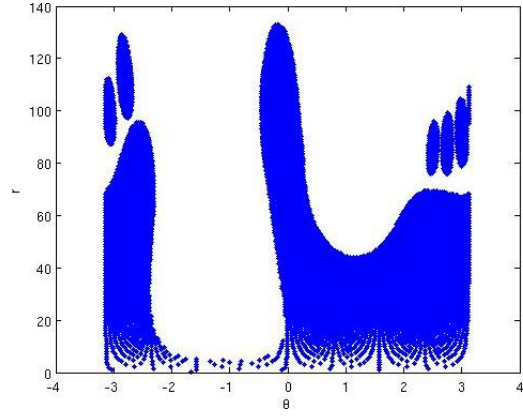
The Fourier-wavelet descriptor (FWD) [49] is a shape descriptor in the transformed domain. To obtain the shape descriptor, the Fourier transform and wavelet transform are applied once along the angular direction and along the radial direction, respectively. There are three major steps in extracting the Fourier-wavelet descriptor from shape image [49] :

1. Transform the shape image $f(x, y)$ into polar image $g(r, \theta)$;
2. Plot the polar image $g(r, \theta)$ into Cartesian coordinate, as shown in Figure 2.9 and Figure 2.10.
3. Apply 1-D Fourier transform on $g(r, \theta)$ along the axis of polar angle θ and get its spectrum: $G(r, \phi) = \|FT_{\theta}(g(r, \theta))\|$.
4. Apply 1-D wavelet transform on $G(r, \phi)$ along the axis of radius r and get the wavelet coefficients: $WF(r, \phi) = WT_r(G(r, \phi))$.

The wavelet coefficients extracted at step 4 are the final shape descriptor, and is used for shape similarity comparison.



(a)Complex shape image

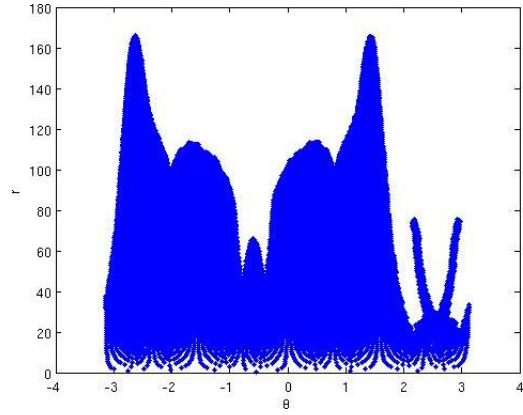


(b)Polar image in Cartesian coordinates

Figure 2.9: Complex shape image and its corresponding polar image in Cartesian coordinates



(a)Simple shape image



(b)Polar image in Cartesian coordinates

Figure 2.10: Simple shape image and its corresponding polar image in Cartesian coordinates

2.3.2 Generic Fourier descriptor

The steps of extracting the generic Fourier descriptor (GFD) [50] is similar to the steps of extracting the FWD. In extracting the GFD, the shape image is also first transformed into a polar image, and then plotted into Cartesian coordinates. The difference from the previous method is that rather than performing a 1D Fourier transform and a 1D wavelet transform

along the two coordinate axes, a single 2D Fourier transform is performed instead, on the polar image in Cartesian coordinates.

2.3.3 R-transform representation

R-transform [51] is a shape representation, based on the 2D Radon transform [52]. The 2D Radon transform includes the integral of a function over straight lines. If we define $T_{R_f}(\rho, \theta)$ as a 1-D projection of a function $f(x, y)$ at an angle θ , $T_{R_f}(\rho, \theta)$ is the integral of the function $f(x, y)$, along a line l that is distance ρ from the origin and at angle θ off the x -axis, i.e.,

$$T_{R_f}(\rho, \theta) = \int_l f(x, y) dl. \quad (2.24)$$

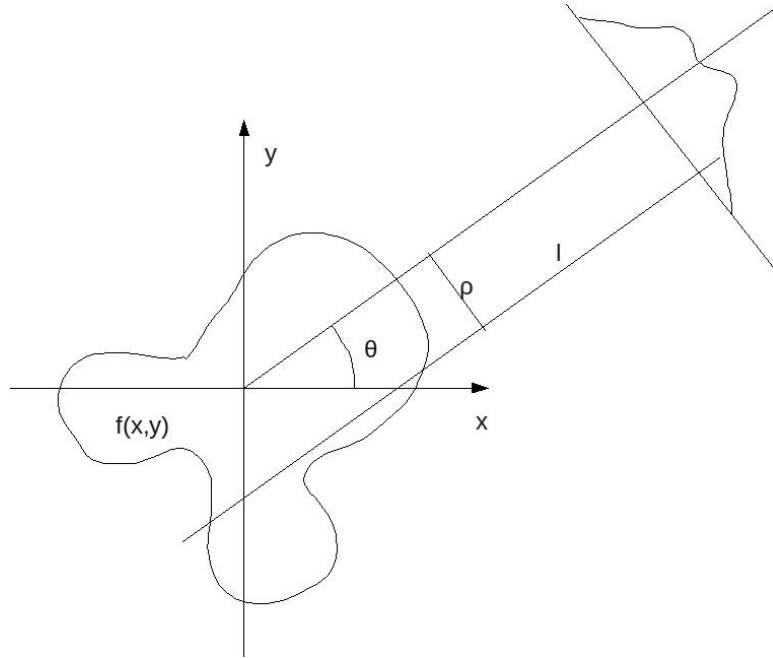


Figure 2.11: Radon transform, projection along line l

Since all points on the line satisfy the equation :

$$\rho = x \sin \theta - y \cos \theta, \quad (2.25)$$

the projection function $T_{R_f}(\rho, \theta)$ can be rewritten as

$$T_{R_f}(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \sin \theta - y \cos \theta - \rho) dx dy. \quad (2.26)$$

The R-Transform is defined as:

$$R_f(\theta) = \int_{-\infty}^{\infty} T_{R_f}^2(\rho, \theta) d\rho, \quad (2.27)$$

and is the integral of the square of the projection function $T_{R_f}(\rho, \theta)$. Figure 2.12 and Figure 2.13, show examples of simple images, their corresponding Randon transform and their corresponding R-transform. Figure 2.14 and Figure 2.15, show examples of complex images, their corresponding Randon transform and their corresponding R-transform.

2.3.4 Geometric moment descriptor

Moments are used to characterize a function and to capture its significant features. They can be obtained by projections of a function onto a polynomial basis. In 2D shape representation, the most commonly used polynomial basis is $x^i y^j$, which leads to the geometric moments [53]:

$$m_{pq} = \iint_G x^p y^q f(x, y) dx dy. \quad (2.28)$$

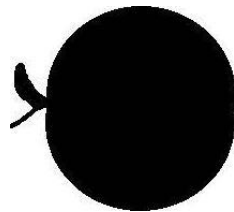
Their corresponding central moments are:

$$\mu_{pq} = \iint_G (x - x_c)^p (y - y_c)^q f(x, y) dx dy. \quad (2.29)$$

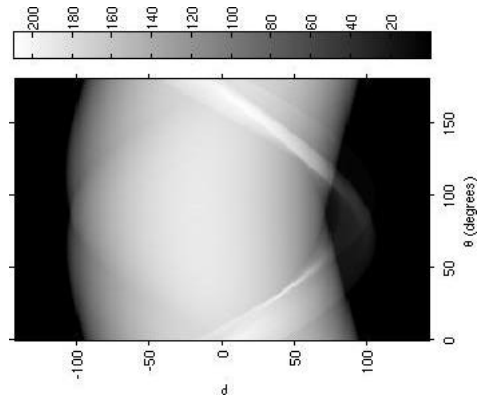
And the discrete forms are:

$$\mu_{pq} = \sum_x \sum_y (x - x_c)^p (y - y_c)^q f(x, y). \quad (2.30)$$

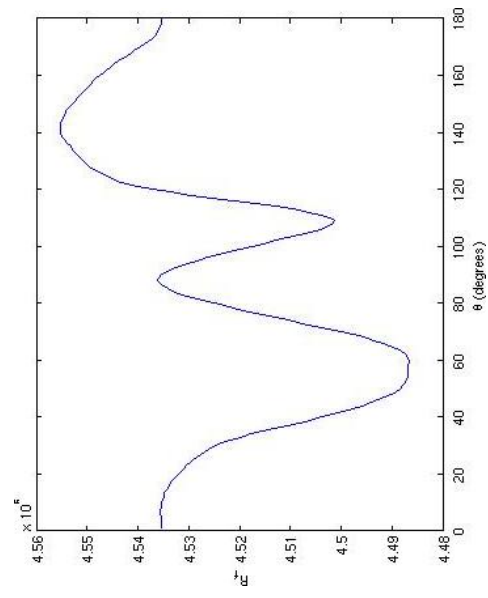
Low order geometric moments often have physical meanings. For binary shape images, m_{00} is the area of the shape image; m_{10} and m_{01} are the centroid of the shape; m_{20} and m_{02}



(a) Shape Image



(b) Radon transform

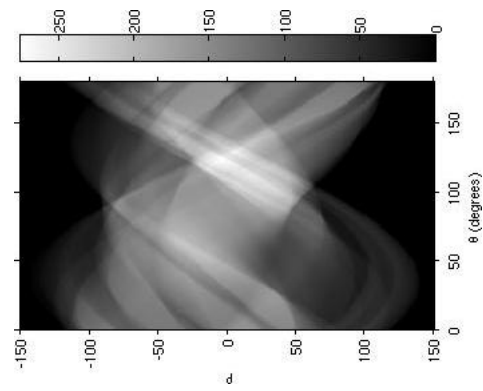


(c) R-transform

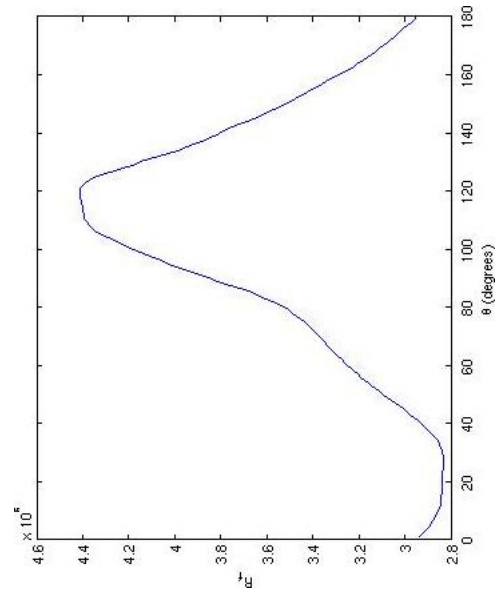
Figure 2.12: Shape image, its corresponding Radon transform and R-transform



(a) Shape image

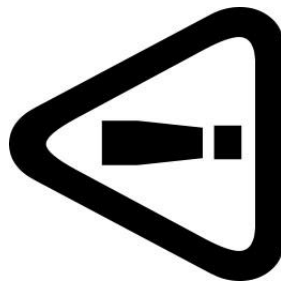


(b) Radon transform

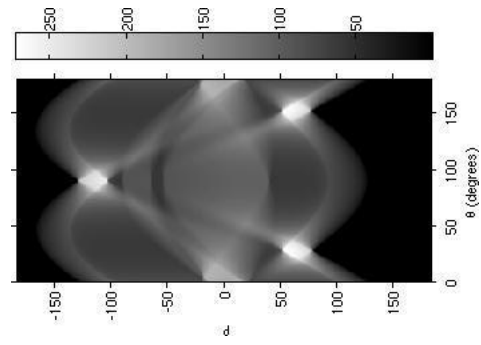


(c) R-transform

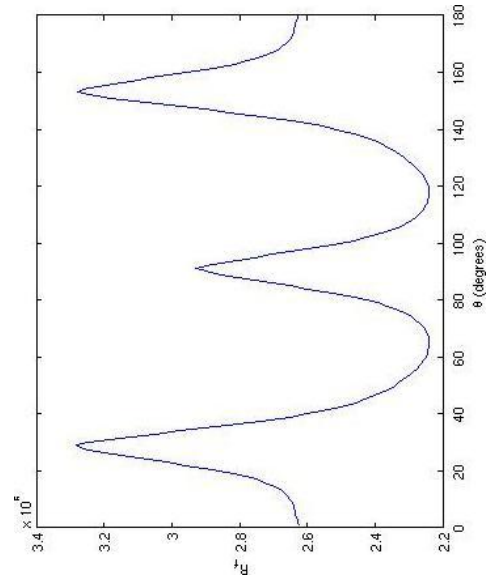
Figure 2.13: Simple shape image, its corresponding Radon transform and R-transform



(a) Shape image



(b) Radon transform

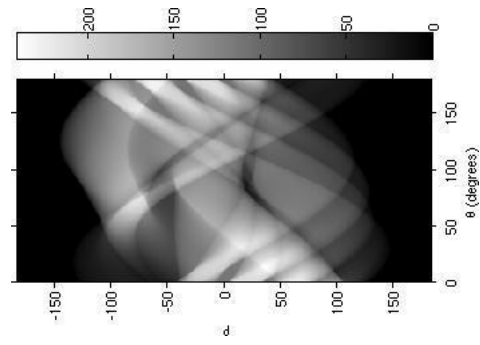


(c) R-transform

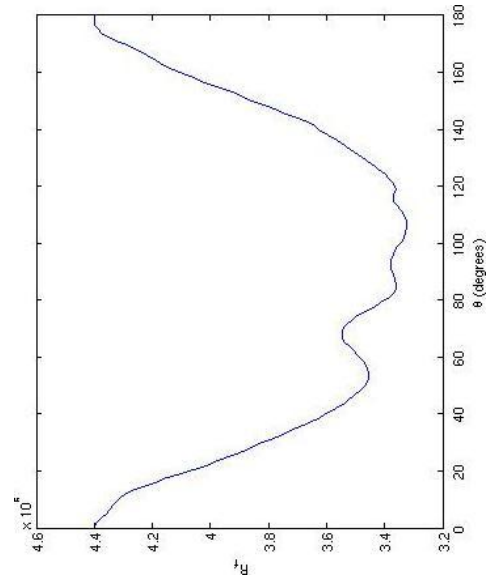
Figure 2.14: Simple shape image, its corresponding Radon transform and R-transform



(a) Shape image



(b) Radon transform



(c) R-transform

Figure 2.15: Complex shape image, its corresponding Radon transform and R-transform

are the variance of the shape along the two coordinates; m_{11} are the covariance between them.

In his pioneering work, Hu [53] derived seven famous moments:

$$\Phi_1 = v_{20} + v_{02}, \quad (2.31)$$

$$\Phi_2 = (v_{20} - v_{02})^2 + 4v_{11}, \quad (2.32)$$

$$\Phi_3 = (v_{30} - 3v_{12})^2 + (3v_{21} - v_{03})^2, \quad (2.33)$$

$$\Phi_4 = (v_{30} + v_{12})^2 + (v_{21} + v_{03})^2, \quad (2.34)$$

$$\begin{aligned} \Phi_5 = & (v_{30} - 3v_{12})(v_{30} + v_{12})[(v_{30} + v_{12})^2 - 3(v_{21} + v_{03})^2] \\ & + (3v_{21} - v_{03})(v_{21} + v_{03})[3(v_{30} + v_{12})^2 - (v_{21} + v_{03})^2], \end{aligned} \quad (2.35)$$

$$\begin{aligned} \Phi_6 = & (v_{20} - v_{02})[(v_{30} + v_{12})^2 - (v_{21} + v_{03})^2] \\ & + 4v_{11}(v_{30} + v_{12})(v_{21} + v_{03}), \end{aligned} \quad (2.36)$$

$$\begin{aligned} \Phi_7 = & (3v_{21} - v_{30})(v_{30} + v_{12})[(v_{30} + v_{12})^2 - 3(v_{21} + v_{03})^2] \\ & + (3v_{12} - v_{03})(v_{21} + v_{03})[3(v_{30} + v_{12})^2 - (v_{21} + v_{03})^2], \end{aligned} \quad (2.37)$$

where $v_{pq} = \mu_{pq}/(\mu_{00})^{(2+p+q)/2}$ for $p+q=2,3,\dots$. Those seven moments are invariant to rotation, translation and scale change.

2.3.5 Complex moment descriptor

Another popular choice of the polynomial basis is $(x + iy)^k (x - iy)^j$, which leads to the geometric moment descriptor [54]:

$$c_{pq} = \iint_G (x + iy)^p (x - iy)^q f(x, y) dx dy. \quad (2.38)$$

Complex moments can be expressed using the geometric moments of the same order as

$$c_{pq} = \sum_{k=0}^p \sum_{j=0}^q \binom{p}{k} \binom{q}{j} (-1)^{q-j} i^{p+q-k-j} m_{k+j, p+q-k-j}, \quad (2.39)$$

and vice versa

$$m_{pq} = \frac{1}{2^{p+q} i^q} \sum_{k=0}^p \sum_{j=0}^q \binom{p}{k} \binom{q}{j} (-1)^{q-j} c_{k+j, p+q-k-j}. \quad (2.40)$$

2.4 Summary

In this chapter, previously known shape representations and their extractions have been reviewed. Generally speaking, they can be divided into two categories: contour-based and region-based. Shape signatures, chain code histograms and contour scale-space are all examples of contour-based shape representations while the Fourier-wavelet, generic Fourier, R-transform and the moment descriptors are examples of region-based representations. Contour-based shape representations can only describe simple shapes, while region-based shape representations can describe both simple and complex ones. Both contour-based and region-based shape representations have been reviewed in this chapter to give a general picture of shape features and their extraction.

Although regular shape representations are not affine invariant, they have inspired the development of affine invariant ones. Some affine invariant shape representations are modified from regular shape representations. This will be evident, when we review and evaluate previously known affine invariant shape representations, in the next chapter.

Chapter 3

Review and Performance Study of Previous Affine-Invariant Shape Representations

3.1 Introduction

In the last chapter, we reviewed different regular shape representations. Although those representations are not affine-invariant, they are sometimes related to the affine-invariant ones. As the goal of this research is to develop new and robust affine-invariant shape representations, previously known ones need to be studied and evaluated first.

Various affine-invariant shape representations have been proposed in the past. In [57, 58, 59], affine-invariant shape descriptor have been derived from the Fourier transform of the object contour. In [1, 60, 61], wavelet transform of the object contour has been used to derive affine-invariant functions. In [62, 63], curvature scale space was used in affine-invariant shape-based retrieval. In [64, 65], independent component analysis was used to derive affine-invariant shape signatures. In [66, 67, 68, 69, 70], different affine-invariant moments were developed.

The rest of the chapter is organized as follows: In Section 3.2, we will first review the definition of the affine transformation; In Section 3.3, we will review two important affine-invariant parametrization methods. From Section 3.4 to 3.8, five previously known

affine-invariant shape representations are described in detail. Section 3.9, compares the retrieval experiment performances of those representations. The last section summarizes this chapter.

3.2 Affine transformation

A shape image of the object taken from a view point could be linked with another shape image of the same object approximately by an affine transformation, if it is viewed from a larger distance than its size along the line of sight [34]. Since images in a database are often geometrically distorted by the change of viewpoints, the affine transform is of great importance in image analysis.

Using a vector-matrix notation, an affine transformation can be defined as:

$$\mathbf{x}_a = \mathbf{A}\mathbf{x} + \mathbf{T}, \quad (3.1)$$

where $\mathbf{x} = [x, y]^T$ and $\mathbf{x}_a = [x_a, y_a]^T$ are the vectors that contain the coordinates of the original and the affine transformed shape images, respectively. The 2×2 nonsingular matrix \mathbf{A} can be decomposed as follow:

$$\mathbf{A} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \quad (3.2)$$

where s , α and θ represent scale, skew and rotation, respectively. s_x and s_y are the scaling factors in the x and the y directions, respectively. When s_x and s_y are of different signs, it corresponds to a reflection. The 2×1 vector $\mathbf{T} = [T_x, T_y]^T$ represents the translation. When the effect of translation is eliminated by setting the origin of the coordinate system to the centroid of the shape, Equation 3.1 reduces to:

$$\mathbf{x}_a = \mathbf{A}\mathbf{x}. \quad (3.3)$$

3.3 Affine-invariant parameters

The arc length parameter, which is normally used in regular contour-based shape representations, is transformed non-linearly under an affine transformation. However, most of the contour-based affine-invariant shape recognition methods will require a parameter that is *linear* under an affine transformation. For example, the affine-invariant Fourier shape descriptor [57], utilizes the property of the Fourier transform that preserves the linearity of affine transform. That property is valid, if and only if the parameter transformation is also *linear*. Therefore, the arc length parameter that is *not linear* under affine transformation, is not suitable to parametrize shape contours under affine transformation. To overcome that problem, affine-invariant parameters need to be used. There are two kinds of affine-invariant parameters that are linear under an affine transformation: One is the affine arc length [71], the other is the enclosed area parameter [57].

3.3.1 Affine arc length

The affine arc length τ is defined as [71]:

$$\tau = \int_a^b \sqrt[3]{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)} dt, \quad (3.4)$$

where $\dot{x}(t)$, $\dot{y}(t)$ are the first derivatives and $\ddot{x}(t)$, $\ddot{y}(t)$ are the second derivatives of the components of $x(t)$ and $y(t)$ with respect to the arc length parameter t . The limits a and b denote the beginning and the end of a segment of the contour. It can be made completely invariant by normalizing it with respect to the total affine arc length of the contour. The main disadvantage of the affine arc length is that its computation requires second order derivatives, which are susceptible to noise. A more stable and popular affine-invariant parameter is the enclosed area parameter.

3.3.2 Enclosed area parameter

The enclosed area parameter is derived based on the property of the affine transform that all areas are changed by the same ratio under an affine mapping [72]. For example, in Figure 3.1, the areas of all the sectors in the transformed object are transformed by the same ratio

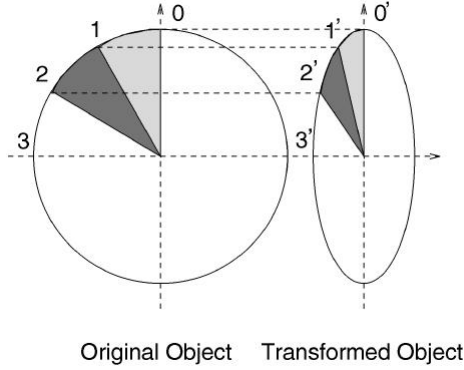


Figure 3.1: All areas are changed in the same ratio under an affine mapping (Figure from [1])

to the areas of the corresponding sectors in the original object, while the segments of the contour are not transformed in the same ratio to the segments of the original contour.

The enclosed area parameter η , which is linear under an affine transformation, is derived [57]:

$$\eta = \frac{1}{2} \int_a^b |x(t)\dot{y}(t) - y(t)\dot{x}(t)| dt, \quad (3.5)$$

where $x(t)$ and $y(t)$ are the coordinates of the points on the contour. The origin of the coordinate system is set at the centroid of the contour. The enclosed area parameter η is a sum of the triangular areas which are created by linking the two points on the contour and the origin. It can be made completely invariant by normalizing it with respect to the total enclosed area of the contour. Since calculating the enclosed area parameter does not require the use of the second derivative, it avoids all the shortcomings and it is currently more widely used in contour-based affine-invariant shape representations.

3.4 Affine-invariant Fourier shape descriptor

The affine-invariant Fourier shape descriptor (AIFSD), which uses the enclosed area parameter, was first proposed in [57]. When the effect of translation is eliminated by setting the origin of the coordinate system to the centroid of the object contour, we have:

$$\begin{bmatrix} x_a(n) \\ y_a(n) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x(n) \\ y(n) \end{bmatrix}. \quad (3.6)$$

After applying the Fourier transform to both side of Equation 3.6, we have

$$\begin{bmatrix} X_k^a \\ Y_k^a \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} X_k \\ Y_k \end{bmatrix}, \quad (3.7)$$

where X_k and Y_k represent the k th Fourier coefficients resulting from the Fourier transform on $x(n)$ and $y(n)$. Similarly, we can get the equation for the p th Fourier coefficients:

$$\begin{bmatrix} X_p^a \\ Y_p^a \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} X_p \\ Y_p \end{bmatrix}. \quad (3.8)$$

Combining Equation 3.7 and Equation 3.8, we can get:

$$\begin{bmatrix} X_k^a & (X_p^a)^* \\ Y_k^a & (Y_p^a)^* \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} X_k & X_p^* \\ Y_k & Y_p^* \end{bmatrix}. \quad (3.9)$$

After taking the determinants of both sides of Equation 3.9, we have:

$$\det \begin{bmatrix} X_k^a & (X_p^a)^* \\ Y_k^a & (Y_p^a)^* \end{bmatrix} = \det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \det \begin{bmatrix} X_k & X_p^* \\ Y_k & Y_p^* \end{bmatrix}. \quad (3.10)$$

When $k = p$, from Equation 3.10, we have:

$$\det \begin{bmatrix} X_p^a & (X_p^a)^* \\ Y_p^a & (Y_p^a)^* \end{bmatrix} = \det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \det \begin{bmatrix} X_p & X_p^* \\ Y_p & Y_p^* \end{bmatrix}. \quad (3.11)$$

Combining Equation 3.10 and Equation 3.11, we have

$$Q_k = \frac{\det \begin{bmatrix} X_k^a & (X_p^a)^* \\ Y_k^a & (Y_p^a)^* \end{bmatrix}}{\det \begin{bmatrix} X_p^a & (X_p^a)^* \\ Y_p^a & (Y_p^a)^* \end{bmatrix}} = \frac{\det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \det \begin{bmatrix} X_k & X_p^* \\ Y_k & Y_p^* \end{bmatrix}}{\det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \det \begin{bmatrix} X_p & X_p^* \\ Y_p & Y_p^* \end{bmatrix}} = \frac{\det \begin{bmatrix} X_k & X_p^* \\ Y_k & Y_p^* \end{bmatrix}}{\det \begin{bmatrix} X_p & X_p^* \\ Y_p & Y_p^* \end{bmatrix}}. \quad (3.12)$$

or

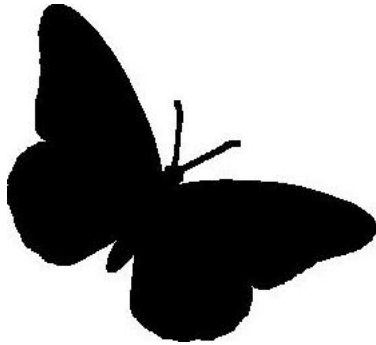
$$Q_k = \frac{X_k^a Y_p^{a*} - Y_k^a X_p^{a*}}{X_p^a Y_p^{a*} - Y_p^a X_p^{a*}} = \frac{X_k Y_p^* - Y_k X_p^*}{X_p Y_p^* - Y_p X_p^*}. \quad (3.13)$$

As the determinant of the affine transform matrix:

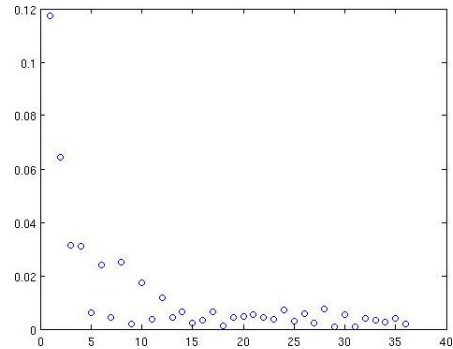
$$\det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

is canceled out and not included in Q_k , Q_k is affine-invariant.

Figures 3.2-3.5 show shape images and their corresponding AIFSDs. The shape images in Figure 3.2 and Figure 3.3 are affine related, so are the shape images in Figure 3.4 and Figure 3.5. As we can see, the AIFSDs of the affine related shape images are almost the same, while the AIFSDs of the unrelated shape images are not the same. So, the AIFSD can be used to discriminate between affine related shape images and unrelated ones.

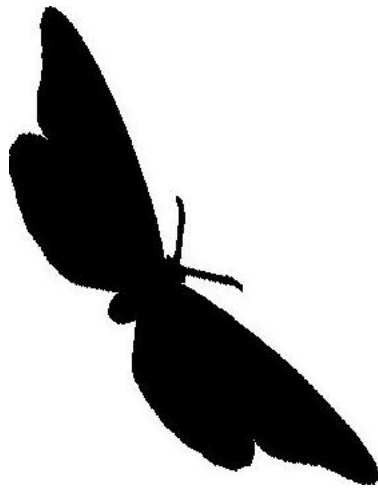


(a) shape image

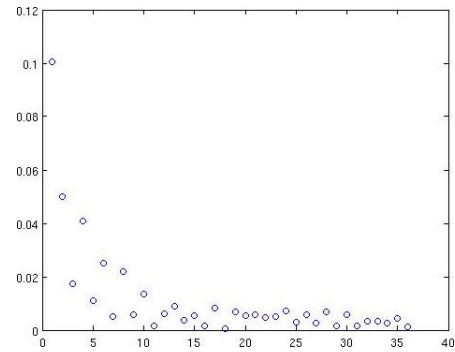


(b) affine-invariant Fourier shape descriptor

Figure 3.2: Shape image and its corresponding AIFSD

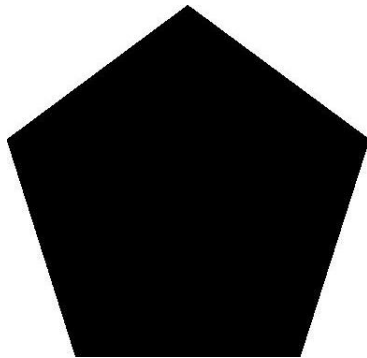


(a)shape image

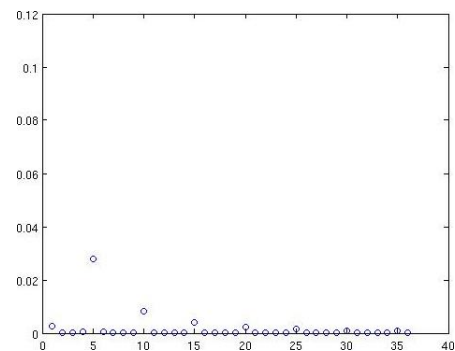


(b)affine-invariant Fourier shape descriptor

Figure 3.3: Shape image and its corresponding AIFSD



(a)shape image



(b)affine-invariant Fourier shape descriptor

Figure 3.4: Shape image and its corresponding AIFSD

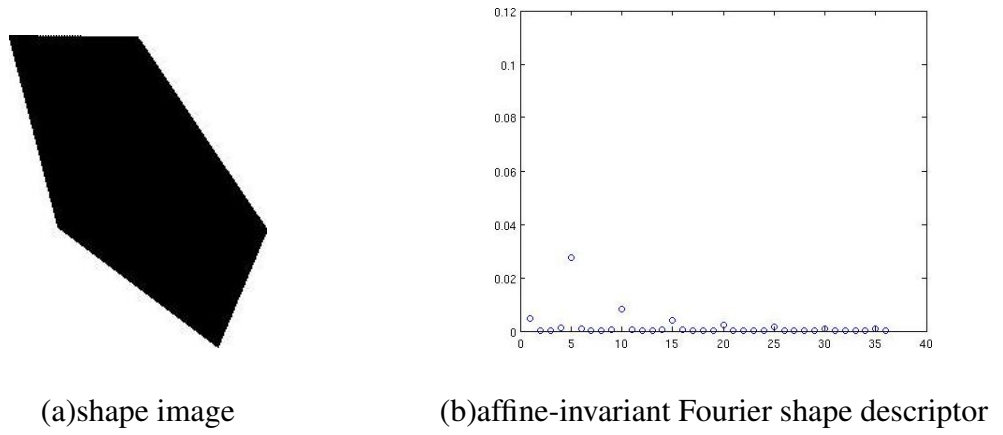


Figure 3.5: Shape image and its corresponding AIFSD

3.5 Affine-invariant wavelet-based shape representation

The affine-invariant wavelet-based shape representation (AIWSR) was first introduced in [1], and has since been modified to other wavelet based representations, such as those in [61].

In extracting the AIWSR, the origin of the coordinate system is first moved to the centroid of the object contour, so that the effect of translation is eliminated. This results with:

$$\begin{bmatrix} x_a(n) \\ y_a(n) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x(n) \\ y(n) \end{bmatrix} \quad (3.14)$$

By applying undecimated wavelet transform [73, 74, 75, 76] to both sides of Equation 3.14, we have:

$$\begin{bmatrix} A_j^{x_a}(k) & W_j^{x_a}(k) \\ A_j^{y_a}(k) & W_j^{y_a}(k) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} A_j^x(k) & W_j^x(k) \\ A_j^y(k) & W_j^y(k) \end{bmatrix}, \quad (3.15)$$

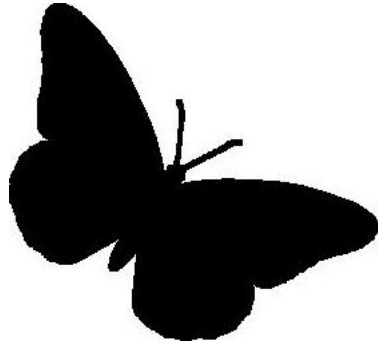
where A_j and W_j denotes the approximation and the detail wavelet coefficients at a particular resolution level j . Similarly, we can also get the equation at resolution level p :

$$\begin{bmatrix} A_p^{x_a}(k) & W_p^{x_a}(k) \\ A_p^{y_a}(k) & W_p^{y_a}(k) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} A_p^x(k) & W_p^x(k) \\ A_p^y(k) & W_p^y(k) \end{bmatrix}, \quad (3.16)$$

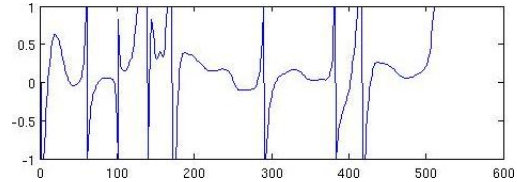
By taking the determinants on both sides of Equation 3.15 and Equation 3.16, and combining the results together, we can get the wavelet based affine-invariant shape representation:

$$M_j(k) = \frac{A_j^{x_a}(k)W_j^{y_a}(k) - A_j^{y_a}(k)W_j^{x_a}(k)}{A_p^{x_a}(k)W_p^{y_a}(k) - A_p^{y_a}(k)W_p^{x_a}(k)} = \frac{A_j^x(k)W_j^y(k) - A_j^y(k)W_j^x(k)}{A_p^x(k)W_p^y(k) - A_p^y(k)W_p^x(k)}. \quad (3.17)$$

Figure 3.6 and Figure 3.7 show a pair of affine related shape images and their corresponding AIWSR; Figure 3.8 and Figure 3.9 show another pair of affine related shape images and their corresponding AIWSR. As we can see, the AIWSRs of the affine related shape images are very similar, while the AIWSRs of the unrelated shape images are not. So, the AIWSR can be used to discriminate between affine related shape images and unrelated ones.

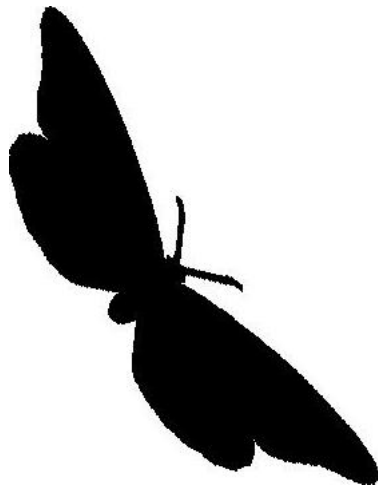


(a)shape image

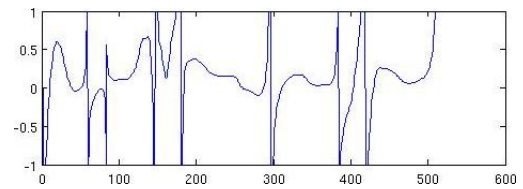


(b)wavelet based affine-invariant shape representation (Level of wavelet used, j=5,p=6.)

Figure 3.6: Shape image and its corresponding AIWSR

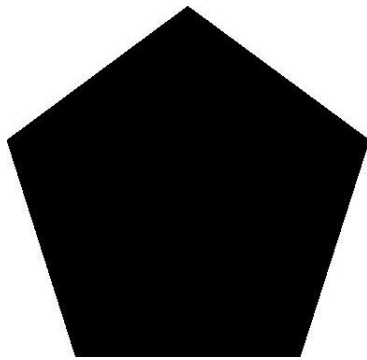


(a)shape image

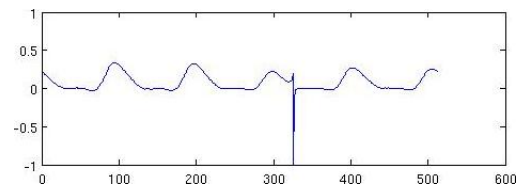


(b)wavelet based affine-invariant shape representation(Level of wavelet used, $j=5, p=6.$)

Figure 3.7: Shape image and its corresponding AIWSR

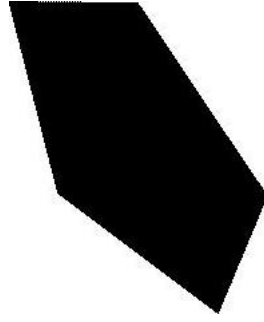


(a)shape image

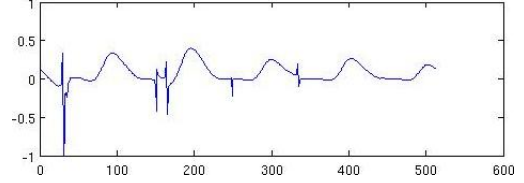


(b)wavelet based affine-invariant shape representation(Level of wavelet used, $j=5, p=6.$)

Figure 3.8: Shape image and its corresponding AIWSR



(a)shape image



(b)wavelet based affine-invariant shape representation(Level of wavelet used, j=5,p=6.)

Figure 3.9: Shape image and its corresponding AIWSR

3.6 Affine-invariant curvature scale space shape descriptor

The affine-invariant curvature scale space shape descriptor (AICSSSD) [62, 77] is extended from the curvature scale space shape descriptor (CSSSD) [78, 79]. In [77], Abbasi and Mokhtarian show that the CSSSD has only minor changes after a shape undergoes affine transformation. And the change is even smaller, when affine-invariant parametrization is also applied. Figures 3.10-3.13, show the shape images and their corresponding AICSSSDs.

After eliminating the translation effects by centering the shape to the origin, the affine transformation can be represented as:

$$\begin{bmatrix} x_a(u) \\ y_a(u) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x(u) \\ y(u) \end{bmatrix}, \quad (3.18)$$

where $x_a(u)$ and $y_a(u)$ represent the coordinates of the shape after affine transformation, and $x(u)$ and $y(u)$ represent the coordinates of the shape before the affine transformation. Since differential operations are linear operations, we have:

$$\dot{x}_a(u) = a_{11}\dot{x}(u) + a_{12}\dot{y}(u), \quad (3.19)$$

$$\dot{y}_a(u) = a_{21}\dot{x}(u) + a_{22}\dot{y}(u), \quad (3.20)$$

$$\ddot{x}_a(u) = a_{11}\ddot{x}(u) + a_{12}\ddot{y}(u), \quad (3.21)$$

and

$$\ddot{y}_a(u) = a_{21}\ddot{x}(u) + a_{22}\ddot{y}(u). \quad (3.22)$$

Since the curvature of a contour is calculated as:

$$\kappa(u) = \frac{\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)}{\left(\dot{x}^2(u) + \dot{y}^2(u)\right)^{\frac{3}{2}}}, \quad (3.23)$$

the curvature of the contour of an affine transformed shape is

$$\kappa_a(u) = \frac{\dot{x}_a(u)\ddot{y}_a(u) - \ddot{x}_a(u)\dot{y}_a(u)}{\left(\dot{x}_a^2(u) + \dot{y}_a^2(u)\right)^{\frac{3}{2}}}. \quad (3.24)$$

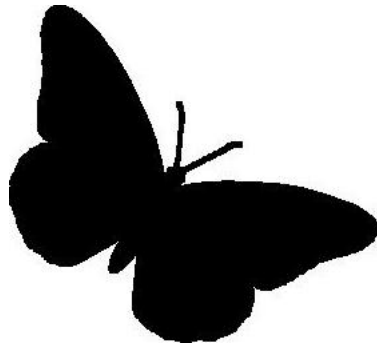
Combining Equation3.19-Equation3.22 and Equation3.24, we have

$$\kappa_a(u) = \frac{(a_{11}a_{22} - a_{12}a_{21})(\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u))}{\left((a_{11}\dot{x}(u) + a_{12}\dot{y}(u))^2 + (a_{21}\dot{x}(u) + a_{22}\dot{y}(u))^2\right)^{\frac{3}{2}}}. \quad (3.25)$$

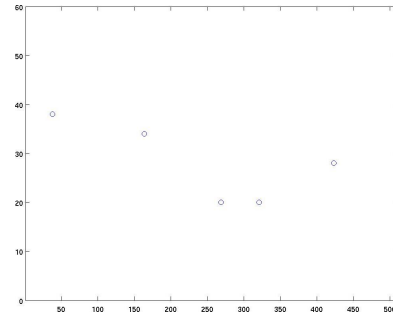
As the zero crossings of $\kappa(u)$ in Equation3.23 and $\kappa_a(u)$ in Equation 3.25 are both determined by $\dot{x}(u)\ddot{y}(u) - \ddot{x}(u)\dot{y}(u)$ in the numerators, the locations of the zero crossings on the curvatures of both the original and the affine transformed shapes are the same.

Figure 3.10 and Figure 3.11 show a pair of affine related shape images and their corresponding AICSSSDs. Figure 3.12 and Figure 3.13 show another pair of affine related shape images and their corresponding AICSSSDs. The AICSSSDs of the two affine related butterfly shapes match very well. The AICSSSDs of the two affine related pentagon shapes

match less well. The AICSSSD in Figure 3.12(b) has less components than the one in Figure 3.13. The reason is that these components with small values have been discarded. We can recall from Subsection 2.2.5.1, that those components less than one sixth of the largest components of the same CSS image are excluded in the shape representation. Overall, the affine-related AICSSSDs are similar and the affine-unrelated AICSSSDs are not similar. Therefore, the AICSSSDs can be used to discriminate between affine related shape images and unrelated ones.

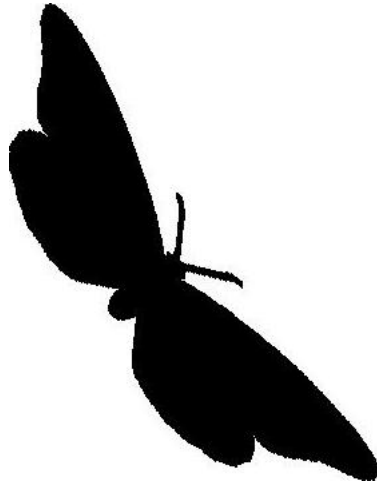


(a)shape image

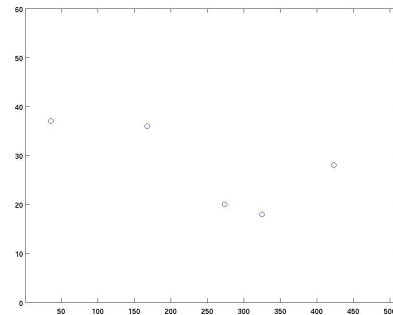


(b)its corresponding AICSSSD

Figure 3.10: Shape image and its corresponding AICSSSD

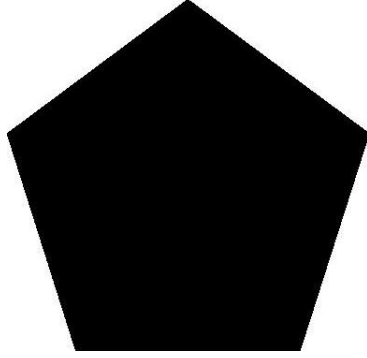


(a)shape image

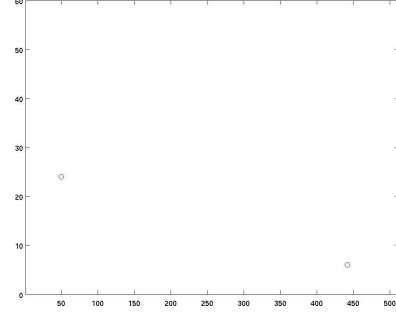


(b)its corresponding AICSSSD

Figure 3.11: Shape image and its corresponding AICSSSD

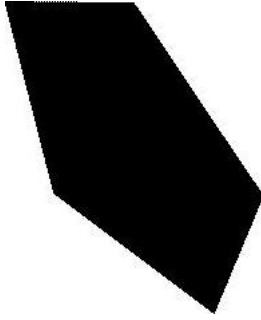


(a)shape image

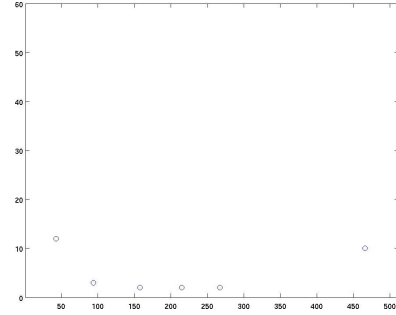


(b)its corresponding AICSSSD

Figure 3.12: Shape image and its corresponding AICSSSD



(a)shape image



(b)its corresponding AICSSSD

Figure 3.13: Shape image and its corresponding AICSSSD

3.7 Affine moment invariants by Taubin and Cooper

The affine-invariant representations we have discussed so far, are all contour-based and are limited to simple shapes. But many times, we would prefer to have region-based ones, since they are not limited to simple shapes.

The affine moment invariants, proposed by Taubin and Cooper (AMI-TC) [69], are region-based affine-invariant shape representation. They are eigenvalues of moment matrices. Altogether, there are eight proposed AMI-TCs, which consist of two eigenvalues of the symmetric 2×2 moment matrix $M'_{[1,2]}M'_{[2,1]}$, three eigenvalues of the 3×3 moment matrix

$M'_{[2,2]}$, two eigenvalues of the 2×2 moment matrix $M'_{[1,2]}M'_{[2,2]}M'_{[2,1]}$, and the value of the 1×1 moment matrix $M'_{[0,2]}M'_{[2,2]}M'_{[2,0]}$. These matrices of centered moments are computed not with respect to the original coordinate system, but with respect to the coordinate system defined by $\mathbf{x}' = L\mathbf{x}$, where L is a 2×2 lower triangular matrix of the results of the Cholesky decomposition [80] of $M_{[1,1]}$, and $M_{[1,1]}$ is the 2×2 matrix of centered moments with respect to the original coordinate system.

The steps of calculating those AMI-TCs could be summarized as following:

1. Calculate $M_{[1,1]} = \begin{bmatrix} \mu_{2,0} & \mu_{1,1} \\ \mu_{1,1} & \mu_{0,2} \end{bmatrix}$, where μ_{pq} are the centered moments defined in Equation 3.34;
2. Calculate the 2×2 lower triangular matrix L of the Cholesky decomposition of $M_{[1,1]}$;
3. Calculate the coordinates of pixels of the shape in the new coordinate system, using $\mathbf{x}' = L\mathbf{x}$, where $\mathbf{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$ are the coordinates of pixels of shape in the new and the original coordinate systems;
4. Calculate μ'_{pq} , the centered moments in the new coordinate system, using \mathbf{x}' ;
5. Calculate $M'_{[1,2]}M'_{[2,1]}$, $M'_{[1,2]}M'_{[2,2]}M'_{[2,1]}$, and $M'_{[0,2]}M'_{[2,2]}M'_{[2,0]}$, the matrices of the the centered moments in the new coordinate system, where

$$M'_{[1,2]} = \begin{bmatrix} \frac{1}{\sqrt{2}}\mu'_{3,0} & \mu'_{2,1} & \frac{1}{\sqrt{2}}\mu'_{1,2} \\ \frac{1}{\sqrt{2}}\mu'_{2,1} & \mu'_{1,2} & \frac{1}{\sqrt{2}}\mu'_{0,3} \end{bmatrix}, \quad (3.26)$$

$$M'_{[2,1]} = M'_{[1,2]}, \quad (3.27)$$

$$M'_{[2,2]} = \begin{bmatrix} \frac{1}{2}\mu'_{4,0} & \frac{1}{\sqrt{2}}\mu'_{3,1} & \frac{1}{2}\mu'_{2,2} \\ \frac{1}{\sqrt{2}}\mu'_{3,1} & \mu'_{2,2} & \frac{1}{\sqrt{2}}\mu'_{1,3} \\ \frac{1}{2}\mu'_{2,2} & \frac{1}{\sqrt{2}}\mu'_{1,3} & \frac{1}{2}\mu'_{0,4} \end{bmatrix}, \quad (3.28)$$

and

$$M'_{[0,2]}M'_{[2,2]}M'_{[2,0]} = \frac{1}{4}\mu'_{4,0} + \frac{1}{4}\mu'_{0,4}; \quad (3.29)$$

6. Calculate the eigenvalues of $M'_{[1,2]}M'_{[2,1]}$, $M'_{[1,2]}M'_{[2,2]}M'_{[2,1]}$, and $M'_{[0,2]}M'_{[2,2]}M'_{[2,0]}$ as the AMI-TCs.

In [69], the retrieval experiment using the AMI-TC is limited to a seven image database, which is rather too small to support its effectiveness. This shape representation doesn't work well under large testing databases, as we will see in Section 3.9.

3.8 Affine moment invariants by Flusser and Suk

In [66, 67, 68], Flusser and Suk proposed four affine moments invariants (AMI-FS), which are also invariant to affine transformations. They are:

$$I_1 = \frac{1}{\mu_{00}^4}(\mu_{20}\mu_{02} - \mu_{11}^2), \quad (3.30)$$

$$I_2 = \frac{1}{\mu_{00}^{10}}(\mu_{30}^2\mu_{03}^2 - 6\mu_{30}\mu_{21}\mu_{12}\mu_{03} + 4\mu_{30}\mu_{12}^3 + 4\mu_{03}\mu_{21}^3 - 3\mu_{21}^2\mu_{12}^2), \quad (3.31)$$

$$I_3 = \frac{1}{\mu_{00}^7}(\mu_{20}(\mu_{21}\mu_{03} - \mu_{12}^2) - \mu_{11}(\mu_{30}\mu_{03} - \mu_{21}\mu_{12}) + \mu_{02}(\mu_{30}\mu_{12} - \mu_{21}^2)), \quad (3.32)$$

and

$$\begin{aligned} I_4 = & \frac{1}{\mu_{00}^{11}}(\mu_{20}^3\mu_{03}^2 - 6\mu_{20}^2\mu_{11}\mu_{12}\mu_{03} - 6\mu_{20}^2\mu_{02}\mu_{21}\mu_{03} \\ & + 9\mu_{20}^2\mu_{02}\mu_{12}^2 + 12\mu_{20}\mu_{11}^2\mu_{21}\mu_{03} \\ & + 6\mu_{20}\mu_{11}\mu_{02}\mu_{30}\mu_{03} - 18\mu_{20}\mu_{11}\mu_{02}\mu_{21}\mu_{12} \\ & - 8\mu_{11}^3\mu_{30}\mu_{03} - 6\mu_{20}\mu_{02}^2\mu_{30}\mu_{12} \\ & + 9\mu_{20}\mu_{02}^2\mu_{21}^2 + 12\mu_{11}^2\mu_{02}\mu_{30}\mu_{12} \\ & - 6\mu_{11}\mu_{02}^2\mu_{30}\mu_{21} + \mu_{02}^3\mu_{30}^2), \end{aligned} \quad (3.33)$$

where

$$\mu_{pq} = \iint_G (x - x_c)^p (y - y_c)^q f(x, y) dx dy, \quad (3.34)$$

and (x_c, y_c) is the centroid of the shape.

The AMI-FS works better than the AMI-TC, which we will see from the experimental results in Section 3.9.

3.9 Experimental results

In the above sections, many previously known affine-invariant techniques have been described. In this section, we test and compare them in retrieval experiments.

3.9.1 Test database

Because contour-based affine-invariant shape representations can be applied only to simple shapes, retrieval experiments were only done on simple shape databases. On the other hand, region-based shape representations can be applied to both simple and complex shapes, so they were tested on both simple and complex shape databases. As current image segmentation techniques are not perfect, segmented shapes are often corrupted with boundary noise. In order to test the performance of the shape representations under noisy conditions, retrieval experiments have also been done on simple shape databases with noise added to the shape boundaries. The test shape database is given in details in the following:

- Complex shape database: Each of the 50 complex shape silhouette images, shown in Figure 3.15, were affine transformed using 80 different transformation matrices, creating a $50 \times 80 = 4000$ image complex shape test database. The parameters of the transformation in the experiments were: $\theta \in [0, \pi/6, 3\pi/5, 6\pi/7]$, $\alpha \in [0, 0.25, 0.5, 0.75, 1]$, $(s_x = s_y = 1)$, $(s_x = 0.7, s_y = -0.7)$, $(s_x = s_y = 2)$, $(s_x = 1, s_y = 2)$, where the relationship between the transformation matrix A and those parameters are:

$$A = \begin{bmatrix} s_x & \\ & s_y \end{bmatrix} \begin{bmatrix} 1 & \alpha \\ & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}.$$

- Simple shape databases (noise free): 70 different simple shape silhouette images of different classes, were selected from the MPEG-7 CE Shape-1 Part-B data set [81] (Figure 3.14). Each of them are affine transformed using the same 80 transformation matrices used in creating the complex shape database. Altogether, there are $70 \times 80 = 5600$ shape images in the simple shape database.
- Simple shape databases with boundary noise: Because shape segmentation techniques are often not perfect, the boundary of the shape is often corrupted by white Gaussian noise. To see the effect of boundary noise on the performance of the shape representations, simple shape databases with boundary noise were also used in the experiments. Based on the noise free simple shape database, zero-mean Gaussian noise with $SNR_{dB} = 30\text{dB}$, 26.9897dB , 26.0206dB , 24.7712dB , 23.0103dB , and 20dB , were added to the shape boundaries to create additional simple shape databases with boundary noise. Each of the database includes 5600 shape images. Some sample shapes are shown in Figure 3.16. Zero-mean Gaussian noise is statistical noise that has a probability density function (pdf) of the zero-mean Gaussian distribution, i.e., $pdf(x) = \frac{1}{\sqrt{2\pi}\sigma_{noise}} e^{-\frac{x^2}{2\sigma_{noise}^2}}$, where σ_{noise} is the standard deviation of the noise. The signal-to-noise ratio (SNR) is defined as the power ratio between the shape centroid distance and the shape boundary noise: $SNR = \frac{P_{cd}}{P_{noise}} = \frac{(A_{cd})^2}{(\sigma_{noise})^2}$, where P_{cd} and P_{noise} are the average power of the centroid distance function and the noise, respectively, A_{cd} is the root mean square amplitude of the centroid distance function. In decibels, the SNR_{dB} is defined as: $SNR_{dB} = 10\log_{10} \frac{P_{cd}}{P_{noise}} = 20\log_{10} \frac{A_{cd}}{\sigma_{noise}}$.

In [69], the retrieval experiment is limited to a seven image database. In [60] and [65], the testing databases include only twenty shape images. In [67], the size of the testing database is twenty-five. In [61], the test database consists of 1400 shape images. In [62], the test database consists of 5000 shape images. It's always preferred to use a large database for retrieval experiments, since shape representations that work well under small testing databases, may not work well under large ones.

The testing databases used in this work are far larger than most of those used in the experiments of other works. In addition, different amounts of affine distortion are included in

the testing shape images, and different levels of noise have been added to the shape boundary, to test the performance of the shape representations under noisy conditions. Thus, the experimental sets are more complete and rigid than those in other works.

3.9.2 Distance measure

In comparing two shape features, different distance measures will be used, depending on the kind of shape representation used.

- For the AIWSR, where the shape representation is a discrete function or a long discrete sequence of values, the normalized cross-correlation coefficient [61] are used in the experiments to measure the similarities between two shape representations. The normalized cross-correlation coefficient $R_{ab}(l)$ is defined as [61]:

$$R_{ab}(l) = \frac{\sum_l \sum_k a_k b_{k-l}}{\sqrt{\sum_k a_k^2 \sum_k b_k^2}}, \quad (3.35)$$

where a_k and b_k are two sequences under comparison, k is the index of the elements in the sequence, l is the step size of a circular shift. Since the cross-correlation coefficient is not translation invariant, one of the sequences, a_k or b_k , is shifted circularly step by step, the value of the correlation under each shift is calculated, and the maximum is selected as the similarity between two shape representations. Such a procedure is used to solve the problem caused by the variation of the shape boundary starting point.

- For those shape descriptors, the AIFSD, the AMI-TC, and the AMI-FS, where the shape representations are vectors, each of them will be tested using two different distance measures. The two distance measures are the Euclidean distance and the sum of absolute distance.

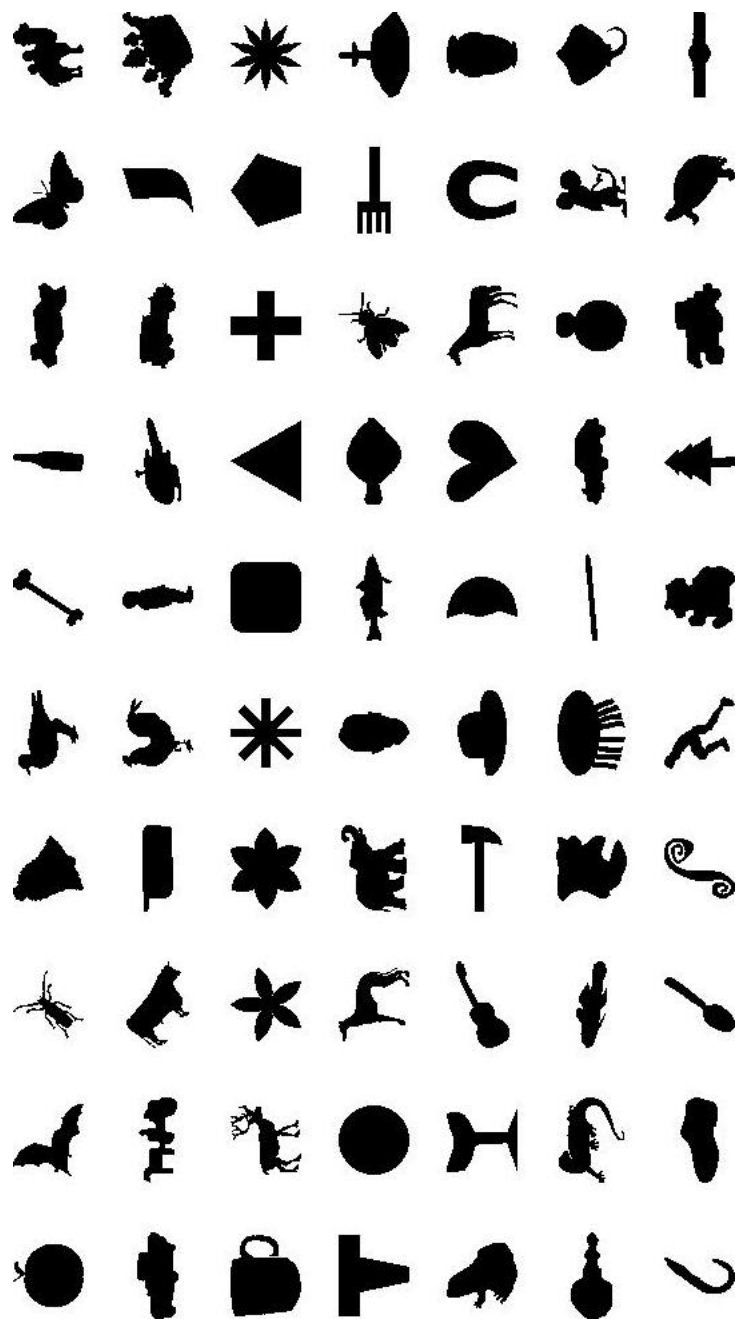


Figure 3.14: 70 Benchmark shape images from MPEG-7 Shape B

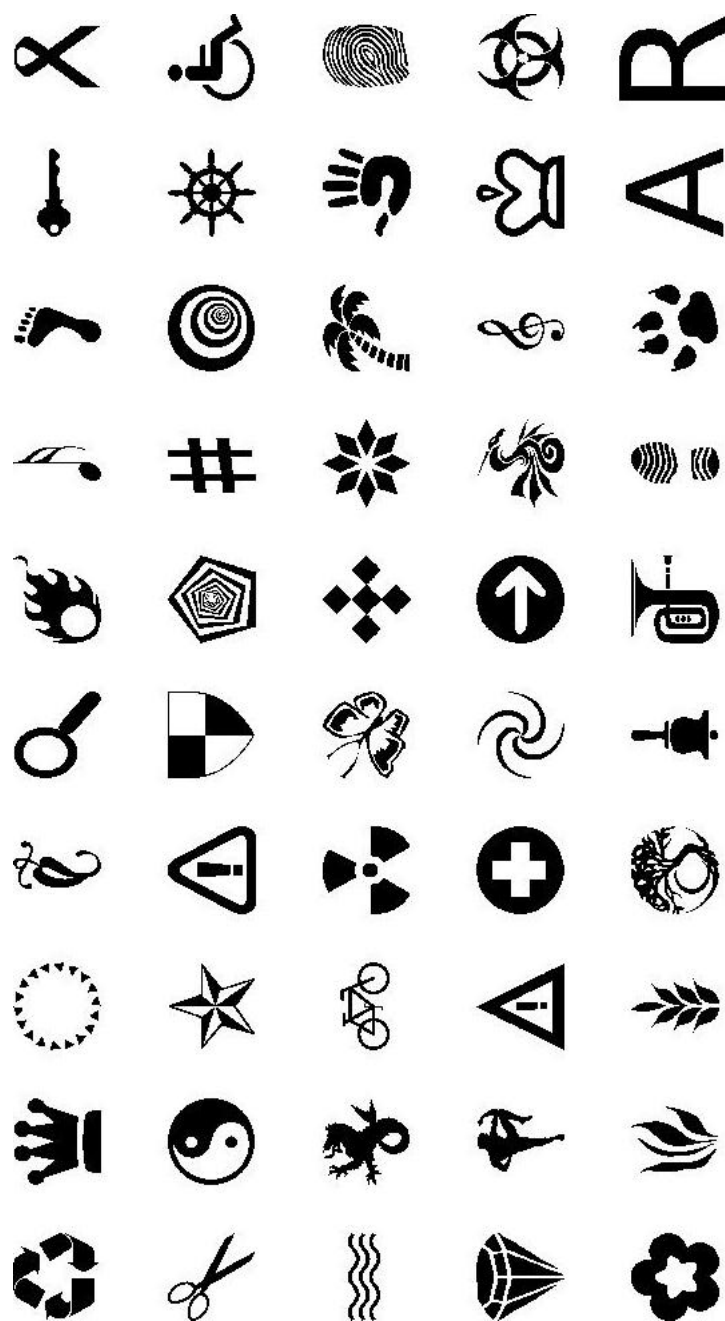


Figure 3.15: 50 Benchmark shape images

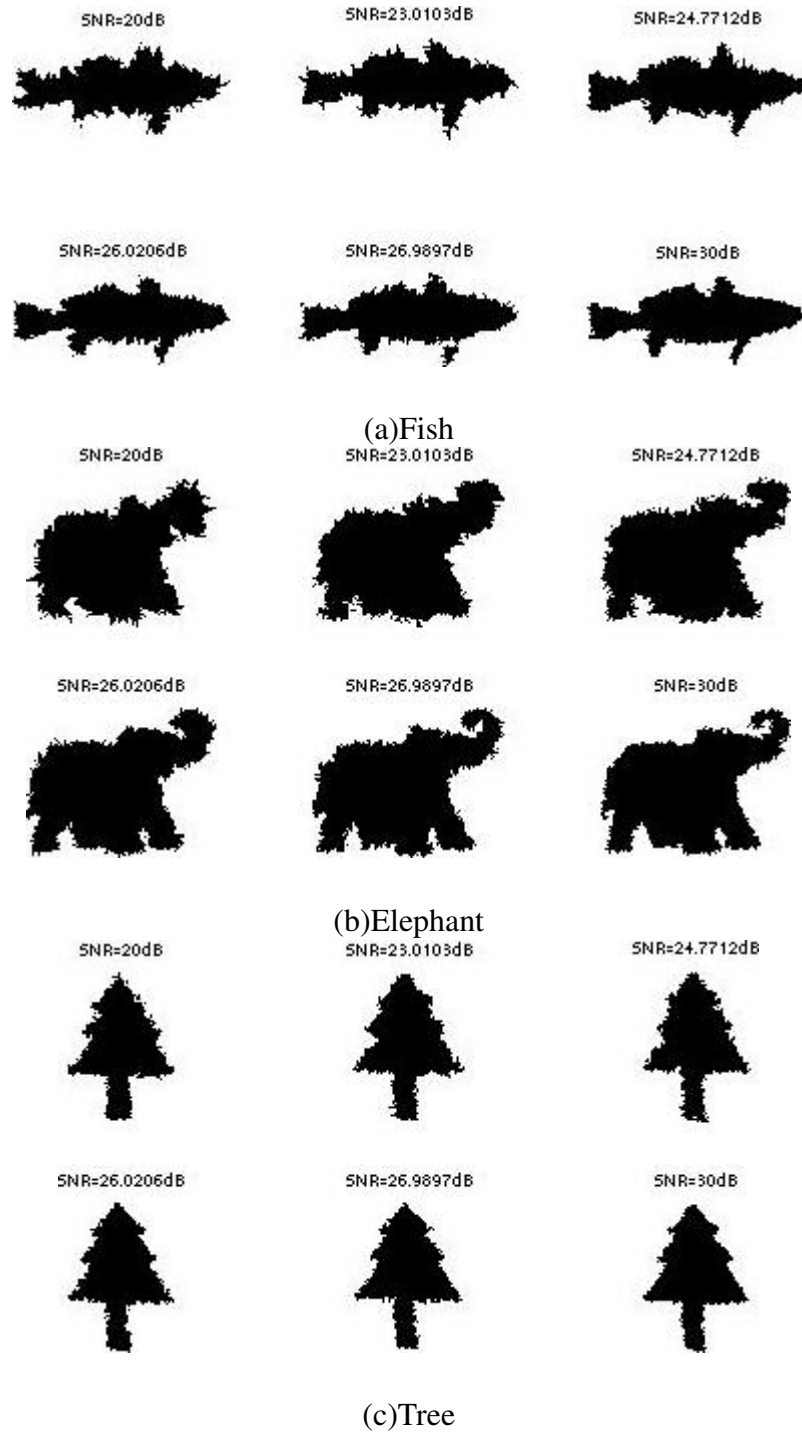


Figure 3.16: Sample shape images with zero-mean Gaussian noise added to the shape boundaries

- The Euclidean distance between two vectors $\mathbf{p} = [p_1, p_2, \dots, p_N]$ and $\mathbf{q} = [q_1, q_2, \dots, q_N]$ is defined as:

$$\begin{aligned} D_{Euclidean}(\mathbf{p}, \mathbf{q}) &= \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_N - q_N)^2} \\ &= \sqrt{\sum_{i=1}^N (p_i - q_i)^2} \end{aligned} \quad (3.36)$$

- The sum of absolute differences (SAD) between two vectors $\mathbf{p} = [p_1, p_2, \dots, p_N]$ and $\mathbf{q} = [q_1, q_2, \dots, q_N]$ is defined as:

$$\begin{aligned} D_{SAD}(\mathbf{p}, \mathbf{q}) &= |p_1 - q_1| + |p_2 - q_2| + \dots + |p_N - q_N| \\ &= \sum_{i=1}^N |p_i - q_i| \end{aligned} \quad (3.37)$$

- For the AICSSSD, the distance measure is the same as the CSSSD distance measure, which was described in Subsection 2.2.5.2.

3.9.3 Retrieval accuracy

The average precision-recall graphs are used as the measure of retrieval accuracy to see the effectiveness of the affine-invariant shape representations. Precision is defined as:

$$Precision = \frac{\text{Number of Retrieved Relevant Images}}{\text{Number of Retrieved Images}};$$

Recall is defined as:

$$Recall = \frac{\text{Number of Retrieved Relevant Images}}{\text{Number of Relevant Images}}.$$

For example, as shown in Figure 3.17, when the number of the relevant images is R , the number of retrieved images is K , and the number of the retrieved relevant images is G , the precision rate is:

$$Precision = \frac{G}{K},$$

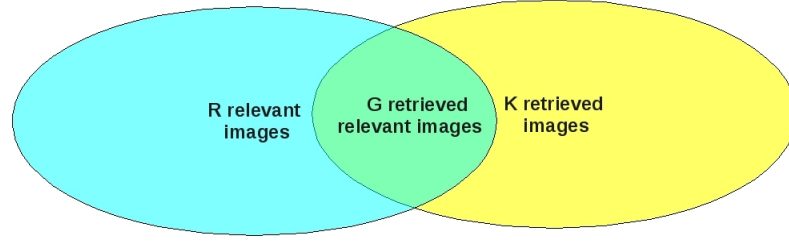


Figure 3.17: Calculation of precision rate and recall rate.

and the recall rate is:

$$Recall = \frac{G}{R}.$$

The precision rate at each level of the recall is calculated and recorded, for each query. The average precision rate using a shape descriptor is the average precision rate of all the query retrievals using that shape representation. A precision-recall curve that is closer to the top-right corner of the graph, indicates a better performance.

3.9.3.1 Comparison of the AMI-TC and the AMI-FS

The retrieval performance of the two affine-invariant region-based shape descriptors, the AMI-TC and the AMI-FS, are compared in Figures 3.18 - 3.25. The experiments have been done on both the complex shape database, the simple shape database without noise, and the simple shape database with different levels of noise. The Euclidean distance and the sum of absolute differences have been used for these experiments. From those average precision-recall graphs, we can see that the average precision-recall curves of the AMI-FS are much closer to the upper-right corners of the graphs than the AMI-TC, which indicates a much better performance of the AMI-FS than the AMI-TC. We can also see that the performance of the AMI-FS is better, regardless of the distance measure or database. From Figures 3.19 - 3.25, we can see that the retrieval performances of the shape descriptors degrade relative to the amount of noise added to the shape boundaries.

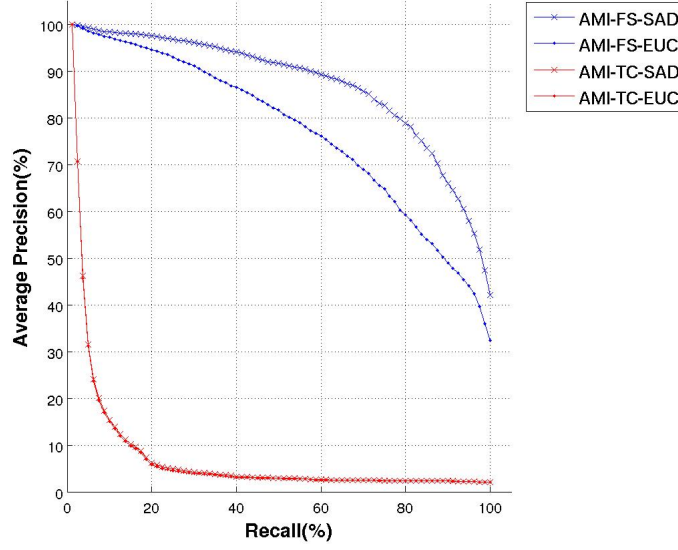


Figure 3.18: Average precision-recall graphs for 4000 objects using different affine-invariant region-based shape descriptors on the 4000 shape complex shape database.

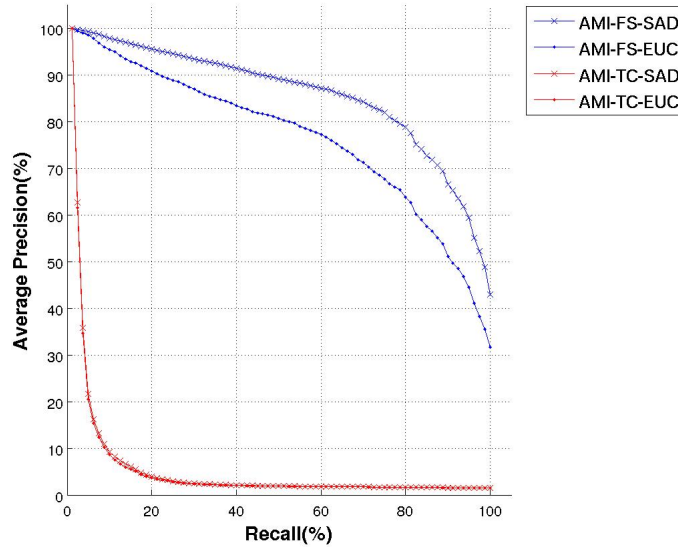


Figure 3.19: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database without noise.

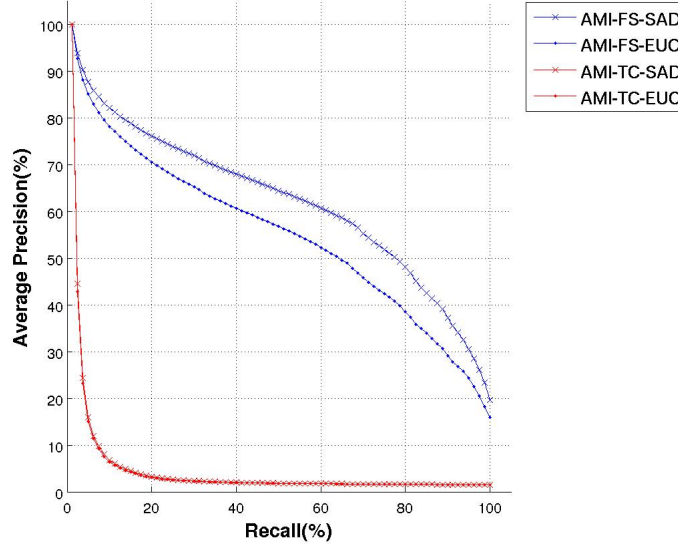


Figure 3.20: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=30\text{dB}$.

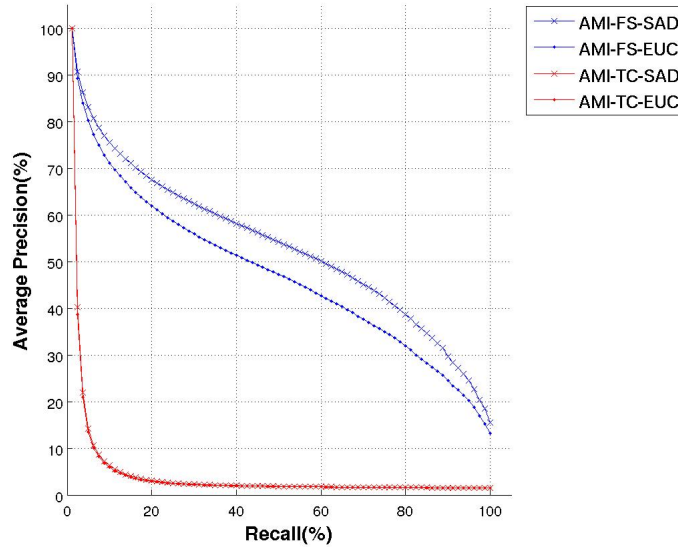


Figure 3.21: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$.

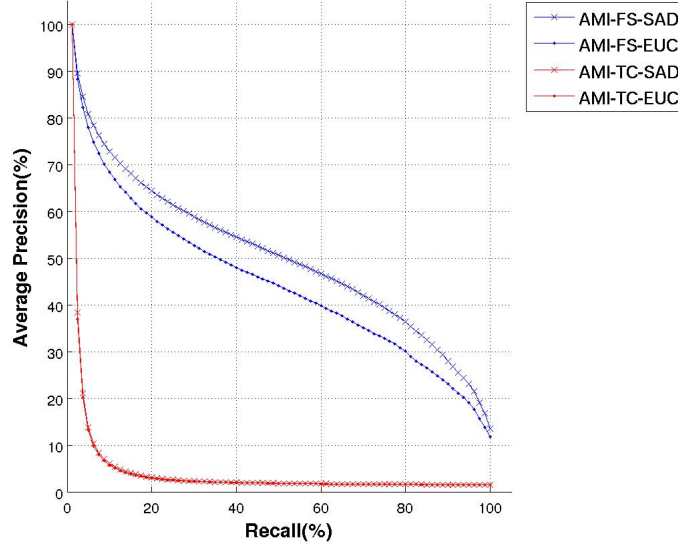


Figure 3.22: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.0206\text{dB}$.

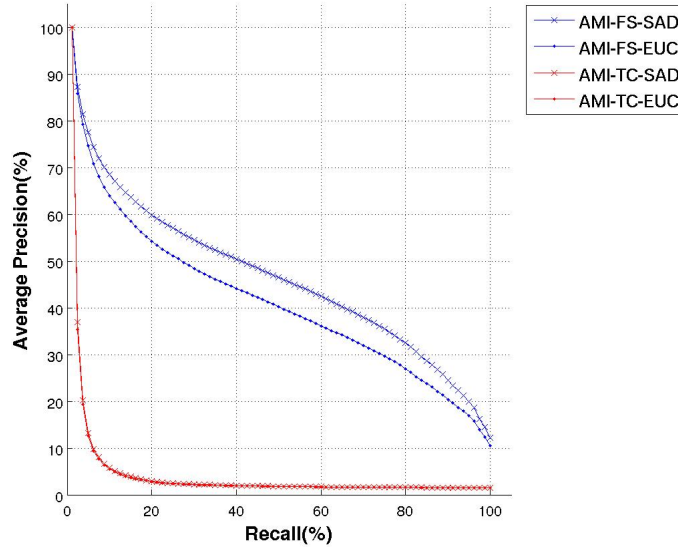


Figure 3.23: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=24.7712\text{dB}$.

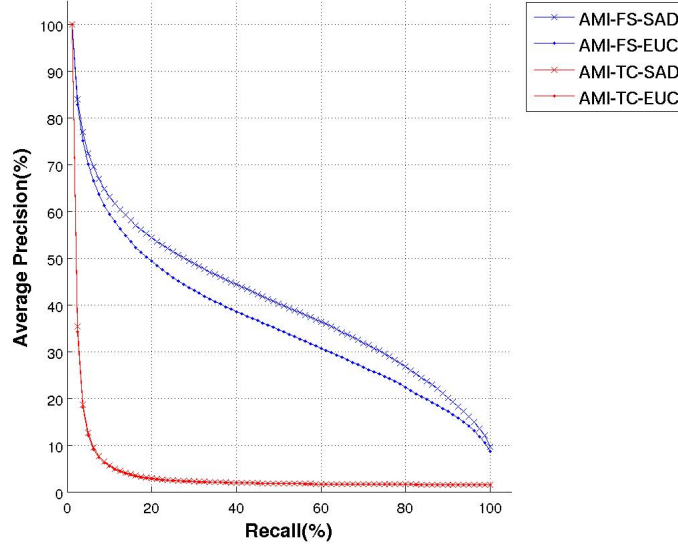


Figure 3.24: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=23.0103\text{dB}$.

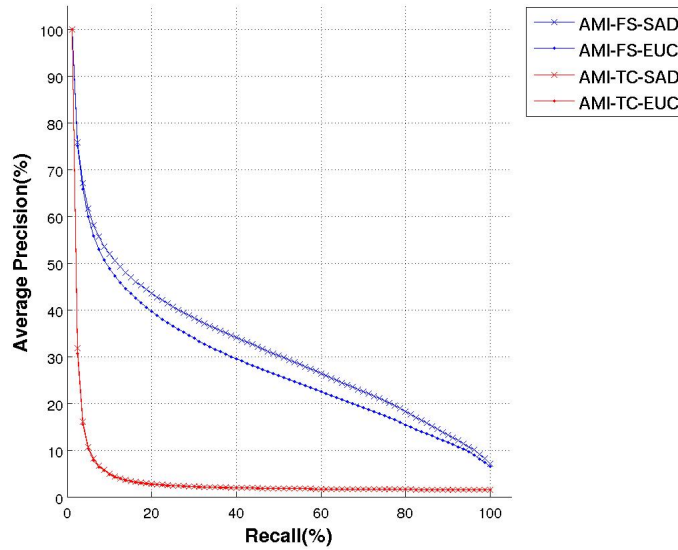


Figure 3.25: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$.

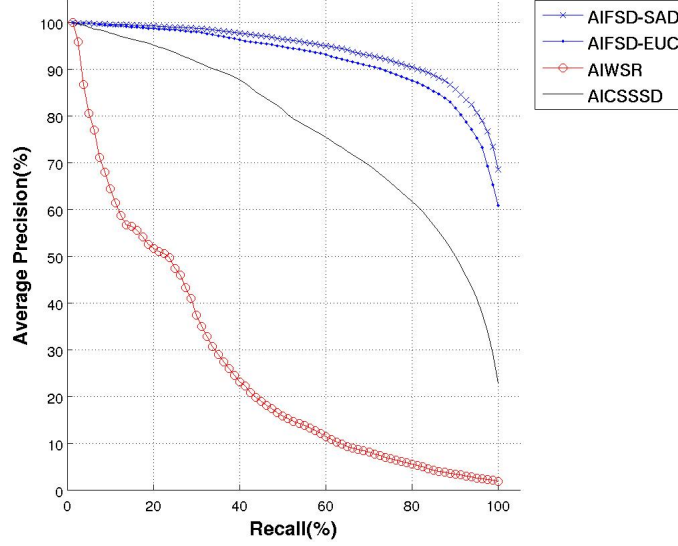


Figure 3.26: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database without noise.

3.9.3.2 Comparison of the AIFSD, the AICSSSD, and the AIWSR

The retrieval performances of the three affine-invariant contour-based shape representations, the AIFSD, the AICSSSD, and the AIWSR, are shown in Figures 3.26 - 3.32. Since those are contour-based shape representations, they are tested on simple shape databases with and without noise. The AIWSR performs the worst out of all of the shape representations in all of the tests. The performance of the AIFSD is better than that of AICSSSD, on the simple shape database without noise while the performance of the AIFSD is worse than the AICSSSD on the simple shape database with noise.

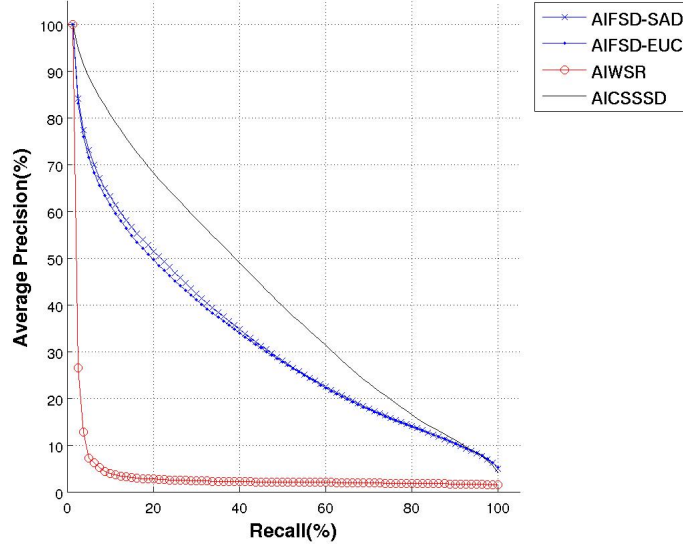


Figure 3.27: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=30\text{dB}$.

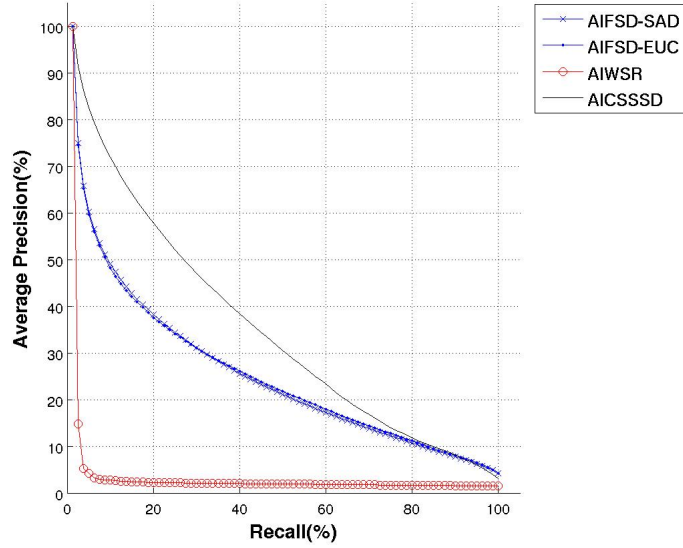


Figure 3.28: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$.

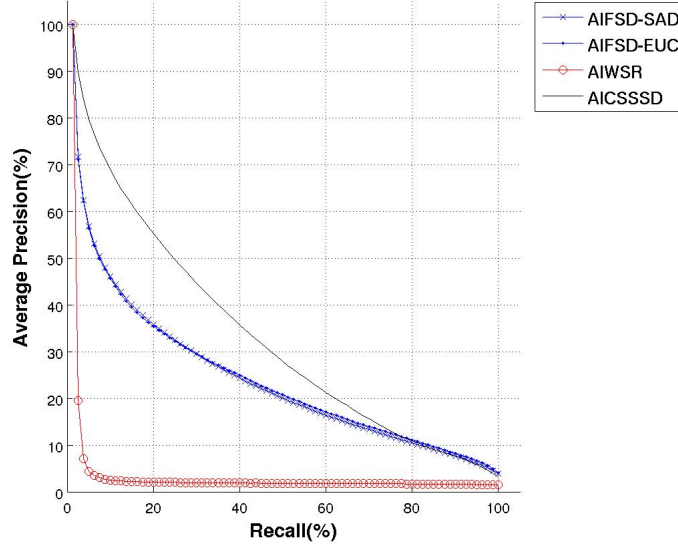


Figure 3.29: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.0206\text{dB}$.

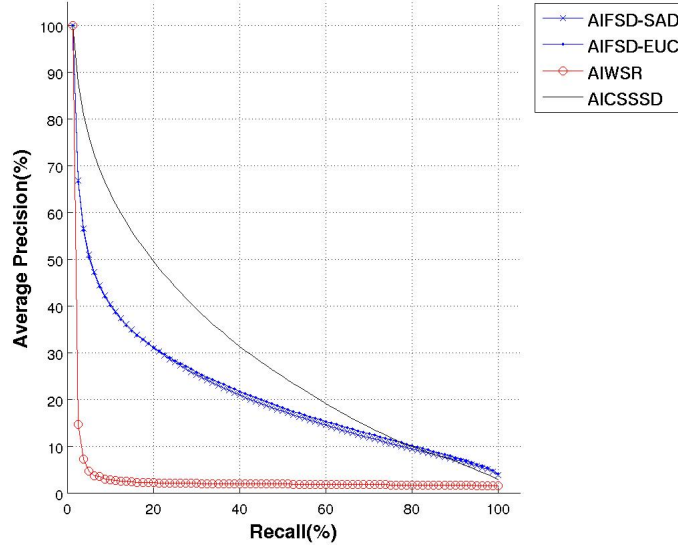


Figure 3.30: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=24.7712\text{dB}$.

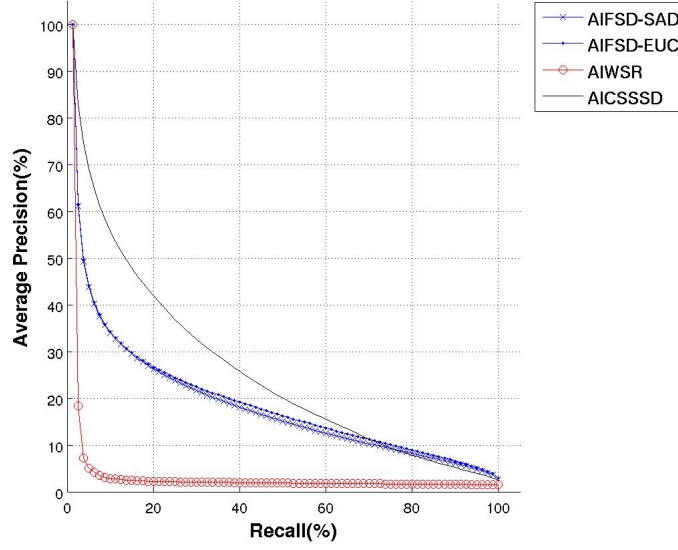


Figure 3.31: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=23.0103\text{dB}$.

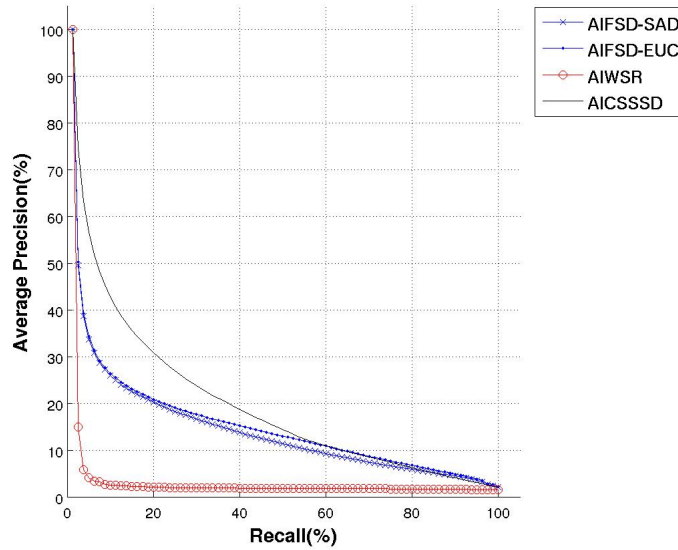


Figure 3.32: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$.

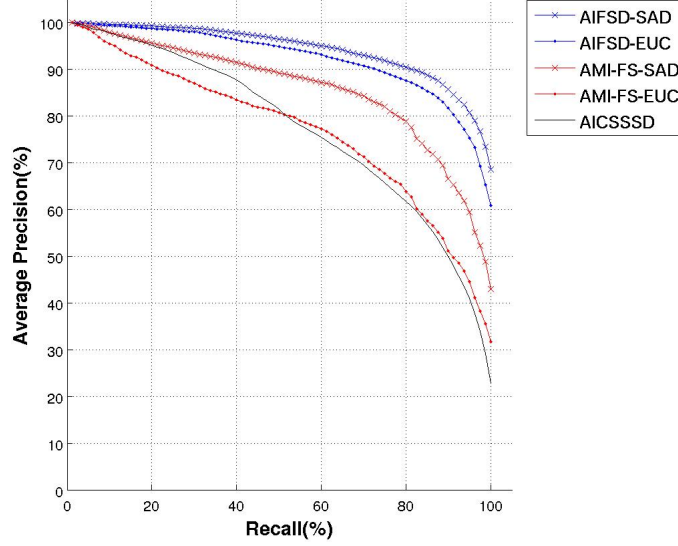


Figure 3.33: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database without noise.

3.9.3.3 Comparison of the AIFSD, the AICSSSD, and the AMI-FS

Here, we compare the performances of the region-based and the contour-based shape representations. The AMI-TC and the AIWSR are not included in the comparison, because of their poor retrieval performances. The performances of the AIFSD, the AICSSSD, and the AMI-FS, on simple databases, are compared in Figure 3.33 - Figure 3.39.

The region-based AMI-FS, generally performs better than the contour-based AIFSD and the contour-based AICSSSD, on shape databases with noise. The contour-based shape descriptors, which rely solely on the contour information of the shape, are more vulnerable to noise added to the shape boundaries. The region-based shape descriptor, which relies on the statistical information of all the shape pixels, is less vulnerable to the noise added only to the shape boundaries.

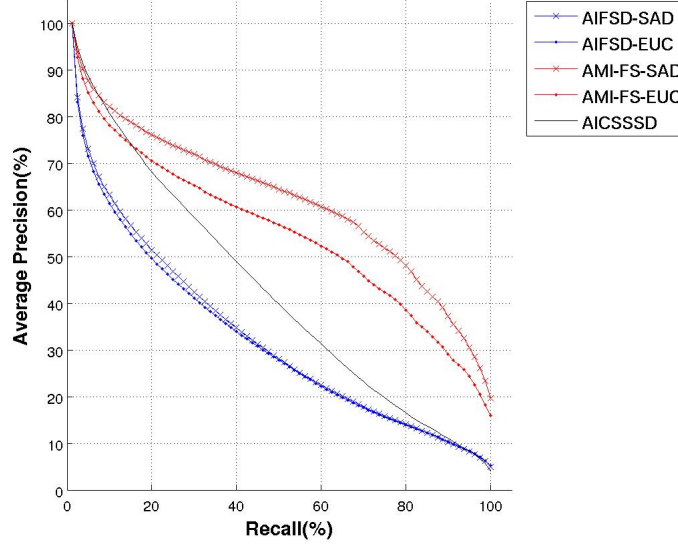


Figure 3.34: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=30\text{dB}$.

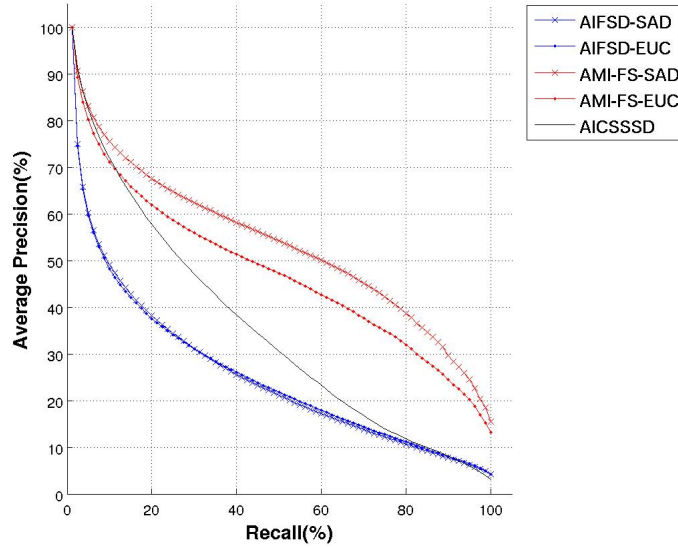


Figure 3.35: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$.

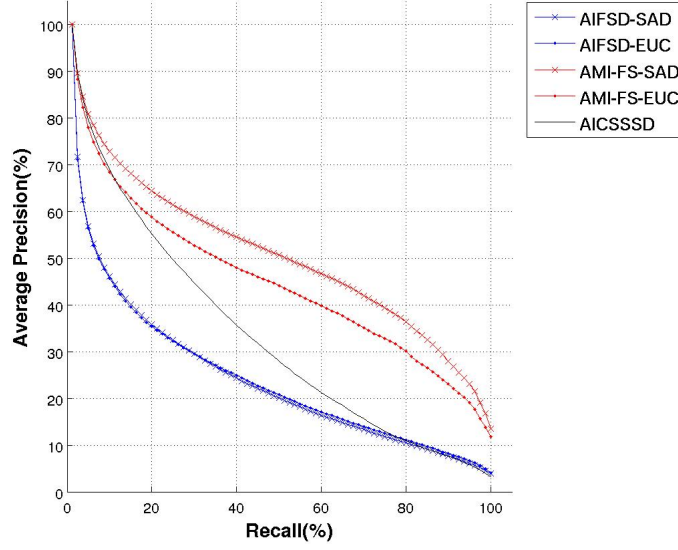


Figure 3.36: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.0206\text{dB}$.

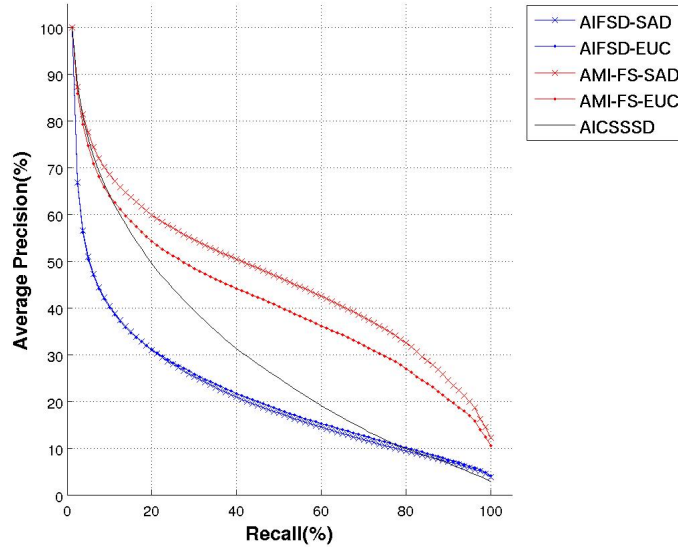


Figure 3.37: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=24.7712\text{dB}$.

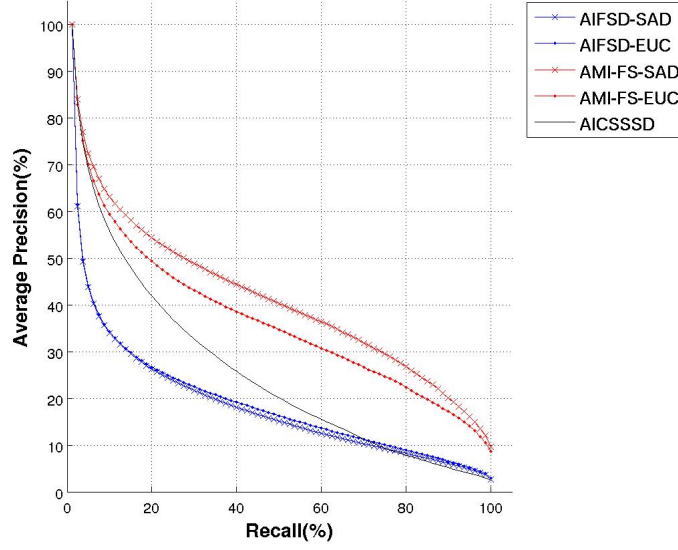


Figure 3.38: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=23.0103\text{dB}$.

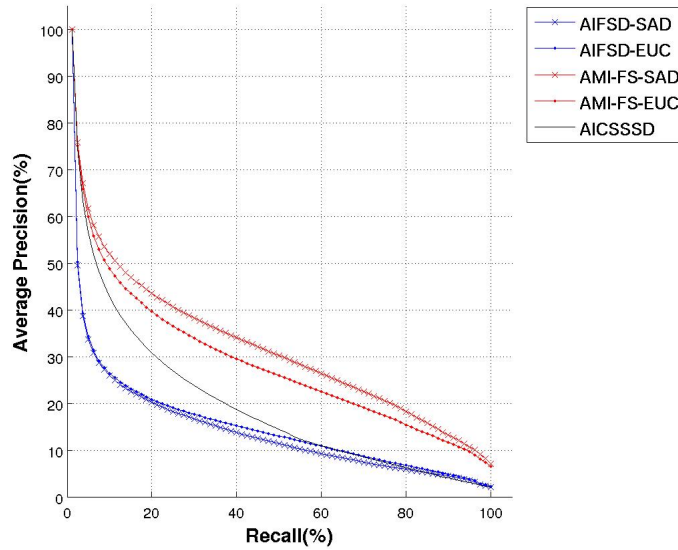


Figure 3.39: Average precision-recall graphs for 5600 objects using different affine-invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$.

3.9.4 Comparison of extraction time, distance calculation time, and compactness

Besides the accuracy of the retrieval performance of the shape representations, the computational time required to extract the shape representations (Table 3.1 and Figure 3.40), the computational time required to calculate the distance of two features (Table 3.2 and Figure 3.41), and the compactness of the shape representations (Table 3.3 and Figure 3.42) are all important criteria in evaluating them. The experiments were done on a Linux (Ubuntu 9.04) machine with Intel Core 2 Duo E6600 processor and 4G of RAM memory. The programming language is Matlab.

From Table 3.1 and Figure 3.40, we see that the AICSSSD requires much more computational time to extract than other shape representations. From Table 3.2 and Figure 3.41, we see that the AIWSR requires much more time to calculate feature distance than other shape representations. From Table 3.3 and Figure 3.42, we see that the AIWSR also has a much bigger feature size than other shape representations.

Shape representation	Extraction time(s)	Shape representation	Extraction time(s)	Shape representation	Extraction time(s)
AIFSD	0.0136	AICSSSD	13.9972	AMI-FS	0.1835
AIWSR	0.1377	AMI-TC	0.0521		

Table 3.1: Feature extraction time (seconds)

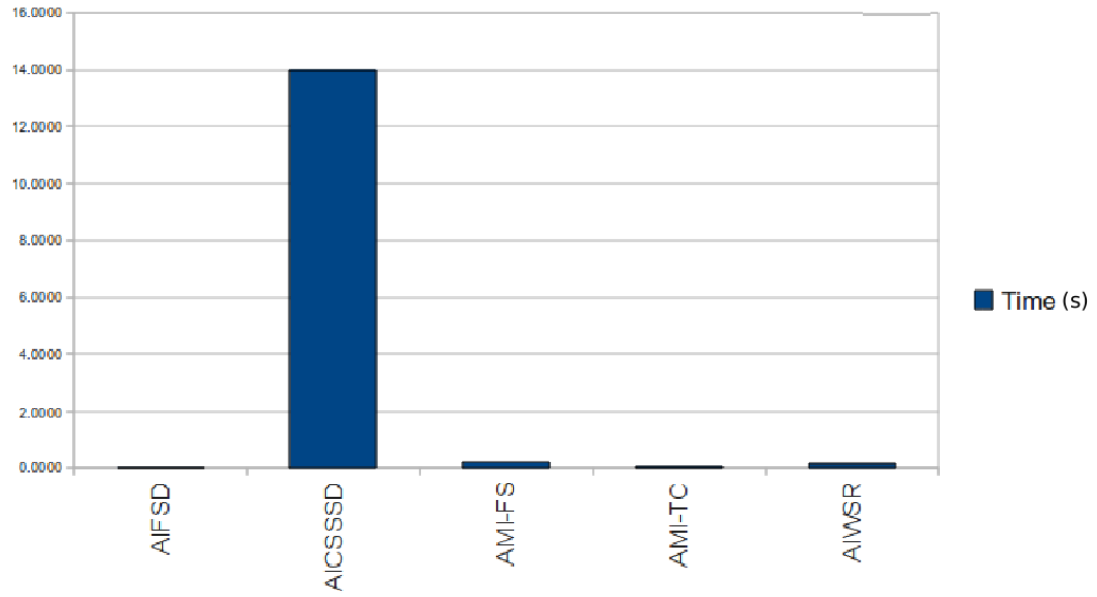


Figure 3.40: Comparison of feature extraction time (seconds)

Shape representation	Extraction time(ms)	Shape representation	Extraction time(ms)
AMI-TC-SAD	0.021	AMI-TC-EUC	0.006
AMI-FS-SAD	0.013	AMI-FS-EUC	0.005
AIFSD-SAD	0.009	AIFSD-EUC	0.006
AICSSD	18.999	AIWSR	120.478

Table 3.2: Feature distance calculation time (milliseconds)

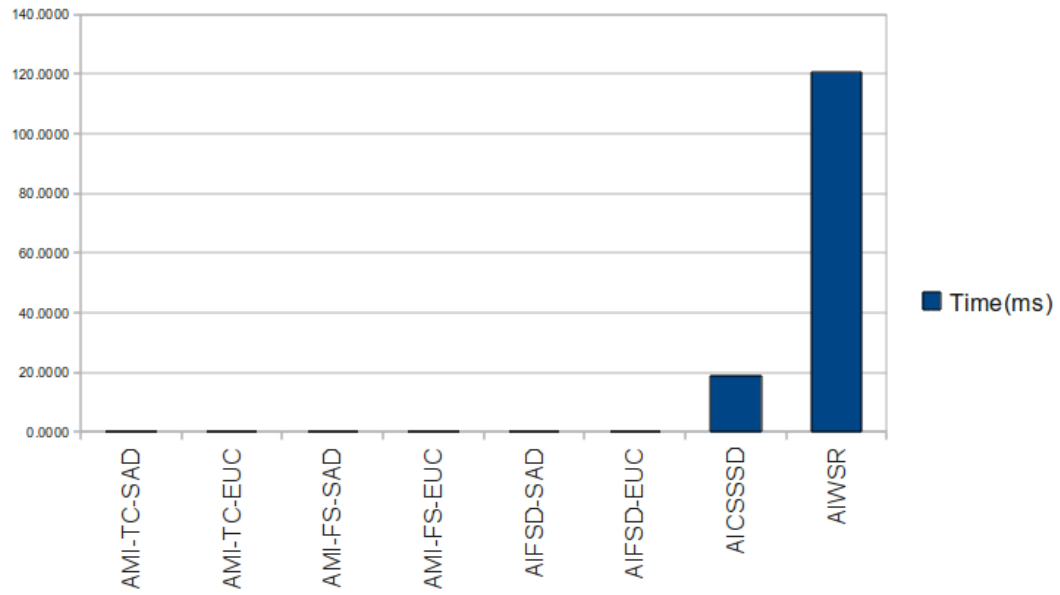


Figure 3.41: Comparison of feature distance calculation time (milliseconds)

Shape representation	Size	Shape representation	Size	Shape representation	Size
AIFSD	36	AICSSSD	(average) 5	AMI-FS	4
AIWSR	512	AMI-TC	8		

Table 3.3: Size of the shape representations

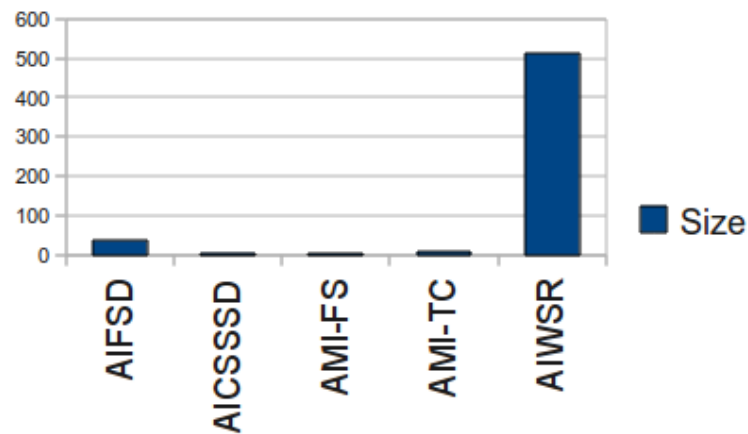


Figure 3.42: Comparison of the size of the shape representations

3.10 Summary

In this chapter, previously developed affine-invariant shape representations have been reviewed and compared in several different retrieval experiments. For affine-invariant contour-based shape representations, affine-invariant parameters, instead of the arc length parameter, are often used, as they are *linear* under an affine transformation. In the contour-based category, the AIFSD and the AICSSSD show better retrieval performance than the AIWSR, while in the region-based category, the AMI-FS performs better than the AMI-TC. The region-based shape descriptor, the AMI-FS, also show better retrieval performance than the two contour-based shape descriptors, the AIFSD and the AICSSSD, on simple shapes with boundary noise. Feature extraction time, feature distance calculation time, and feature compactness of the shape representations were also tested and compared. The AICSSSD requires the most of feature extraction time. The AIWSR, which is a shape signature, requires more computational time to calculate the feature distance and is less compact than those shape descriptors that are in a vector form.

Chapter 4

Independent Component Analysis

4.1 Introduction

In the previous chapter, we studied different affine-invariant shape representations and found that their performance is not satisfactory. In this work, several new affine-invariant shape techniques that have very good retrieval performance and are very robust under noisy conditions, have been developed. Since those new techniques are all related to the technique of independent component analysis (ICA), it will first be reviewed in this chapter.

ICA is a relatively new statistical technique and was first introduced in the early eighties by J. Herault, C. Jutten, and B. Ans [82]. The ICA model first came up in a neurophysiological setting, where it was used to extract the position and velocity signals, $s_1(t)$ and $s_2(t)$, of a moving joint from the two signals of muscle contraction, $x_1(t)$ and $x_2(t)$. Since then, ICA has been applied in different areas. One of the most famous application of ICA is to solve the Cocktail Party Problem [83, 84], where the original speakers' voices can be separated using the received stereo sounds in a noisy cocktail party environment. Similar applications include separating useful signals from undesired noise signals in electroencephalography (EEG) [85], in magnetoencephalography (MEG) [86], in electrocardiogram (ECG) [87][88], and in electrogastrogram (EGG)[89]. In image processing, ICA has been used to extract image bases for sparse code shrinkage based image denoising [90, 91, 92]. In financial applications, ICA has been used to find hidden factors in financial data [93],

and to forecast financial time series data [94]. In this work, a totally new usage of ICA for affine-invariant feature extraction is introduced.

In Section 4.2, we introduce the ICA model, the ambiguities of ICA, its relationship with whitening and the criteria used in ICA estimation. After that, in Section 4.3, examples using two statistically different types of data will be used to illustrate the theories discussed in Section 4.2. Section 4.4 summarizes this chapter.

4.2 ICA, its ambiguities, its relationship with whitening, and its estimation criteria

4.2.1 The ICA model

ICA is a statistical technique that extracts independent source signals from their observed linear mixtures. Using a vector-matrix notation, the ICA model could be defined using the following equation:

$$\mathbf{x}(t) = \mathbf{M} \cdot \mathbf{s}(t), \quad (4.1)$$

where $\mathbf{x}(t) = [x_1(t), \dots, x_N(t)]^T$ is one sample of the observed signals, $\mathbf{s}(t) = [s_1(t), \dots, s_M(t)]^T$ is one sample of the independent source signals, and \mathbf{M} is the mixing matrix. For simplicity, we can assume $N = M$ and the unknown mixing matrix \mathbf{M} is square. The task of ICA is to compute its inverse, say \mathbf{W} , and obtain the independent components by:

$$\mathbf{y}(t) = \mathbf{W} \cdot \mathbf{x}(t), \quad (4.2)$$

where $\mathbf{y}(t) = [y_1(t), \dots, y_N(t)]^T$ is one sample of the estimated independent signals. To simplify the notation, from now on, we ignore the time index t in the ICA model, so Equation 4.1 appears as:

$$\mathbf{x} = \mathbf{M} \cdot \mathbf{s}, \quad (4.3)$$

and Equation 4.2 appears as:

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{x}. \quad (4.4)$$

4.2.2 Ambiguities of ICA

ICA can find out the underlining independence components, but cannot determine their variances, signs, or order. This is known as the ambiguities of ICA [95].

We can see from Equation 4.3, as both \mathbf{M} and \mathbf{s} are unknown, multiplying any scalars β_i to the independent sources s_i and divide the corresponding columns \mathbf{m}_i in the mixing matrix \mathbf{M} by the same scalars β_i , will still keep the equation valid:

$$\mathbf{x} = \mathbf{M} \cdot \mathbf{s} = \sum_i \mathbf{m}_i \cdot s_i = \sum_i \left(\frac{\mathbf{m}_i}{\beta_i} \right) \cdot (s_i \beta_i). \quad (4.5)$$

So, the variances or the energies of the independent component sources s_i cannot be determined. In most ICA algorithms, by convention, the sources s_i are assumed to have unit variance: $E\{s_i^2\} = 1$ and the unknown mixing matrix \mathbf{M} is adapted to the assumption.

Similarly, there is the ambiguity of the sign: multiplying any independent sources s_i with -1 and multiplying the corresponding columns \mathbf{a}_i with -1 at the same time, will not affect the balance of Equation 4.3.

The ambiguity of the order, can be proved by modifying Equation 4.3 with a permutation matrix \mathbf{P} and its inverse \mathbf{P}^{-1} as:

$$\mathbf{x} = \mathbf{M} \cdot \mathbf{s} = \mathbf{M}\mathbf{P}^{-1}\mathbf{P}\mathbf{s} = (\mathbf{M}\mathbf{P}^{-1}) \cdot (\mathbf{P}\mathbf{s}). \quad (4.6)$$

Then, $\mathbf{M}\mathbf{P}^{-1}$ is the new unknown mixing matrix and $\mathbf{P}\mathbf{s}$ contains the new independent component sources s_j with different order.

Those ambiguities of ICA will appear again later, when we discuss how ICA will be used in extracting affine invariant shape descriptors.

4.2.3 Whitening as a preprocessing step in ICA

In most of the ICA algorithms, whitening is an important and useful preprocessing step in ICA. Whitening the observed data, which includes decorrelating it and normalizing it along

its principle component directions. will make the ICA estimation easier. To see that, we will first review the concepts of independence, uncorrelatedness and whiteness.

4.2.3.1 Independence

Two scalar random variables x and y are independent, if and only if their joint density $p_{xy}(x, y)$ can be factorized into their marginal density $p_x(x)$ and $p_y(y)$:

$$p_{xy}(x, y) = p_x(x) \cdot p_y(y). \quad (4.7)$$

4.2.3.2 Uncorrelatedness

Two scalar random variables x and y are uncorrelated, if their covariance is zero:

$$c_{xy} = E \{ (x - m_x)(y - m_y) \} = 0, \quad (4.8)$$

where m_x and m_y denote the mean of x and y . If different components of a random vector $\mathbf{z} = (z_1, z_2, \dots, z_n)^T$ are uncorrelated, then the covariance matrix of \mathbf{z} is diagonal:

$$\mathbf{C}_z = E \{ (\mathbf{z} - \mathbf{m}_z)(\mathbf{z} - \mathbf{m}_z)^T \} = \mathbf{D}, \quad (4.9)$$

where \mathbf{m}_z is the mean of the random vector \mathbf{z} . \mathbf{D} is a $n \times n$ diagonal matrix:

$$\mathbf{D} = \text{diag}(c_{11}, c_{22}, \dots, c_{nn}) = \text{diag}(\sigma_{z_1}^2, \sigma_{z_2}^2, \dots, \sigma_{z_n}^2), \quad (4.10)$$

where i th ($i = 1, 2, \dots, n$) diagonal elements of the matrix are the variances $\sigma_{z_i}^2$ of the components, z_i . A common way to decorrelate data is to perform principle component analysis (PCA) [96] on them. Uncorrelatedness is a weaker form of independence. Uncorrelated random variables are not necessarily independent, on the contrary, independent random variables must be uncorrelated.

4.2.3.3 Whiteness

Whiteness is a stronger condition than uncorrelatedness, but a weaker condition than independence. Random vector \mathbf{z} is white, when it has zero mean and unit covariance matrix:

$$\mathbf{m}_z = \mathbf{0}, \quad (4.11)$$

and

$$\mathbf{C}_z = \mathbf{I}, \quad (4.12)$$

where \mathbf{I} is the $n \times n$ identity matrix. Data whitening goes one further step than decorrelating, and can be done by normalizing the uncorrelated data along its principle component directions.

4.2.3.4 Whitening simplifies the ICA estimation

Suppose \mathbf{z} is the whitened random vector and \mathbf{V} is the whitening matrix, we have:

$$\mathbf{z} = \mathbf{V} \cdot \mathbf{x} = \mathbf{V} \cdot \mathbf{M} \cdot \mathbf{s} = \tilde{\mathbf{M}} \cdot \mathbf{s}, \quad (4.13)$$

where $\tilde{\mathbf{M}} = \mathbf{V} \cdot \mathbf{M}$ is the $n \times n$, new mixing matrix. Since \mathbf{z} is already whitened, and the components in \mathbf{s} is independent of each others and fixed to unit variance, we have:

$$E \{ \mathbf{z} \mathbf{z}^T \} = \tilde{\mathbf{M}} \cdot E \{ \mathbf{s} \mathbf{s}^T \} \cdot \tilde{\mathbf{M}}^T = \tilde{\mathbf{M}} \cdot \tilde{\mathbf{M}}^T = \mathbf{I}. \quad (4.14)$$

That means the new mixing matrix $\tilde{\mathbf{M}}$ and its inverse matrix $\tilde{\mathbf{W}}$ that needs to be estimated are orthogonal. An $n \times n$ orthogonal matrix has $n(n-1)/2$ degrees of freedom. On the another hand, the demixing matrix \mathbf{W} will need n^2 elements to describe. So, using the whitened vector \mathbf{z} , instead of using the observed vector \mathbf{x} , to estimate the independent components \mathbf{s} , will reduce the number of parameters that need to be estimated. For example, in a two dimensional case ($n=2$), four parameters describing \mathbf{W} need to be estimated, but only one parameter describing $\tilde{\mathbf{M}}$, or equivalently its inverse matrix $\tilde{\mathbf{W}}$, needs to be estimated. Since an orthogonal transformation is either a rigid rotation or a rotation followed by a flip [97], the difference between the whitened variables and the final estimated independent component is either only a rotation or a rotation followed by a flip.

As we have seen, whitening is an important preprocessing step in ICA, as it reduces the number of parameters that need to be estimated, from n^2 to $n(n-1)/2$ and greatly

simplifies ICA algorithms. The core part of ICA algorithms begins after the whitening of the data. In the following sections, principles used in estimating ICA will be discussed.

4.2.4 ICA by maximization of non-gaussianity

We know from the Central Limit Theorem that the sum of non-gaussian variables tends to be more gaussian-distributed than the original ones [98]. So, the problem of estimating the inverse matrix $\widetilde{\mathbf{W}}$ that makes y_i as independent of other values as possible can be turned into finding the new data projections where the data y_i are the most non-gaussian.

4.2.4.1 Measuring non-gaussianity by kurtosis

Since non-gaussianity is so important in estimation ICA, we need to have quantitative measures of non-gaussianity of a random variable. A classic measure of non-gaussianity is the fourth-order cumulant, or kurtosis. The kurtosis of a random variable y , denoted by $kurt(y)$, is defined as [36]:

$$kurt(y) = E \{y^4\} - 3 (E \{y^2\})^2. \quad (4.15)$$

When the variable y has been normalized, i.e. $E \{y^2\} = 1$, the kurtosis of y is simplified to:

$$kurt(y) = E \{y^4\} - 3. \quad (4.16)$$

The kurtosis of a gaussian random variable is zero, while that of a nongaussian random variable is either bigger or smaller than zero. Random variables with positive kurtosis values are called supergaussian, and those with negative kurtosis values are called subgaussian. Supergaussian random variables have typically a *spiky* distribution, while subgaussian random variables have typically a *flat* distribution. The Laplacian distribution and the uniform distribution are typical examples of supergaussian and subgaussian distributions, respectively.

Since the gaussian distribution has a kurtosis value of zero, estimating ICA by the maximization of non-gaussianity, is equivalent to estimating the inverse matrix $\widetilde{\mathbf{W}}$ that

maximizes the absolute value of the kurtosis of the estimated independent components [36].

4.2.4.2 Measuring non-gaussianity by negentropy

Although kurtosis is an important measure of non-gaussianity and is simple to calculate, it is not a robust measure. That is because kurtosis is very sensitive to outliers and the value of kurtosis can depend on only a few outliers. Negentropy, the normalized version of differential entropy, is a robust estimator of non-gaussianity. The differential entropy H of a random variable y is defined as [99]:

$$H(y) = - \int p_y(\eta) \log p_y(\eta) d\eta, \quad (4.17)$$

where $p_y(\eta)$ is the probability distribution of y . Differential entropy is a measure of the randomness of the variable. The less structured a distribution is, the larger its differential entropy is. Since the distribution of gaussian variable is the least structured, its differential entropy is the largest among the distributions of all the random variables that has the same variance. The negentropy J of multidimensional random vector \mathbf{y} is defined as:

$$J(\mathbf{y}) = H(\mathbf{y}_{\text{gauss}}) - H(\mathbf{y}), \quad (4.18)$$

where $\mathbf{y}_{\text{gauss}}$ is a gaussian random vector that has the same covariance matrix as that of random vector \mathbf{y} . Since the gaussian distribution has the largest differential entropy, negentropy is always non-negative and is equal to zero if, and only if, the random vector has a gaussian distribution. To look for the projection that generates the least gaussian distributed data is to look for the projection that generates the data with the highest negentropy value.

In [100], where the objective function utilized negentropy, the inverse matrix $\widetilde{\mathbf{W}}$, which is the solution to ICA, was found at the maxima of the objective function. The advantage of using negentropy over kurtosis is that it is more robust. The disadvantage is that the calculation of negentropy by definition is computationally complicated. However, some simpler approximations of negentropy have been developed in [101, 100].

In addition to its robustness, negentropy has an interesting and useful feature in that it is invariant to linear transforms. We will see this, when discussing its relationship with mutual information.

4.2.5 ICA by minimization of mutual information

An alternative to estimating ICA by maximization of non-gaussianity, is estimating ICA by minimization of mutual information. The mutual information I between scalar random variables $y_i (i = 1..N)$, is defined as [99]:

$$I(y_1, \dots, y_N) = \sum_{i=1}^N H(y_i) - H(\mathbf{y}), \quad (4.19)$$

Mutual information is a measure of dependence between random variables. It is always non-negative, and equals zero if and only if the random variables are statistically independent. Thus, mutual information could be used as a measure of dependence in ICA algorithm [102, 100]. Since the estimated independent components y_i should be independent of each other, they should also have the smallest mutual information. In [100], objective function through mutual information was also used, the inverse matrix $\widetilde{\mathbf{W}}$, which was the solution to ICA, was found at the minima of the objective function.

The relationship between mutual information and negentropy is [95]:

$$I(y_1, \dots, y_N) = J(\mathbf{y}) - \sum J(y_i) + \frac{1}{2} \log \frac{\prod c_{ii}}{\det \mathbf{C}^y}, \quad (4.20)$$

where J is the negentropy, \mathbf{C}^y is the covariance matrix of \mathbf{y} , and the c_{ii} are its diagonal elements. Since the observation is already whitened in the preprocessing step, the correlations in the data are removed and the third term in Equation 4.20 equals zero. Equation 4.20 reduces to:

$$I(y_1, \dots, y_N) = J(\mathbf{y}) - \sum J(y_i). \quad (4.21)$$

Since negentropy is invariant to linear transforms, $J(\mathbf{y})$ is a fixed term. Thus, the minimization of mutual information is equivalent to the maximization of negentropy.

4.3 Illustration of ICA

Depending on whether their distributions are *flatter* or *spikier* than the gaussian distribution, random variables can be generally categorized into three categories: *subgaussian*, *gaussian*, and *supergaussian* variables. Because ICA cannot separate mixtures of gaussian variables [103], we will illustrate the theories discussed in the previous subsection, using experiments on mixtures of subgaussian data and on mixtures of supergaussian data.

4.3.1 With subgaussian data

Random variables that have a negative kurtosis are called subgaussian. A typical example of subgaussian distribution is the uniform distribution $p(y)$:

$$p(y) = \begin{cases} \frac{1}{2b} & \text{if } -b \leq y \leq b \\ 0 & \text{elsewhere} \end{cases}, \quad (4.22)$$

where the parameter b determines the width of the probability density function. Let's consider two independent components s_i with uniform distribution:

$$p(s_i) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{if } -\sqrt{3} \leq s_i \leq \sqrt{3} \\ 0 & \text{elsewhere} \end{cases}, \quad (4.23)$$

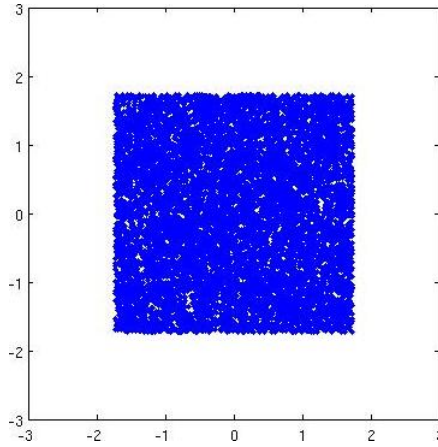
where the value of the parameter b is chosen as $\sqrt{3}$, so that s_i will have unit variances. The joint distribution of those independent source components s_1 and s_2 , are shown in Figure 4.1(a). As we can see from the figure, the joint distribution of two uniform distributed components s_1 and s_2 , appears as a square. The marginal histogram of s_1 and that of s_2 are shown in Figure 4.1(b) and Figure 4.1(c), respectively. They are much more “flat” in comparison with the gaussian distribution. The two components also have negative kurtosis values.

After the independent components are mixed using Equation 4.3, the newly mixed observations, x_1 and x_2 , shown in Figure 4.2.(a), no longer have the square shape. The marginal histogram of x_1 and x_2 , are shown in Figure 4.2.(b) and Figure 4.2.(c), respectively. Note they are closer to the gaussian distribution than those of s_1 and s_2 . That is in

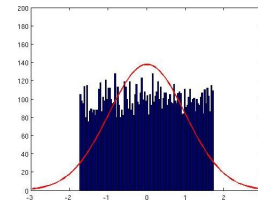
accordance with the central limit theory that mixed data tends to be more gaussian. The two mixtures also show kurtosis values closer to zero as expected.

The joint distribution of the whitened data, z_1 and z_2 , is shown in Figure 4.3.(a). We can see that the joint distribution of the whitened data has a square shape as in Figure 4.1(a), but it is rotated. Therefore, the degree of that rotation is the only thing remains to be estimated, given the already whitened data, z_1 and z_2 . We can recall from Subsection 4.2.3.4, that whitening reduces the degree of freedom, or equivalently, the unknown parameters to $n(n-1)$. In this case, $n = 2$ and there is only one remaining unknown parameter: the degree of rotation of the orthogonal matrix $\tilde{\mathbf{M}}$, or equivalently, that of its inverse $\tilde{\mathbf{W}}$.

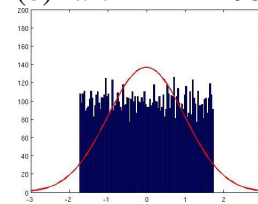
Using one of the estimation criteria discussed in Subsection 4.2.4 and Subsection 4.2.5, the degree of rotation can be estimation. The joint distribution of the estimated independent components y_1 and y_2 , are shown in Figure 4.4.(a). Because of the ambiguities of ICA, we cannot tell, if y_1 corresponds to s_1 or s_2 .



(a)The joint distribution of the independent components s_1 and s_2 with uniform distributions. Horizontal axis: s_1 , vertical axis: s_2 .

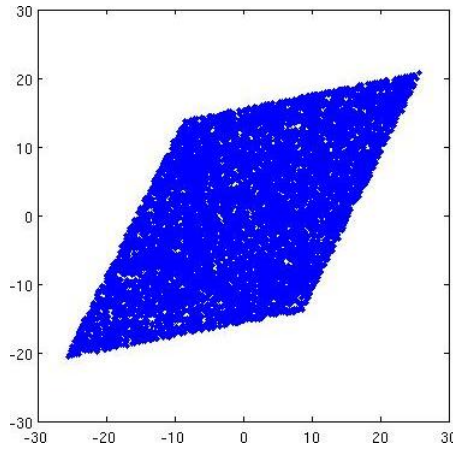


(b) $Kurt = -1.2195$



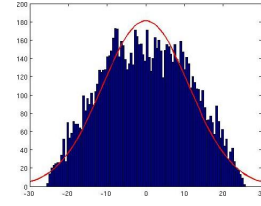
(c) $Kurt = -1.2171$

Figure 4.1: (a)The joint distribution of the independent components s_1 and s_2 with uniform distributions; (b)Histogram of s_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c)Histogram of s_2 , in comparison with gaussian distribution with unit variance (shown in red colour).

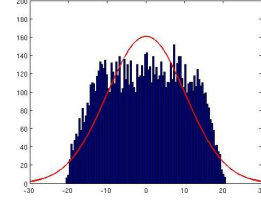


(a) The joint distribution of the mixtures x_1 and x_2 .

Horizontal axis: x_1 , vertical axis: x_2 .

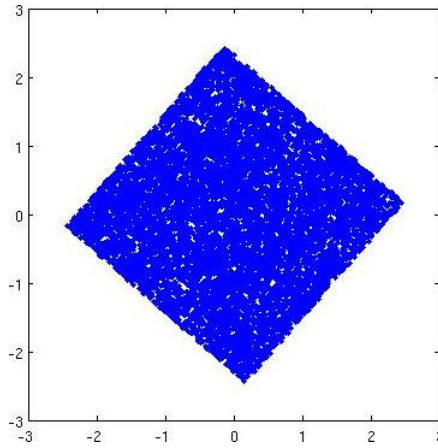


(b) $Kurt = -0.8428$

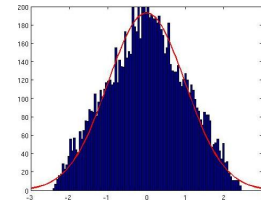


(c) $Kurt = -1.1300$

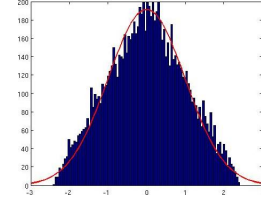
Figure 4.2: (a) The joint distribution of the mixtures x_1 and x_2 ; (b) Histogram of x_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c) Histogram of x_2 , in comparison with gaussian distribution with unit variance (shown in red colour).



(a) The joint distribution of the whitened mixtures z_1 and z_2 . Horizontal axis: z_1 , vertical axis: z_2 .

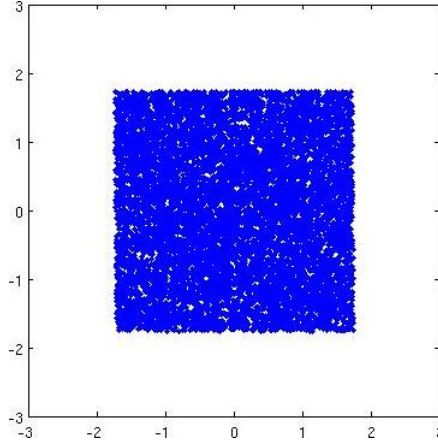


(b) $Kurt = -0.5892$



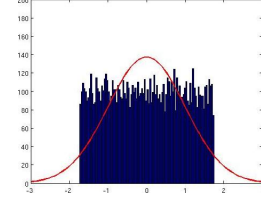
(c) $Kurt = -0.6352$

Figure 4.3: (a) The joint distribution of the mixtures z_1 and z_2 ; (b) Histogram of z_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c) Histogram of z_2 , in comparison with gaussian distribution with unit variance (shown in red colour).

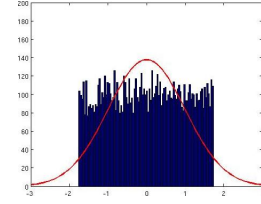


(a) The joint distribution of the estimated independent components y_1 and y_2 with uniform distributions.

Horizontal axis: y_1 , vertical axis: y_2 .



(b) $Kurt = -1.2174$



(c) $Kurt = -1.2186$

Figure 4.4: (a) The joint distribution of the estimated independent components y_1 and y_2 with uniform distributions; (b) Histogram of y_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c) Histogram of y_2 , in comparison with gaussian distribution with unit variance (shown in red colour).

4.3.2 With supergaussian data

Random variables with a positive kurtosis are called supergaussian. A typical example of a supergaussian distribution is the Laplacian distribution $p(y)$:

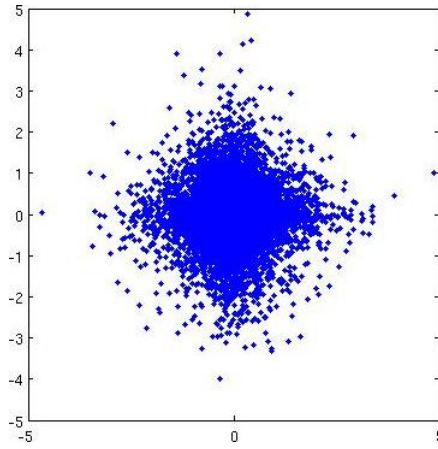
$$p(y) = \frac{\lambda}{2} \exp(-\lambda |y|), \dots (\lambda > 0) \quad (4.24)$$

where the parameter λ determines the width and the height of the peak of the probability density function. Let's consider two independent components s_i with Laplacian distribution:

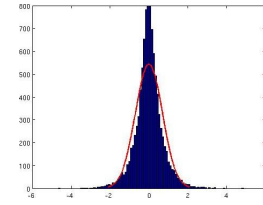
$$p(s_i) = \frac{\sqrt{2}}{2} \exp(-\sqrt{2} |s_i|), \quad (4.25)$$

where the value of the parameter λ is chosen as $\sqrt{2}$, so that s_i will have unit variances. The illustration of ICA on supergaussian data, is similar to those on subgaussian data. Figures

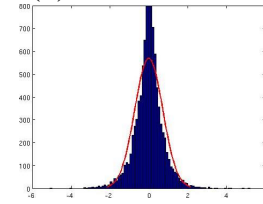
4.5-4.8 show the joint distribution and the marginal histograms, of the original independent components, the mixtures, the mixtures after whitening, and the final estimated independent components, respectively. We can see again the marginal histograms of the variables become more gaussian when mixed, and become less gaussian after ICA estimation. Also, whitening transform the data into its original shape and the estimation of the rotation matrix $\widetilde{\mathbf{W}}$ will transform the data back to its original position.



(a) The joint distribution of the independent components s_1 and s_2 with Laplacian distributions. Horizontal axis: s_1 , vertical axis: s_2 .

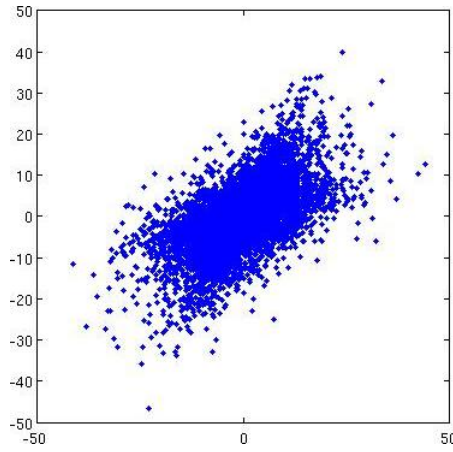


(b) $Kurt = 2.8108$



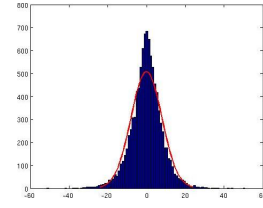
(c) $Kurt = 2.9974$

Figure 4.5: (a) The joint distribution of the independent components s_1 and s_2 with Laplacian distributions; (b) Histogram of s_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c) Histogram of s_2 , in comparison with gaussian distribution with unit variance (shown in red colour).

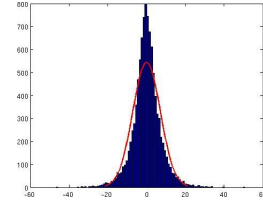


(a)The joint distribution of the mixtures x_1 and x_2 .

Horizontal axis: x_1 , vertical axis: x_2 ;

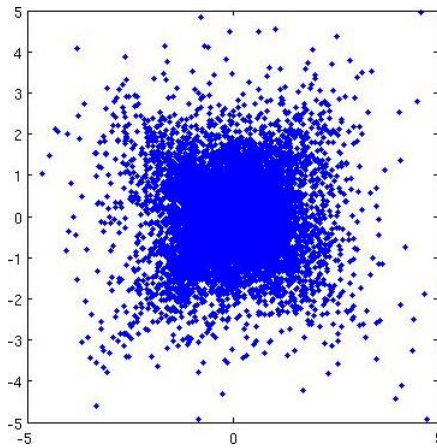


(b)Kurtosis=1.9348



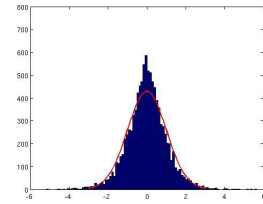
(c)Kurtosis=2.5664

Figure 4.6: (a)The joint distribution of the mixtures x_1 and x_2 ; (b)Histogram of x_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c)Histogram of x_2 , in comparison with gaussian distribution with unit variance (shown in red colour).

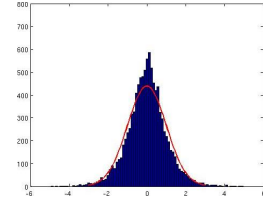


(a)The joint distribution of the mixtures z_1 and z_2 .

Horizontal axis: z_1 , vertical axis: z_2 ;

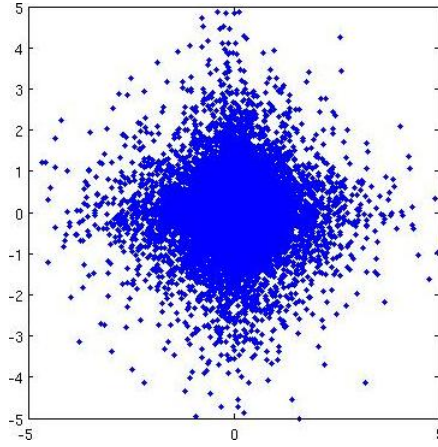


(b) $Kurt = 1.5840$

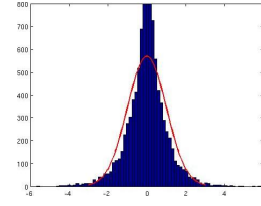


(c) $Kurt = 1.3603$

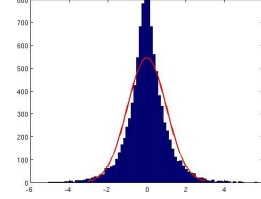
Figure 4.7: (a)The joint distribution of the mixtures z_1 and z_2 ; (b)Histogram of z_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c)Histogram of z_2 , in comparison with gaussian distribution with unit variance (shown in red colour).



(a) The joint distribution of the estimated independent components y_1 and y_2 with Laplacian distributions.
Horizontal axis: y_1 , vertical axis: y_2 ;



(b) $Kurt = 2.8119$



(c) $Kurt = 2.9940$

Figure 4.8: (a) The joint distribution of the estimated independent components y_1 and y_2 with Laplacian distributions; (b) Histogram of y_1 , in comparison with gaussian distribution with unit variance (shown in red colour); (c) Histogram of y_2 , in comparison with gaussian distribution with unit variance (shown in red colour).

4.4 Summary

ICA is a relatively new statistical technique that has many useful applications. ICA can find the underlining independent components from mixtures but cannot determine their variances, signs or order. Whitening, as an important preprocessing step, greatly simplifies the ICA estimation. After whitening, instead of having to estimate the matrix \mathbf{W} with n^2 unknown elements, only the orthogonal matrix $\widetilde{\mathbf{W}}$ with $n(n-1)$ degrees of freedom has to be estimated. Since an orthogonal transformation is either a rigid rotation or a rotation followed by a flip, the difference between the whitened variables and the final estimated independent components are either only a rotation or a rotation followed by a flip. The core part of the ICA algorithm begins after the whitening of data. To estimate the final rotation matrix $\widetilde{\mathbf{W}}$, different principles, such as maximization of non-gaussianity and minimization of mutual information, are used to construct the objective functions in the estimation algorithms. Both

ICA and whitening are used in developing new affine invariant shape descriptors, which we will discuss in the next chapters.

Chapter 5

Novel Contour-Based Affine Invariant Shape Descriptors

5.1 Introduction

In the previous chapter, we have reviewed ICA, its estimation principles and its relationship with whitening. In this chapter, we introduce two newly developed affine invariant contour-based shape descriptors, the ICA Fourier shape descriptor (ICAFSD) and the whitening Fourier shape descriptor (WFSD). These two affine invariant descriptors utilize either ICA or whitening, and are more powerful than most of the existing ones. The rest of the chapter is organized as follows: Section 5.2 explains how a shape contour can be canonicalized into a standard position by ICA. Section 5.3 explains how the new ICAFSD is extracted from the canonicalized shape contour. In Section 5.4, the WFSD is introduced. Section 5.5 shows the results of the retrieval experiments using those newly developed shape descriptors. Section 5.6 summarizes this chapter.

5.2 Canonicalization of shape contour by ICA

The extraction of the ICAFSD, has two major steps. The first is the canonicalization of shape contour by ICA.

In order to compare shapes from a database where they may be affine related, we can first transform the shape contours into their canonical form using ICA. Let us consider a two source, two mixture ICA mixing model:

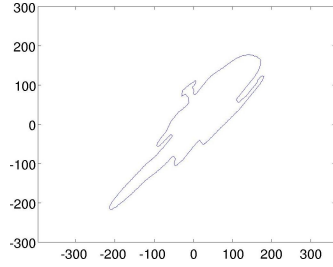
$$\mathbf{x}[k] = \mathbf{M} \cdot \mathbf{s}[k], \quad (5.1)$$

where $\mathbf{s}[k] = [s_1[k], s_2[k]]^T$ represent the vectors of the two independent, unit variance, source data, $\mathbf{x}[k] = [x_1[k], x_2[k]]^T$ represent the vectors of the two mixtures, k is the data index, and \mathbf{M} is a 2×2 mixing matrix. Given the observed mixtures $x_1[k]$ and $x_2[k]$, ICA can extract the two concealed "independent components" $s_1[k]$ and $s_2[k]$, regardless of what mixing matrix \mathbf{M} was used. If the two mixtures $x_1[k]$ and $x_2[k]$ are the pixel coordinates of the shape contour, and $\mathbf{x}_a[k] = [x_{a1}[k], x_{a2}[k]]^T$ are the pixel coordinates of an affine related shape contour, related by the transform matrix \mathbf{A} , from Equation 3.3 and Equation 5.1, we have

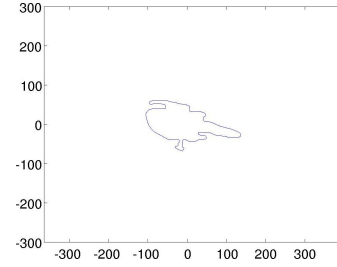
$$\mathbf{x}_a[k] = \mathbf{A} \cdot \mathbf{x}[k] = \underbrace{\mathbf{A} \cdot \mathbf{M}}_{\mathbf{M}_a} \cdot \mathbf{s}[k] = \mathbf{M}_a \cdot \mathbf{s}[k], \quad (5.2)$$

where \mathbf{M}_a is a new mixing matrix, combined of \mathbf{A} and \mathbf{M} . ICA will extract the same independent components $\mathbf{s}[k]$, whether \mathbf{M} or \mathbf{M}_a is the mixing matrix, and whether $\mathbf{x}[k]$ or $\mathbf{x}_a[k]$ are the observed mixtures. The extracted independent components are the same only in a general sense, their orders and signs cannot be determined. That is because of the order, sign and scale ambiguities of ICA [95], which are reduced to the order and sign ambiguities as the source components are fixed to have unit variance. The extracted independent components, $\mathbf{s}[k] = [s_1[k], s_2[k]]^T$, are used as the pixel coordinates of the shape contour in its canonical form.

Figures 5.1 (a) and (b), show the contours of two affine related helicopter shapes. Figures 5.2 (a-h) show eight possible ICA-canonicalized shape contours transformed from the shape contour in Figure 5.1 (a). Those eight possible ICA-canonicalized shape contours are the results of different trials of ICA estimations. We can see that those contours are related by a $n \cdot 90$ ($n=1, 2$, or 3) degree of rotation. For example, Figure 5.2 (a), (c), (d) and (h) are 90, 180, or 270 degree rotated version of each others. The reason for the variation is the order ambiguity of ICA, i.e., the x and y coordinates of the ICA-canonicalized



(a) Helicopter 1



(b) Helicopter 2

Figure 5.1: Contours of two affine-related shapes (helicopter)

shape contour can be switched. Similarly, because of the sign ambiguity of ICA, the ICA-canonicalized shape contour can be flipped against the x axis and/or y axis. For example, Figure 5.2 (a) is an x -axis flipped version of (f). When the effects of the sign and order ambiguities combine, two ICA-canonicalized shape contours transformed from the same shape contour are related by a rotation and flips. Because of the two ambiguities of ICA, for each shape contour, there are in total eight possible variations of ICA-canonicalized shape contours related by a $n \cdot 90$ ($n=1, 2$, or 3) degree rotation and/or flips.

Figures 5.3 (a-h) show eight possible ICA-canonicalized shape contours transformed from the shape contour in Figure 5.1 (b). As we expect, the shape contours in Figures 5.2 and in Figures 5.3, which are the ICA-canonicalized shape contours of the two affine related shapes in Figures 5.1 (a) and (b), are very similar. Each contour in Figure 5.3 is practically identical to those in Figure 5.2; the only difference is a flip or rotation.

Figures 5.4 (a) and (b), show the contour of a horseshoe shape, and one of its ICA canonicalized shape contour, respectively. We can see that the ICA canonicalized shape contour of the horseshoe, is very different from those of the helicopter.

5.3 ICA Fourier shape descriptor

In the last section, we saw that ICA can transform shape contours into canonical shape contours, eliminating the scaling and skew effect of any affine transforms. However, the contours in the canonical form are not yet ready for comparison, because of the following problems:

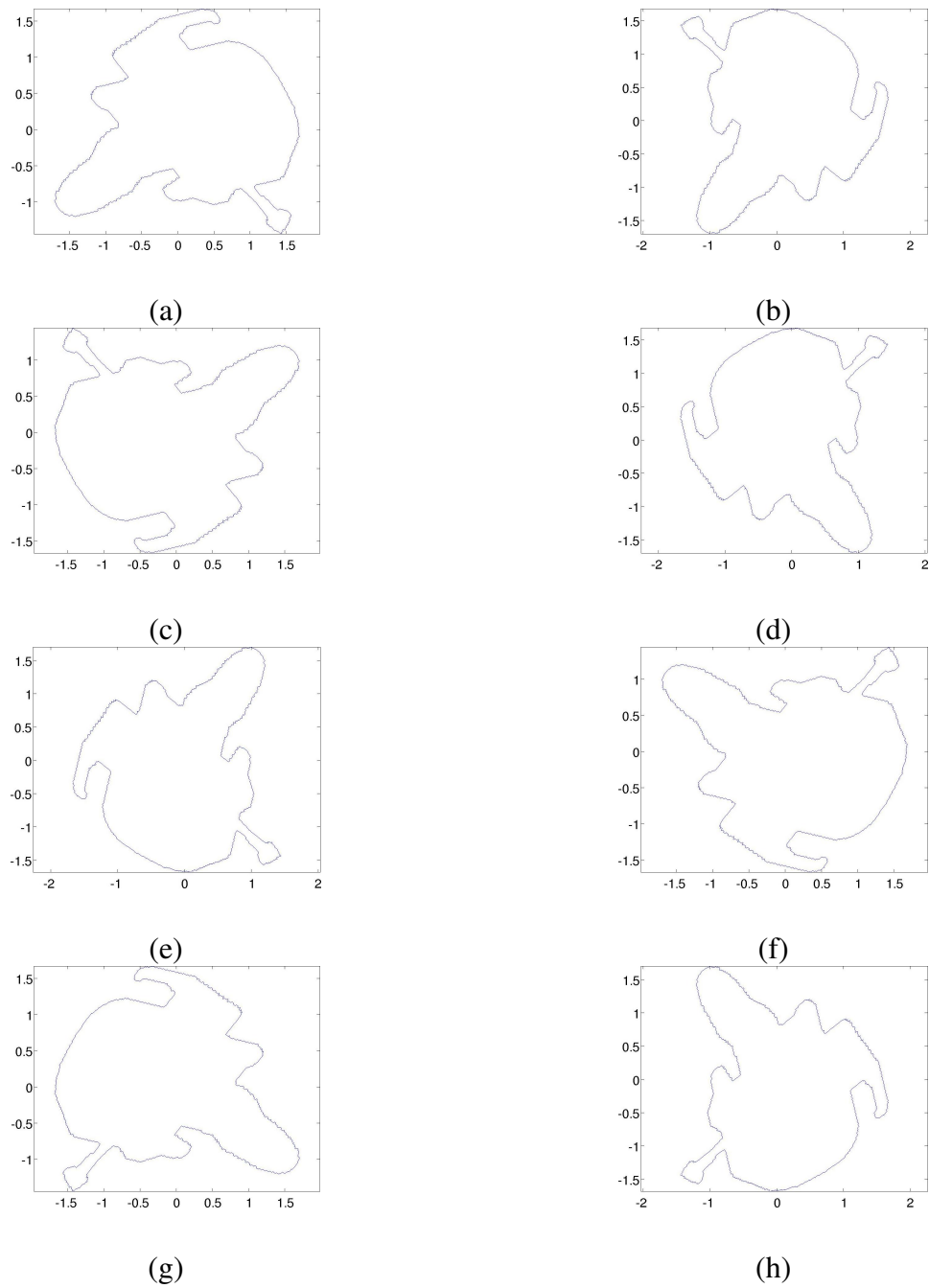
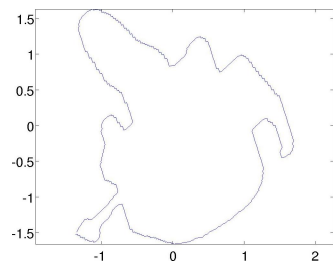
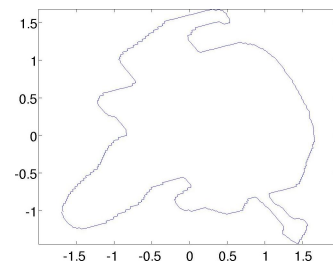


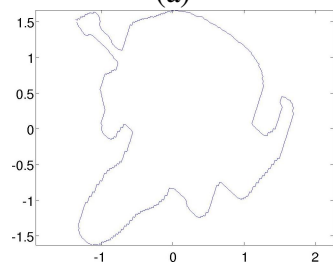
Figure 5.2: ICA canonicalized shapes corresponding to the shape in Figure 5.1 (a)



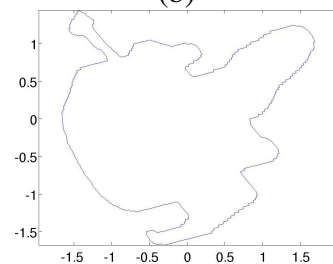
(a)



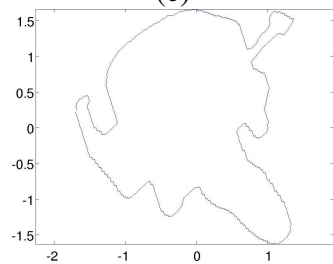
(b)



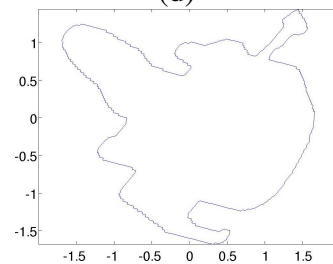
(c)



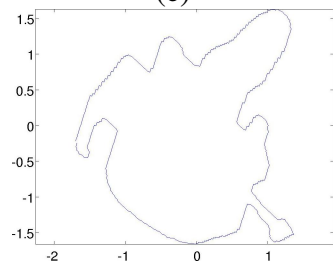
(d)



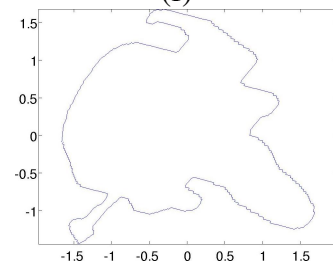
(e)



(f)



(g)



(h)

Figure 5.3: ICA canonicalized shapes corresponding to the shape in Figure 5.1 (b)

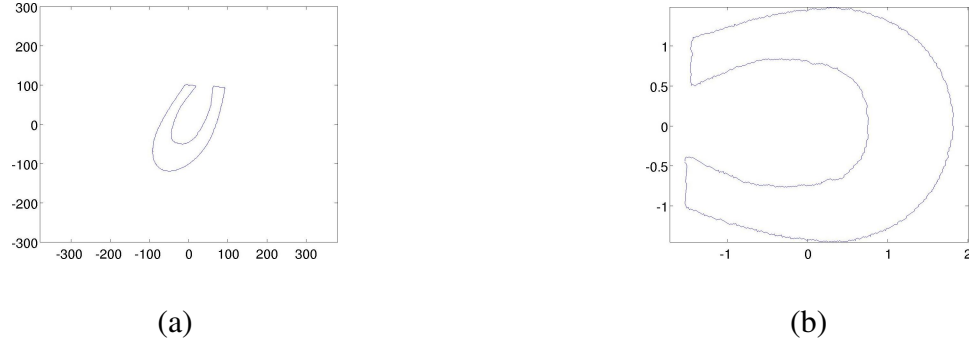


Figure 5.4: (a) Shape contour of a horse shoe, and (b) its ICA-canonicalized shape contour

1. The sample points of two canonical shape contours do not correspond to each other exactly, even though they are affine related.
2. There are the coordinate and sign ambiguities in the canonical shape contours, which are inherited from the order and sign ambiguities of ICA.
3. There is a mismatch of starting points since the original shape contours are different and the contour extraction algorithms can start from any point.

To solve the first problem, the coordinate points are resampled along the length of the shape contour in its canonical form. To solve the second and third problems, we extract the centroid distance of the resampled contour and then perform the Fourier transformation on the centroid distance. Let's first review the Fourier transform (FT), and its discrete form - the discrete Fourier transform (DFT).

5.3.1 FT and DFT

The FT is defined as [104]:

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \omega} dx, \quad (5.3)$$

where $f(x)$ is the original function, x is the variable in the time domain, $F(\omega)$ is the FT of $f(x)$, ω is the variable in the frequency domain and $i = \sqrt{-1}$. The inverse transform of the FT is [104]:

$$f(x) = \int_{-\infty}^{\infty} F(\omega) e^{2\pi i x \omega} d\omega. \quad (5.4)$$

As we are dealing with digital images, we need to use the discrete form of the FT, the DFT [37], which is defined as [37]:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{2\pi i}{N} kn} \quad (k = 0, \dots, N-1), \quad (5.5)$$

where $x[n]$ is the original function, n is the index in the time domain, $X[k]$ is the DFT of $x[n]$, k is the index in the frequency domain and $i = \sqrt{-1}$. The inverse transform of DFT is [37]:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{\frac{2\pi i}{N} kn} \quad (n = 0, \dots, N-1). \quad (5.6)$$

The DFT has the following important properties [37], which we will later explore to solve the ambiguities and the starting point mismatching problems mentioned before:

$$|X_1[k]| = |X_2[k]|, \quad \text{if } x_1[n] = x_2[(n)_N], \quad (5.7)$$

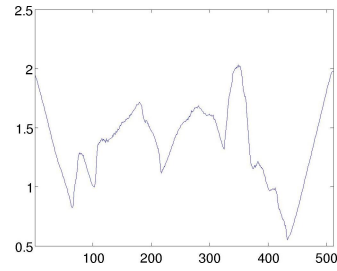
and

$$|X_1[k]| = |X_2[k]|, \quad \text{if } x_1[n] = x_2[(n-m)_N], \quad (5.8)$$

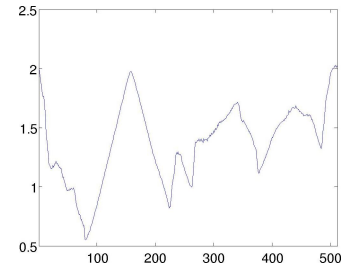
where n and N are the index and the length of the sequence, respectively. $((n))_N$ represents modulo N for n , and m corresponds to a circular shift.

5.3.2 DFT on ICA-canonicalized shape contour

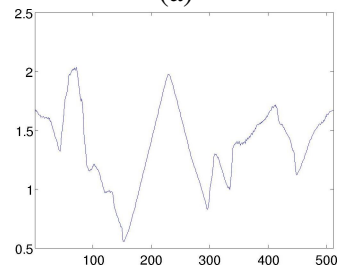
Figures 5.5 (a-h) show the centroid distances of the ICA-canonicalized shape contours shown in Figures 5.2 (a-h), respectively. Because of the ambiguities of the canonical shape contours, and because of the mismatch of starting points, the centroid distances of the ICA-canonicalized shape contours are either the same or different by a circular shift and/or



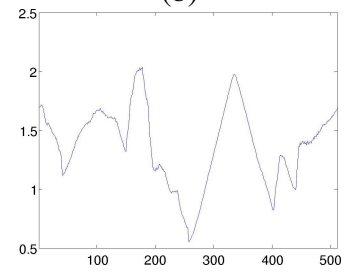
(a)



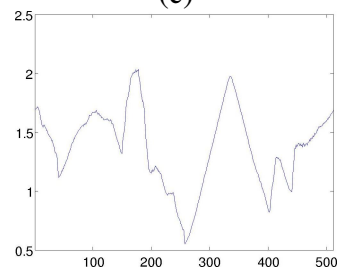
(b)



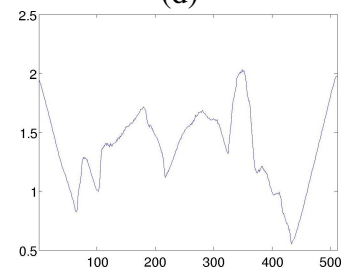
(c)



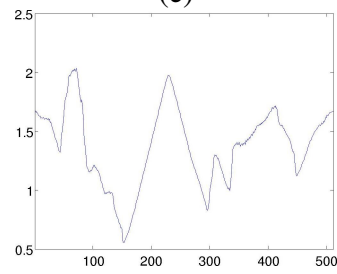
(d)



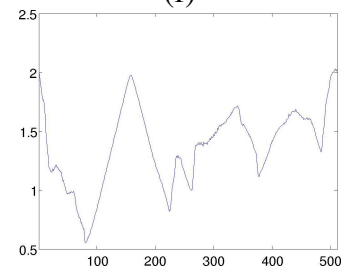
(e)



(f)

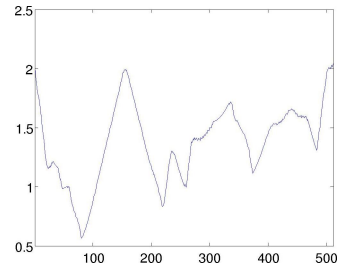


(g)

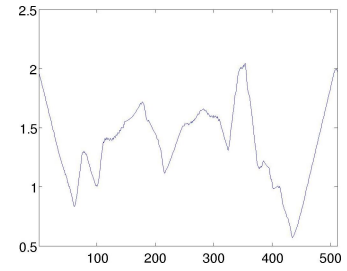


(h)

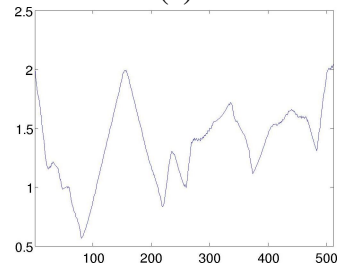
Figure 5.5: Centroid distances, corresponding to the ICA canonicalized shapes in Figure 5.2



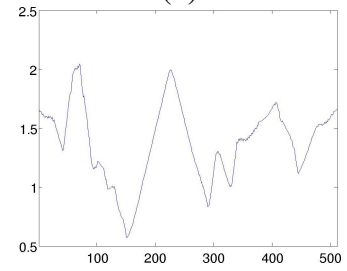
(a)



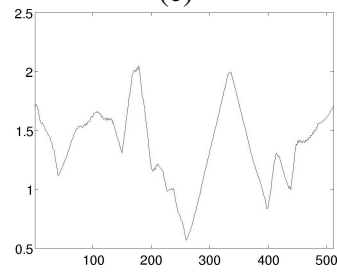
(b)



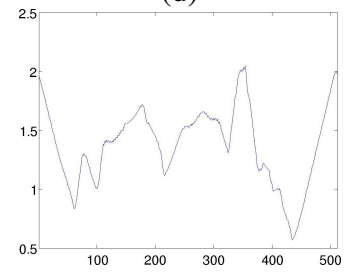
(c)



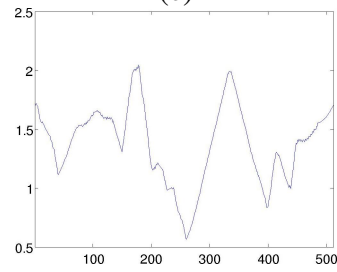
(d)



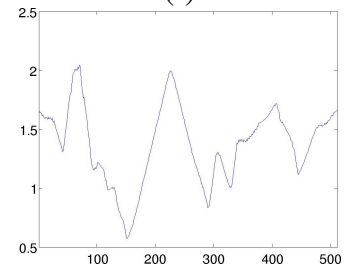
(e)



(f)



(g)



(h)

Figure 5.6: Centroid distances, corresponding to the ICA canonicalized shapes in Figure 5.3

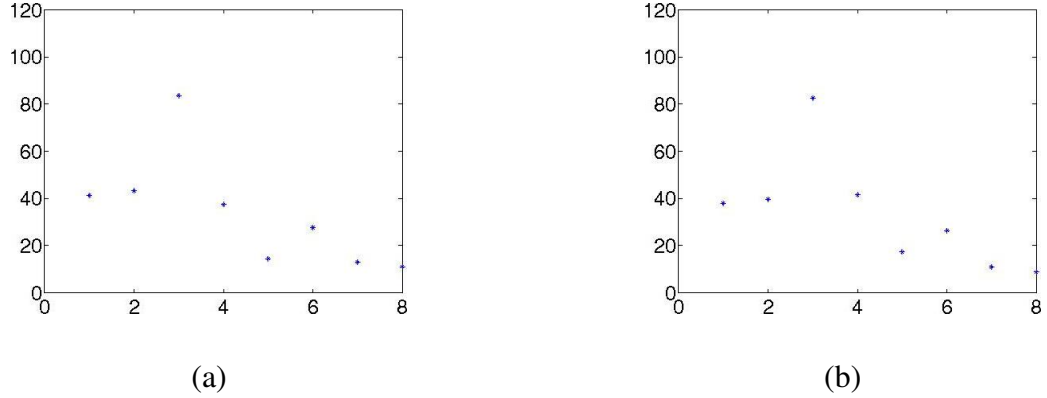


Figure 5.7: ICAFSDs of the two affine-related shapes (helicopter) shown in Figure 5.1

a reflection. That can be observed, when we compare any two centroid distances x_a and x_b , in Figure 5.5. Mathematically, their relationship can be expressed as:

$$x_a[n] = x_b[((n - m))_N] \quad (5.9)$$

or

$$x_a[n] = x_b[(-(n - m))_N]. \quad (5.10)$$

Now, we will utilize the properties of the DFT and apply the DFT to the centroid distances. Because of Equation 5.7 and Equation 5.8, whether $x_a[n] = x_b[((n - m))_N]$ or $x_a[n] = x_b[(-(n - m))_N]$, $|X_a[k]|$ always equals to $|X_b[k]|$. The magnitudes of the first eight components of the DFTs, $|X[k]|$, are our proposed affine invariant ICA Fourier shape descriptor (ICAFSD), and there is only one single ICAFSD for each shape, even though it has eight possible ICA-canonicalized shape contours. Figure 5.7 (a) shows the ICAFSD of the shape shown in Figure 5.1 (a).

As we know from Section 5.2, the ICA-canonicalized shape contours in Figure 5.2 are related to the ICA-canonicalized shape contours in Figure 5.3, by a rotation and/or flip. Because of that, any of their corresponding centroid distances shown in Figure 5.5 and in Figure 5.6 are also related, either by Equation 5.9 or by Equation 5.10. Thus, the two corresponding ICAFSDs of the two affine related helicopter shapes are also the same, as we can observe from Figures 5.7 (a) and (b). At the same time they are different from the ICAFSD of the horseshoe shape, shown in Figure 5.8. This can be seen more clearly, when

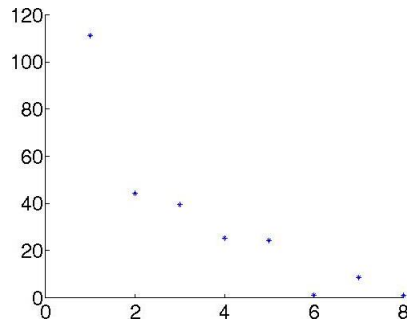


Figure 5.8: ICAFSD of the horseshoe shape shown in Figure 5.4 (a)

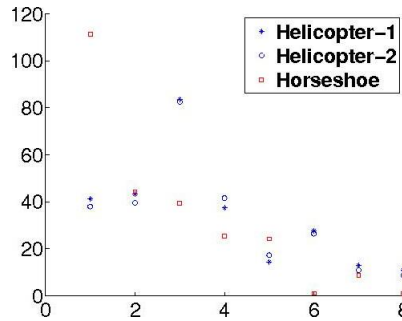


Figure 5.9: Comparison of the ICAFSDs

we compare those three ICAFSDs in the same figure (Figure 5.9). Thus, the ICAFSD can be used to discriminate between affine related shape images and unrelated ones.

5.4 Whitening Fourier shape descriptor

As we have discussed in Chapter 4, the difference between the whitened variables and the final estimated independent components are either a rotation or a rotation followed by a flip. Given the relationship between whitening and ICA, we further proposed the whitening Fourier shape descriptor (WFSD) which has descriptor extraction steps, shown in Figure 5.11, similar to that of its ICA counterpart. The only difference is that only whitening will be performed on the shape contour data and the whitened data are not further rotated and/or flipped as in the ICAFSD. Figure 5.12(a) shows the same shape contour in Figure 5.1(b). Figure 5.12(b) shows its shape contour data whitened using PCA and normalization. Comparing the whitened shape contour in Figure 5.12(b) and the ICA-ed shape contours in

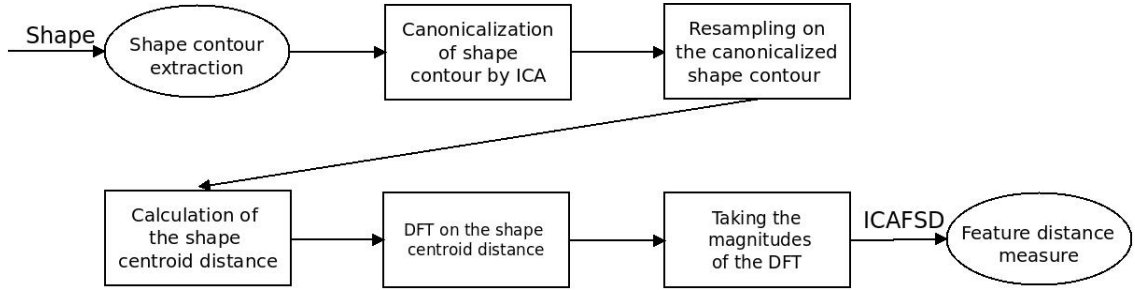


Figure 5.10: Diagram of ICA-Fourier shape descriptor extraction

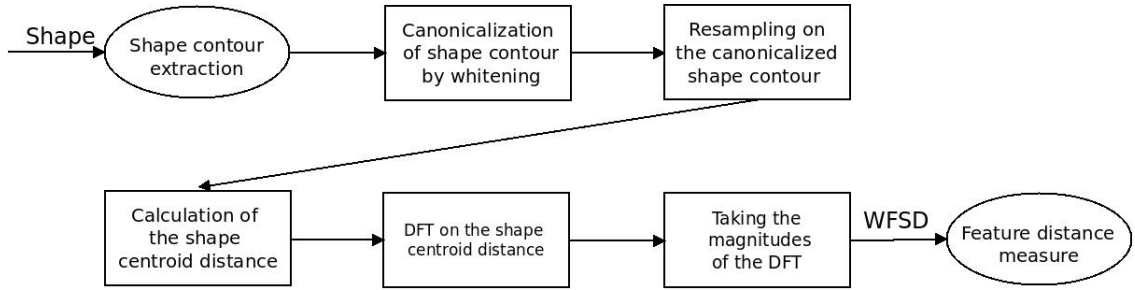


Figure 5.11: Diagram of whitening-Fourier shape descriptor extraction

Figure 5.2, we find that the ICA-ed ones are the rotated, or rotated and flipped, version of the whitened one. However, such a rotation and flip will not be necessary, as it will turn into a circular shift of the index in the centroid distance and will not effect the magnitudes of the Fourier transform. As we can compare the WFSD in Figure 5.12(d) and the ICAFSDs in Figure 5.7 (b), they are almost the same. The proposed WFSD avoids the calculation of the rotation, which is the most computationally costly step in ICA, while still having similar performance as its ICA counterpart.

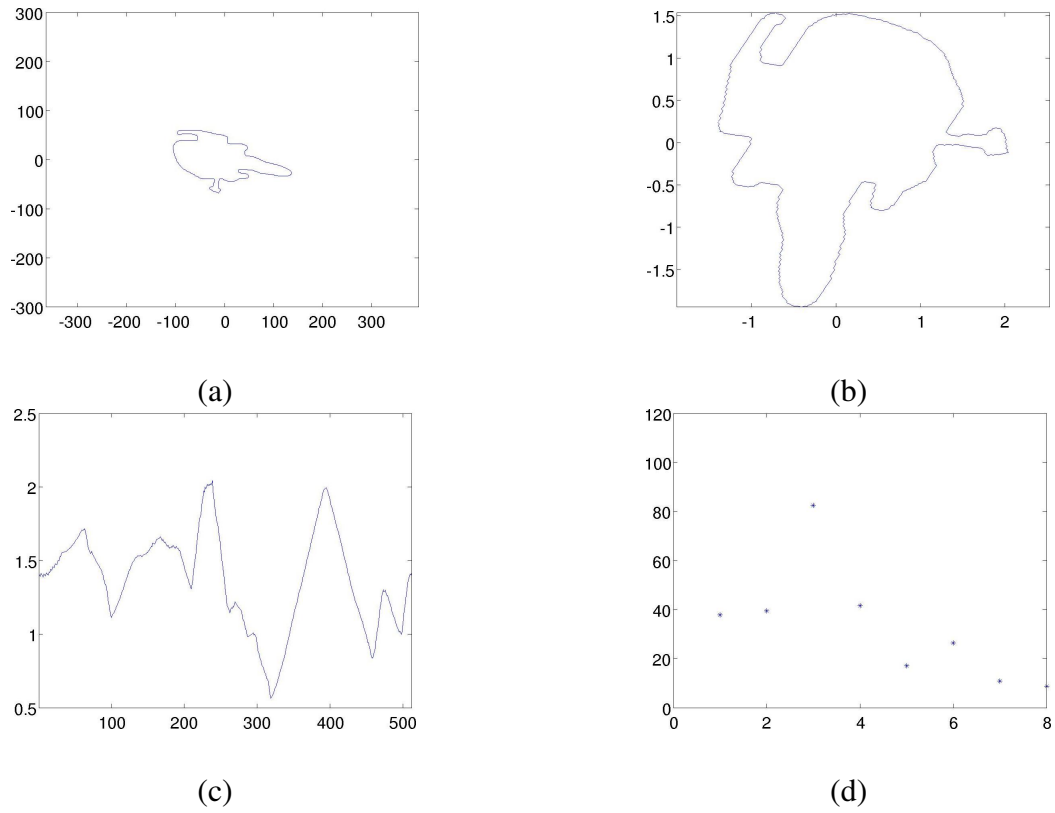


Figure 5.12: (a) shape contour, (b) whitened shape contour, (c) centroid distance and (d) WFSD

5.5 Experimental results

The newly developed ICAFSD and WFSD are tested and compared with those previously known affine invariant shape representations, studied in Chapter 3. The AIWSR and the AMI-TC are not included in the comparison of retrieval accuracy, because of their poor retrieval performances.

The ICA shape signature (ICASS) [64, 65] is another affine-invariant shape representation that employs ICA. However, because of the inherited order and sign ambiguities of ICA, and the mismatching of the starting points, an exhaustive trial approach needs to be used to compare two ICASSs. That requires lots of computational time and is not suitable for online retrieval applications. The ICASS is also tested and compared in the experiments.

5.5.1 Experimental data

The experiments use the same simple shape databases in Section 3.

5.5.2 Retrieval accuracy

Figure 5.13 shows the average precision-recall graph of the affine-invariant shape representations tested on the noise free shape database. As we can see, the newly developed ICAFSD performs better than the AMI-FS and the ACSSSD, but worse than the AIFSD and the ICASS. Figures 5.14-5.19 show the precision-recall graphs of the affine-invariant shape representations tested on the simple shape database with different levels of added noise. The newly developed ICAFSD performs better than the AIFSD, on simple shape database with noise.

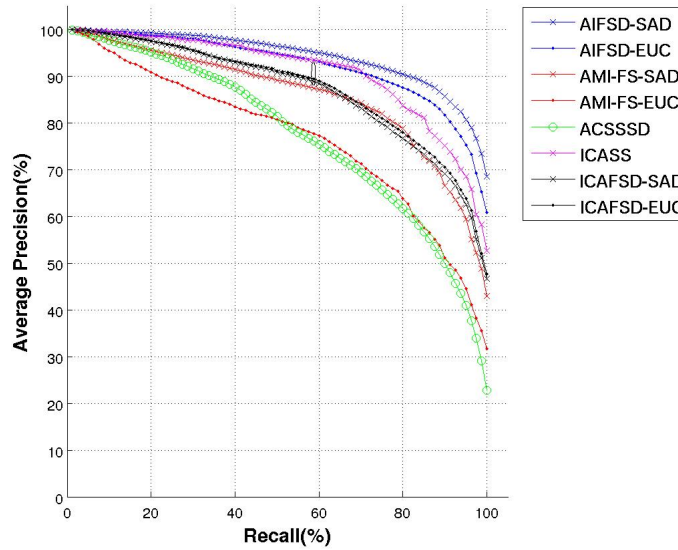


Figure 5.13: Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database without noise.

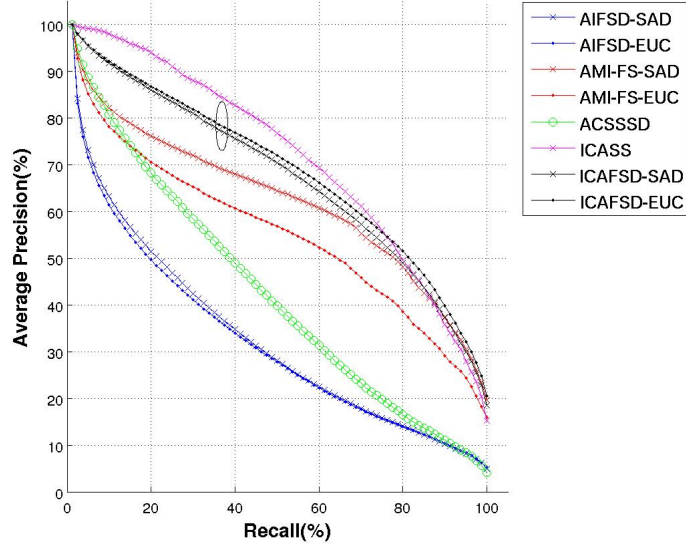


Figure 5.14: Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=30\text{dB}$.

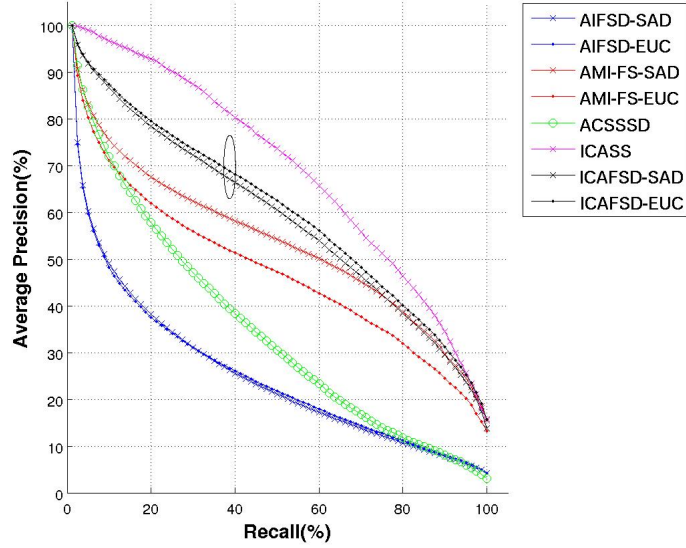


Figure 5.15: Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$.

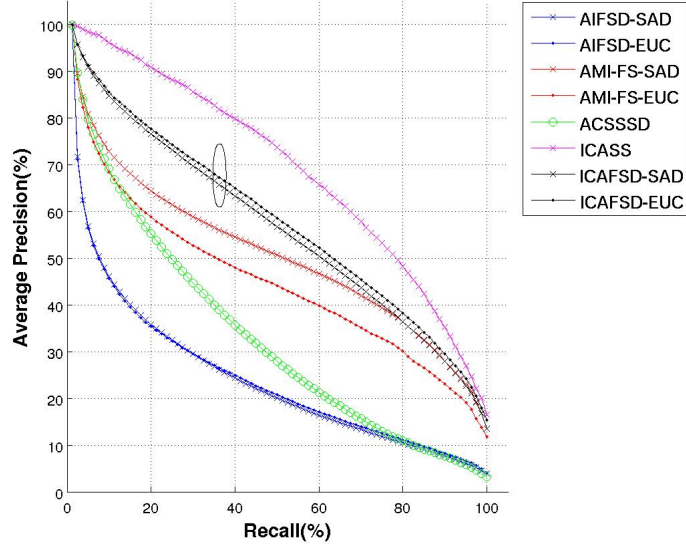


Figure 5.16: Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.0206\text{dB}$.

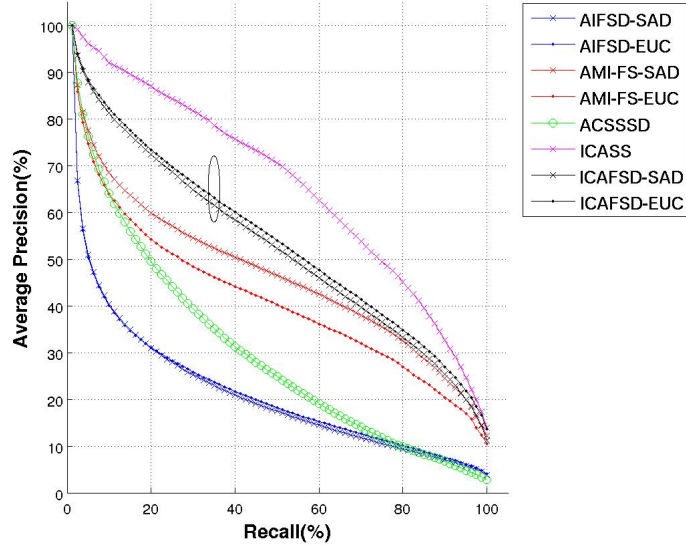


Figure 5.17: Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=24.7712\text{dB}$.

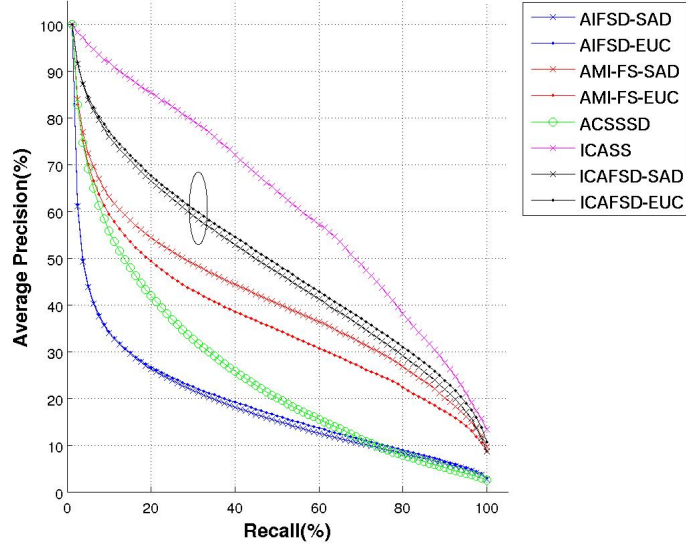


Figure 5.18: Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=23.0103\text{dB}$.

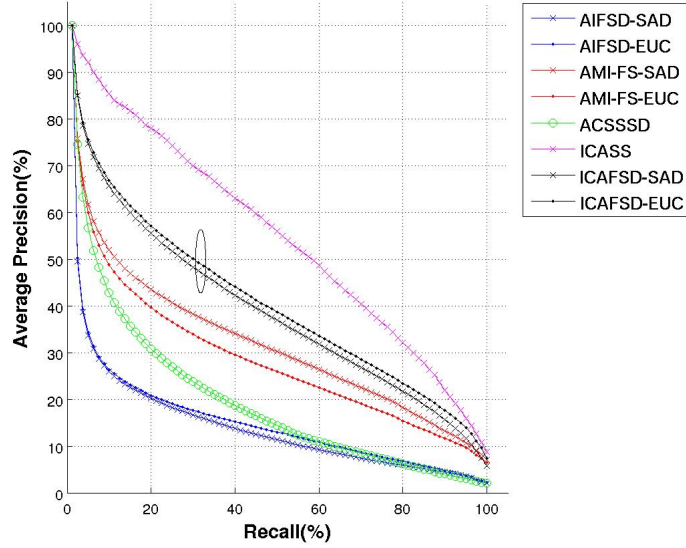


Figure 5.19: Average precision-recall graphs of retrievals using different affine invariant shape descriptors on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$.

The performances of the WFSD are not compared in the figures, as they have exactly the same precision-recall curves as the ICAFSD and the two curves will overlap each others in the graph. Figure 5.20 show the average precision-recall curve of the ICAFSD and that of the WFSD, tested on the same shape database. The two average precision-recall curves overlap each others, as expected.

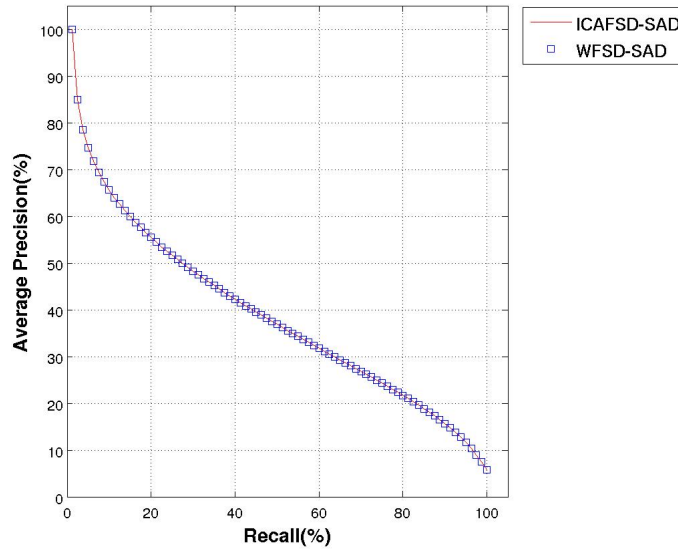


Figure 5.20: Overlapping of the average precision-recall curve of the ICAFSD and that of the WFSD (on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$).

5.5.3 Comparison of extraction time, distance calculation time, and compactness

The extraction time, distance calculation time, and compactness of the affine invariant shape representations are compared in Tables 5.1 - 5.3 and Figures 5.21 - 5.23

From Table 5.1 and Figure 5.21, we see that the ICAFSD requires 0.1390 seconds for extraction while the WFSD requires only 0.0164 seconds. From Table 5.2 and Figure 5.22, we see that the ICAFSD and the WFSD require far less time for feature comparison, than the AICSSSD, the ICASS and the AIWSD. From Table 5.3 and Figure 5.23, we see that

the ICAFSD and the WFSD are far more compact than the ICASS and the AIWSD, which are shape signatures instead of shape descriptors.

Shape representation	Extraction time(s)	Shape representation	Extraction time(s)	Shape representation	Extraction time(s)
AIFSD	0.0136	AICSSSD	13.9972	AMI-FS	0.1835
AIWSR	0.1377	AMI-TC	0.0521	ICASS	0.0178
<i>ICAFSD</i>	<i>0.1390</i>	<i>WFSD</i>	<i>0.0164</i>		

Table 5.1: Feature extraction time(seconds)

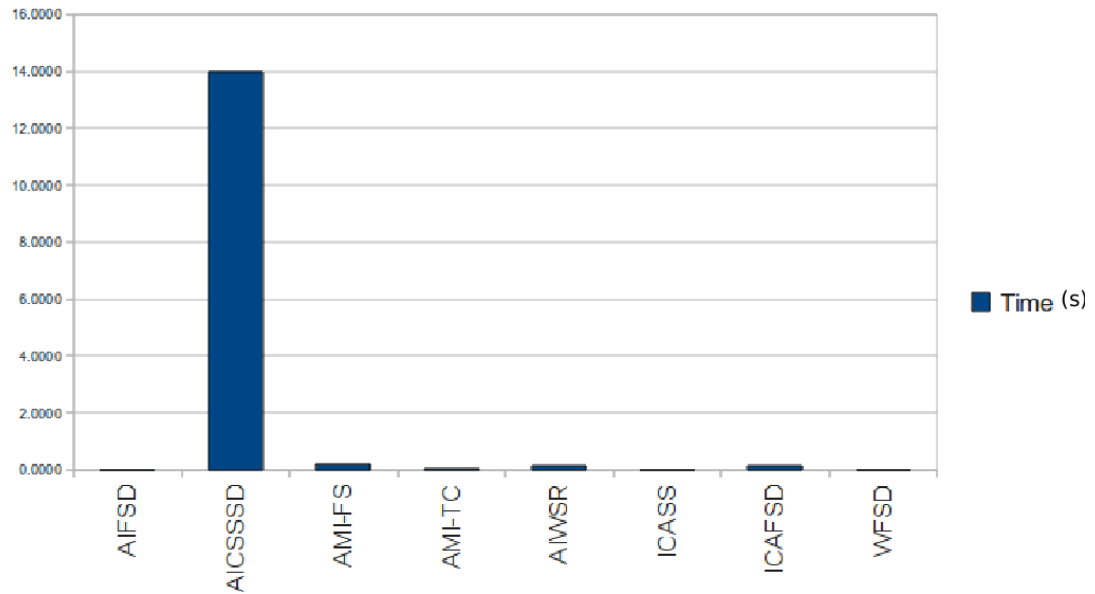


Figure 5.21: Comparison of feature extraction time (seconds)

Shape representation	Extraction time(ms)	Shape representation	Extraction time(ms)
AMI-TC-SAD	0.021	AMI-TC-EUC	0.006
AMI-FS-SAD	0.013	AMI-FS-EUC	0.005
AIFSD-SAD	0.009	AIFSD-EUC	0.006
<i>ICAFSD-SAD</i>	<i>0.009</i>	<i>ICAFSD-EUC</i>	<i>0.009</i>
<i>WFSD-SAD</i>	<i>0.009</i>	<i>WFSD-EUC</i>	<i>0.009</i>
AICSSSD	18.999	ICASS	148.886
AIWSR	120.478		

Table 5.2: Feature distance calculation time(milliseconds)

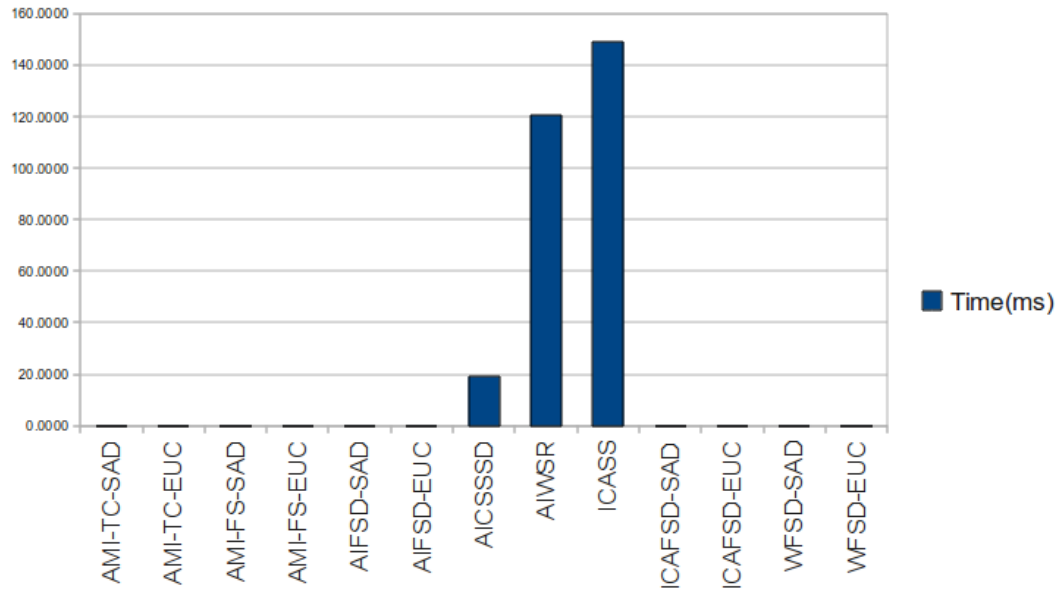


Figure 5.22: Comparison of feature distance calculation time (milliseconds)

Shape representation	Size	Shape representation	Size	Shape representation	Size
AIFSD	36	AICSSSD	(average) 5	AMI-FS	4
AIWSR	512	AMI-TC	8	ICASS	512
<i>ICAFSD</i>	8	<i>WFSD</i>	8		

Table 5.3: Size of the shape representations

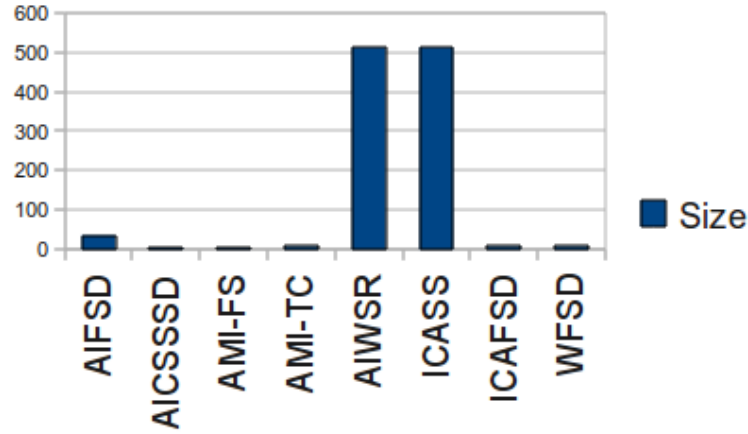


Figure 5.23: Comparison of the size of the shape representations

5.6 Summary

In this chapter, the ICAFSD and the WFSD were introduced. In extracting the ICAFSD, ICA is used first to transform the shape contour into canonicalized shape contour. The DFT is then applied on the centroid distance of the canonicalized shape contour. The magnitudes of the extracted DFT coefficients are the ICAFSD. Because of the relationship between ICA and whitening, we further proposed the WFSD as a simplified, but faster, method. The WFSD also solves the affine problem by transforming shape contours into canonicalized ones. But it used whitening only, instead of ICA. That avoids the estimation of the rotation in the ICA algorithm and saves computational time. The ICAFSD and the WFSD have better retrieval performance in comparison with most of the other shape representations, except the ICASS. However, both the ICAFSD and the WFSD are far more compact and require far less computational time for feature comparison than the ICASS.

Chapter 6

Novel Region-Based Affine Invariant Shape Descriptors

6.1 Introduction

In the previous chapter, we have introduced two contour-based affine invariant shape descriptors, which show robust performance in shape retrieval experiments. Contour-based affine invariant shape descriptors are limited that they can only be applied to simple shapes. In many situations, region-based affine invariant shape descriptors are preferred, as they can be applied to both simple and complex shapes. As we saw in Chapter 3, the performance of the previously known region-based affine invariant shape descriptors is not satisfactory. Therefore, there is a need to develop new robust region-based affine invariant shape descriptors. In this chapter, four new region-based affine invariant shape descriptors will be introduced. They are the ICA Zernike moment shape descriptor (ICAZMSD), the Whitening Zernike moment shape descriptor (WZMSD), the ICA orthogonal Fourier Mellin moment shape descriptor (ICAOFMMSD), the Whitening orthogonal Fourier Mellin moment shape descriptor (WOFMMSD). These four proposed shape descriptors show good performance in comparison with existing ones. The rest of the chapter is organized as follows: Section 6.2 explains how a shape can be canonicalized into a standard position by ICA. Sections 6.3 to 6.5, introduce the four newly developed region-based affine invariant shape descriptors:

the ICAZMSD, the WZMSD, the ICAOFMMSD and the WOFMMSD, respectively. Section 6.6 shows the results of the retrieval experiments using those newly developed shape descriptors. Section 6.7 shows the application of the newly proposed descriptors in traffic sign recognition. Section 6.8 summarizes this chapter.

6.2 Canonicalization of shape by ICA

The extraction of the two newly developed ICA-based affine invariant shape descriptors, have two major steps. The first step is the canonicalization of shape by ICA. In Chapter 5, we have seen how ICA can transform shape contours into canonical shape contours in extracting affine invariant *contour-based* shape descriptors. In order to develop new robust affine invariant *region-based* shape descriptors, it is natural also to think of using ICA as a canonicalizing tool for shapes, instead of shape contours. Here, we explain how ICA can turn shapes into their canonical shapes. The proof can be done in analogy to the proof for the contour-based one.

In order to compare shapes from a database where they may be affine related, we can first transform the shape into their canonical form using ICA. Let us consider a two source, two mixture ICA mixing model:

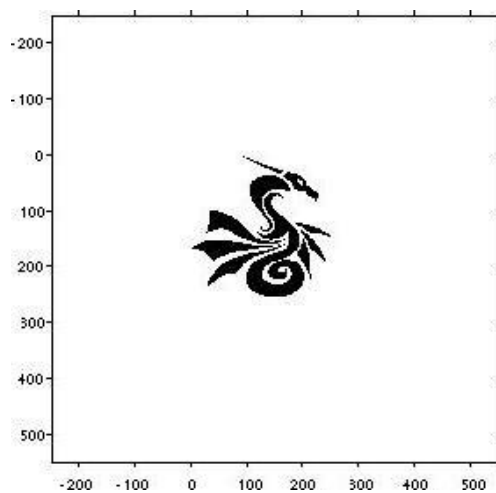
$$\mathbf{x}[k] = \mathbf{M} \cdot \mathbf{s}[k], \quad (6.1)$$

where $\mathbf{s}[k] = [s_1[k], s_2[k]]^T$ represent the vectors of the two independent, unit variance, source data, $\mathbf{x}[k] = [x_1[k], x_2[k]]^T$ represent the vectors of the two mixtures, k is the data index, and \mathbf{M} is a 2×2 mixing matrix. Given the observed mixtures $x_1[k]$ and $x_2[k]$, ICA can extract the two concealed "independent components" $s_1[k]$ and $s_2[k]$, regardless of what mixing matrix \mathbf{M} was used. If the two mixtures $x_1[k]$ and $x_2[k]$ are the pixel coordinates of all the pixels in the shape, and $\mathbf{x}[k] = [x_1[k], x_2[k]]^T$ are the pixel coordinates of all the pixels in an affine related shape, related by the transform matrix \mathbf{A} , from Equation 3.3 and Equation 6.1, we have

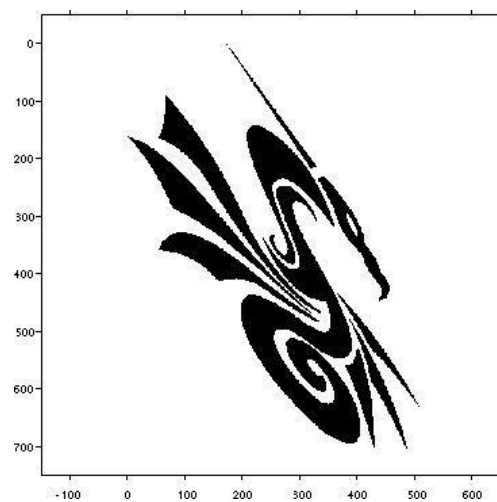
$$\mathbf{x}_a[k] = \mathbf{A} \cdot \mathbf{x}[k] = \underbrace{\mathbf{A} \cdot \mathbf{M}}_{\mathbf{M}_a} \cdot \mathbf{s}[k] = \mathbf{M}_a \cdot \mathbf{s}[k], \quad (6.2)$$

where \mathbf{M}_a is a new mixing matrix, combined of \mathbf{A} and \mathbf{M} . ICA will extract the same independent components $\mathbf{s}[k]$, whether \mathbf{M} or \mathbf{M}_a is the mixing matrix, and whether $\mathbf{x}[k]$ or $\mathbf{x}_a[k]$ are the observed mixtures. The extracted independent components are the same only in a general sense, their orders and signs cannot be determined. That is because of the order, sign and scale ambiguities of ICA [95], which are reduced to the order and sign ambiguities as the source components are fixed to have unit variance. The extracted independent components, $\mathbf{s}[k] = [s_1[k], s_2[k]]^T$, are used as the pixel coordinates of the shape in its canonical form.

Figures 6.1 (a) and (b), show the two affine related eagle shapes. Figures 6.2 (a~h) show eight possible ICA-canonicalized shapes transformed from the shapes in Figure 6.1 (a). Those eight possible ICA-canonicalized shapes are the results of different trials of ICA estimations. We can see that they are related by a $n \cdot 90$ ($n=1, 2$, or 3) degree rotation and/or flips. For example, 6.2 (a), (b), (c) and (e) are 90, 180, or 270 degree rotated version of each others. The reason for the variation is the order ambiguity of ICA, i.e., the x and y coordinates of the ICA-canonicalized shape can be switched. Similarly, because of the sign ambiguity of ICA, the ICA-canonicalized shape can be flipped against the x axis and/or y axis. For example, 6.2 (a) is an x -axis flipped version of (h). When the effects of the sign and order ambiguities combine, two ICA-canonicalized shapes transformed from the same shape are related by a rotation and flip. Because of the two ambiguities of ICA, for each shape, there are all together eight possible variations of ICA-canonicalized shapes related by a rotation and/or flips. Figures 6.3 (a~h) show eight possible ICA-canonicalized shapes transformed from the shape in Figure 6.1 (b). We can see that those ICA-canonicalized shapes are very similar to the ICA-canonicalized shapes in Figures 6.2. Each of the shape in Figures 6.3, can find a corresponding shape that is similar and in exactly the same position, and seven other shapes that are different only by a rotation and/or flips, in Figures 6.2. Figures 6.4 (a) and (b), show a pepper shape, and one of its ICA canonicalized shape, respectively. We can see that the ICA canonicalized pepper shape, is very different from the ICA canonicalized eagle shape.

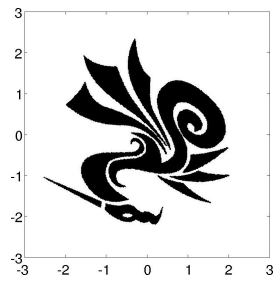


(a)Eagle-1

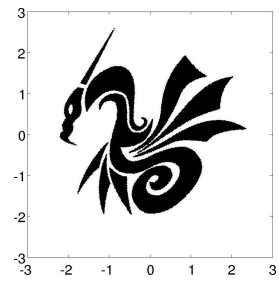


(b)Eagle-2

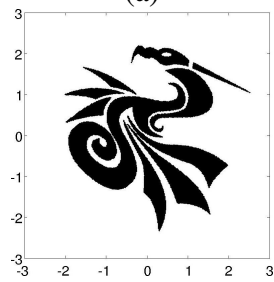
Figure 6.1: Two affine-related shapes (eagle)



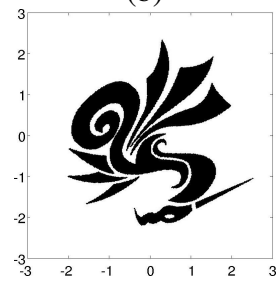
(a)



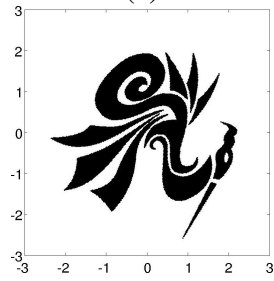
(b)



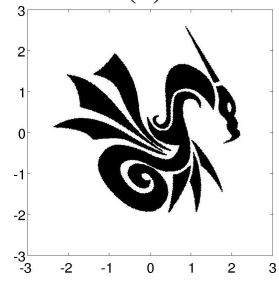
(c)



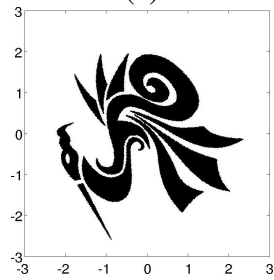
(d)



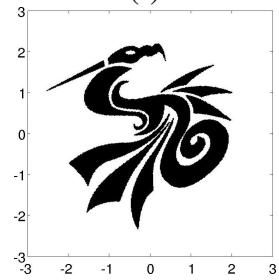
(e)



(f)

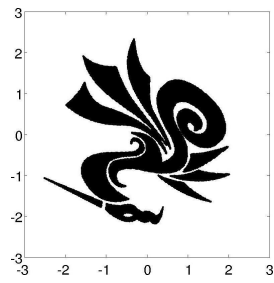


(g)

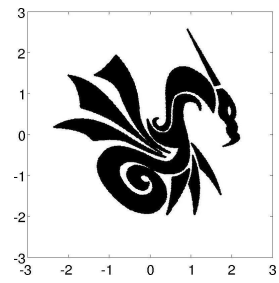


(h)

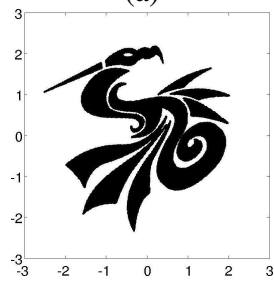
Figure 6.2: ICA canonicalized shapes corresponding to the shape in Figure 6.1 (a)



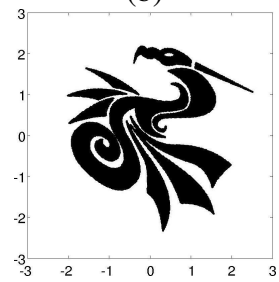
(a)



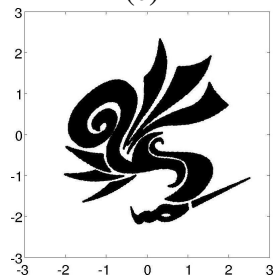
(b)



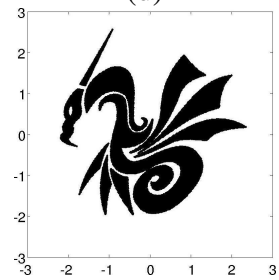
(c)



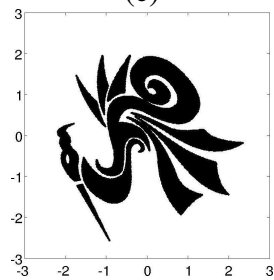
(d)



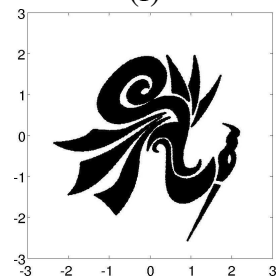
(e)



(f)



(g)



(h)

Figure 6.3: ICA canonicalized shapes corresponding to the shape in Figure 6.1 (b)

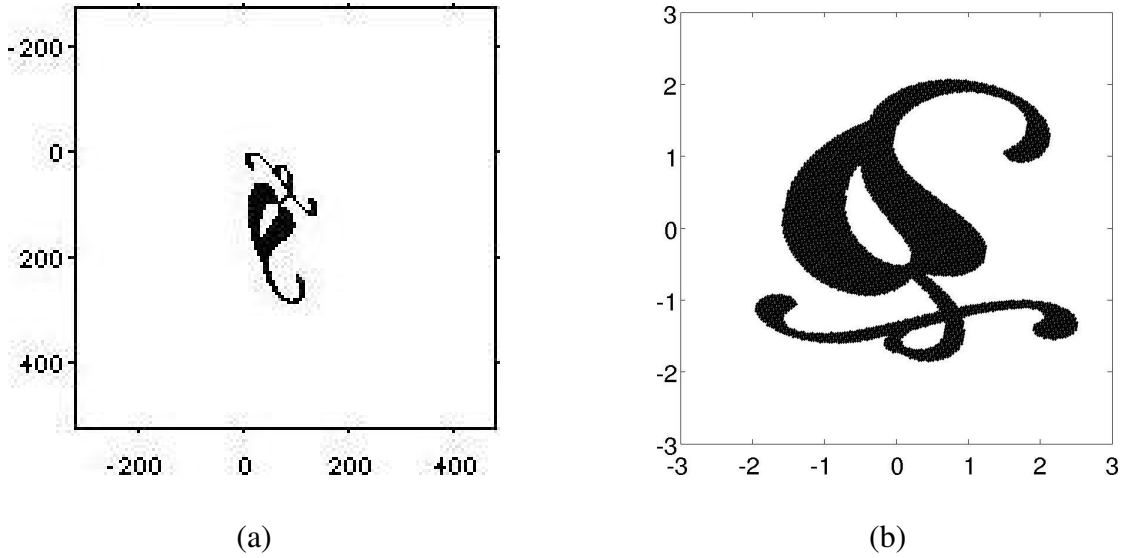


Figure 6.4: (a) A pepper shape, and (b) its ICA-canonicalized shape

6.3 ICA Zernike moment shape descriptor

While ICA can transform a shape into one of eight possible canonicalized shapes, we need a tool that can further extract shape descriptors from the canonicalized shapes. Moments have been used in image analysis and pattern recognition before [53, 54, 66, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114]. Among them, orthogonal moments are projections of an image function to a set of mutually independent basis functions. Thus, the orthogonal moment features extracted from an image function are independent of each others, and have the least amount of information redundancy. Since redundancy is undesired in pattern recognition, orthogonal moment features have advantages over other moment features.

Based on the domains of their orthogonal basis functions, orthogonal moments can be divided into two groups: moments orthogonal on a unit disk [55, 115, 116, 117, 118], such as the Zernike moments [55], and moments orthogonal on a unit square [105, 108, 113], such as the Legendre moments [105]. Moments orthogonal on a unit disk usually *have* the rotation and reflection invariant properties, while moments orthogonal on a unit square usually *do not have* these invariant properties. Because the eight possible ICA-canonicalized shapes are rotation and/or reflected versions of each others, the moments

orthogonal on a unit disk will be chosen to further extract the shape descriptor, so that there will be only a single shape descriptor to represent the original shape, regardless of which of the eight canonicalized shapes is generated by ICA.

Zernike moments [107, 109, 110, 111, 112, 119, 120] are moments defined on a unit disk. Zernike moments have been used in modeling corneal surface [121, 122], image segmentation [123], edge detection [124], image reconstruction [125], watermarking [126].

Due to its robust performance, Zernike moments have also been a popular choice in extracting regular shape descriptors [55, 56, 127, 128]. In this work, they are adopted to extract affine invariant shape descriptors.

6.3.1 Zernike moments

Zernike moments are projections of a function to the Zernike polynomials, which were first proposed in 1934 by Zernike [129]. The Zernike polynomials are a set of complex polynomials that form a complete set of orthogonal bases defined on the unit disc, i.e., $x^2 + y^2 \leq 1$. The Zernike polynomials $\{V_{nm}(x, y)\}$ have the following form:

$$V_{nm}(x, y) = V_{nm}(r, \theta) = R_{nm}(r) \exp(jm\theta), \quad (6.3)$$

where

n is a positive integer or zero that defines the order.

m is a positive or negative integer constrained by the conditions: $n - |m| = \text{even}$ and $|m| \leq n$.

x is the horizontal distance in the Cartesian coordinate system.

y is the vertical distance in the Cartesian coordinate system.

r is the length of vector from the origin, in the polar coordinate system

θ is the angle between the vector and the x axis counterclockwise.

$R_{nm}(r)$ is the radial polynomial defined as:

$$R_{nm}(r) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \cdot \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s}. \quad (6.4)$$

Note that the $R_{n,-m}(r) = R_{n,m}(r)$.

The orthogonal radial polynomials $\{R_{nm}(r)\}$ of order 0 to 10 are shown in Table 6.1. $\{R_{nm}(r)\}$ of order 0 to 4 are plotted in Figure 6.5. More complete plots of $\{R_{nm}(r)\}$ of order 0 to 10 are shown in Appendix A.

The set of Zernike polynomials (the ZM bases) $\{V_{nm}(x,y)\}$ of order $n = 0$ to 3, are shown in Figure 6.6 - Figure 6.11. More complete plots of $\{V_{nm}(x,y)\}$ of order $n = 0$ to 5, are shown in Appendix B. The Zernike polynomials $\{V_{nm}(x,y)\}$, which form orthogonal bases, satisfy

$$\iint_{x^2+y^2 \leq 1} V_{nm}(x,y) V_{pq}^*(x,y) dx dy = \frac{\pi}{n+1} \delta_{np} \delta_{mq} \quad (6.5)$$

with

$$\delta_{ab} = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases}$$

Zernike moments $\{Z_{nm}\}$ are the projection of a 2-D function to these orthogonal basis. The Zernike moment of order n and repetition m for a continuous function $f(x,y)$ defined inside the unit circle is

$$\begin{aligned} Z_{nm} &= \frac{\pi}{n+1} \iint_{x^2+y^2 \leq 1} f(x,y) V_{nm}^*(x,y) dx dy \\ &= \frac{\pi}{n+1} \int_0^{2\pi} \int_0^1 f(r,\theta) V_{nm}^*(r,\theta) r dr d\theta. \end{aligned} \quad (6.6)$$

Note that the $Z_{nm}^* = Z_{n,-m}$.

Order(n)		Number of moments in each order n
0	$R_{0,0} = 1$	1
1	$R_{1,1} = r$	1
2	$R_{2,0} = 2r^2 - 1, R_{2,2} = r^2$	2
3	$R_{3,1} = 3r^3 - 2r, R_{3,3} = r^3$	2
4	$R_{4,0} = 6r^4 - 6r^2 - 1, R_{4,2} = 4r^4 - 3r^2, R_{4,4} = r^4$	3
5	$R_{5,1} = 10r^5 - 12r^3 + 3r, R_{5,3} = 5r^5 - 4r^3, R_{5,5} = r^5$	3
6	$R_{6,0} = 2r^4 - 30r^4 + 12r^2 - 1, R_{6,2} = 15r^6 - 20r^4 + 6r^2,$ $R_{6,4} = 6r^6 - 5r^4, R_{6,6} = r^6$	4
7	$R_{7,1} = 35r^7 - 60r^5 + 30r^3 - 4r, R_{7,3} = 21r^7 - 30r^5 + 10r^3,$ $R_{7,5} = 7r^7 - 6r^5, R_{7,7} = r^7$	4
8	$R_{8,0} = 70r^8 - 140r^6 + 90r^4 - 20r^2 - 1,$ $R_{8,2} = 56r^8 - 105r^6 + 60r^4 - 10r^2,$ $R_{8,4} = 28r^8 - 42r^6 + 15r^4, R_{8,6} = 8r^8 - 7r^6, R_{8,8} = r^8$	5
9	$R_{9,1} = 126r^9 - 280r^7 + 210r^5 - 60r^3 - 5r,$ $R_{9,3} = 84r^9 - 168r^7 + 105r^5 - 20r^3,$ $R_{9,5} = 36r^9 - 56r^7 + 21r^5, R_{9,7} = 9r^7 - 8r^5, R_{9,9} = r^7$	5
10	$R_{10,0} = 252r^{10} - 630r^8 + 560r^6 - 210r^4 + 30r^2 - 1,$ $R_{10,2} = 210r^{10} - 504r^8 + 420r^6 - 140r^4 + 15r^2,$ $R_{10,4} = 120r^{10} - 252r^8 + 168r^6 - 35r^4,$ $R_{10,6} = 45r^{10} - 72r^8 + 28r^6, R_{10,8} = 10r^{10} - 9r^8, R_{10,10} = r^{10}$	6

Table 6.1: Zernike radial polynomials $R_{nm}(r)$

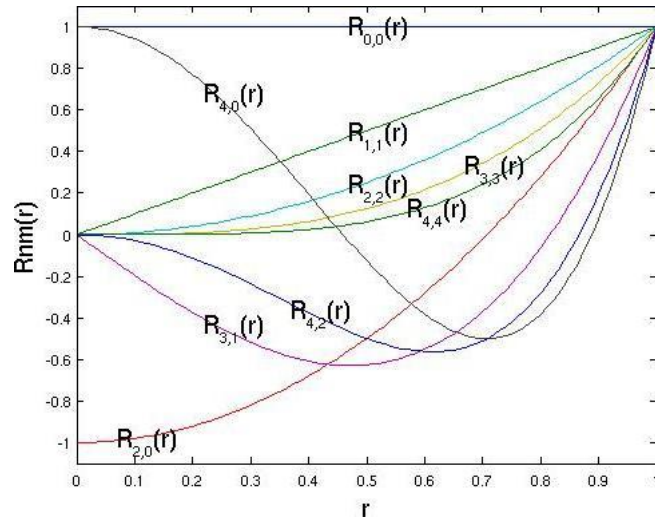


Figure 6.5: Zernike radial polynomials $R_{nm}(r)$, $(n = 0, 1, 2, 3, 4)$

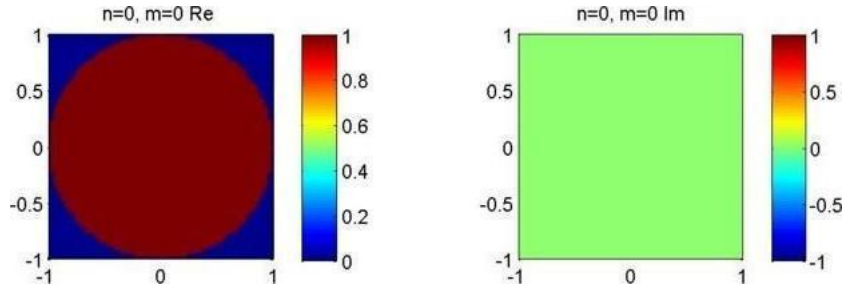


Figure 6.6: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 0, m = 0)$

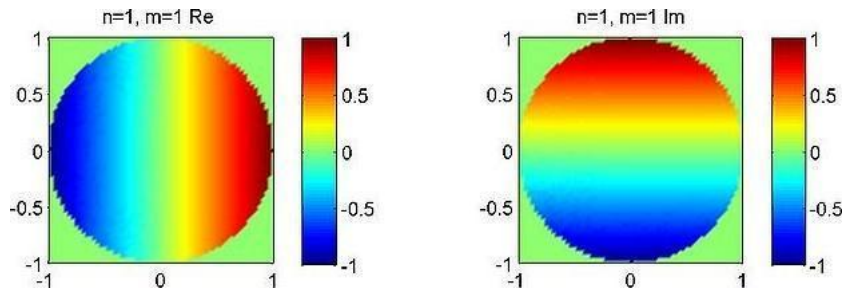


Figure 6.7: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 1, m = 1)$

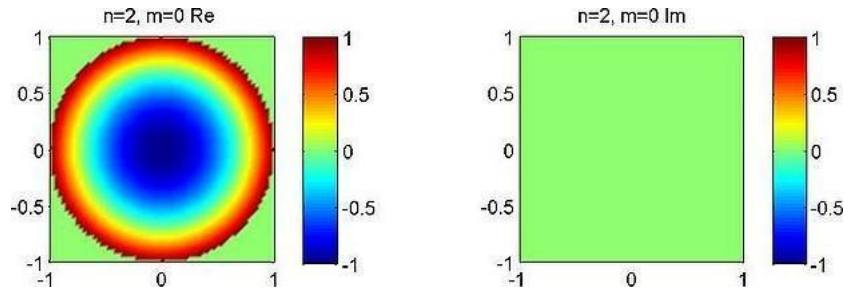


Figure 6.8: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 2, m = 0)$

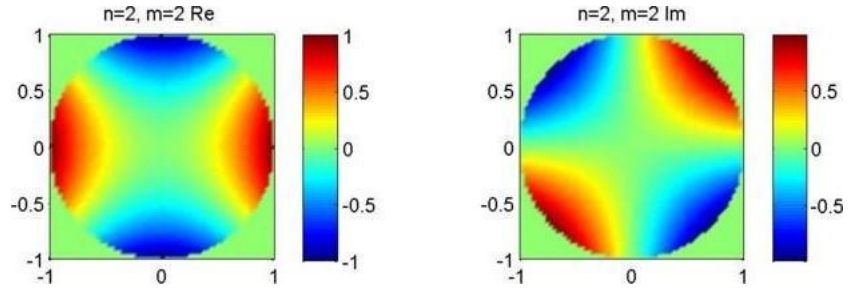


Figure 6.9: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 2, m = 2)$

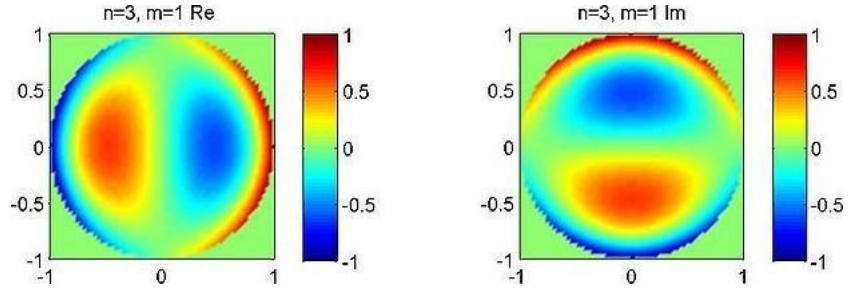


Figure 6.10: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 3, m = 1)$

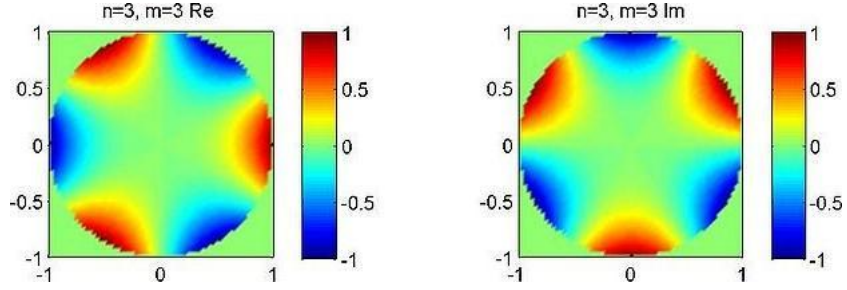


Figure 6.11: Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 3, m = 3$)

6.3.2 The invariant properties of the Zernike moments

The rotation and reflection invariant properties of the Zernike moments are utilized to extract invariant shape descriptors from ICA-canonicalized shapes that are different only by a rotation and/or reflection. Here, we prove those two invariant properties of the Zernike moments:

6.3.2.1 Rotation invariant

If the rotated image against the x axis is denoted by $f^{Rotated}$, the relationship between the rotated image and the original one in the same polar coordinates is:

$$f^{Rotated}(r, \theta) = f(r, \theta - \alpha). \quad (6.7)$$

We have

$$\begin{aligned} Z_{nm} &= \frac{\pi}{n+1} \int_0^{2\pi} \int_0^1 f(r, \theta) V_{nm}^*(r, \theta) r dr d\theta \\ &= \frac{\pi}{n+1} \int_0^{2\pi} \int_0^1 f(r, \theta) R_{nm}(r) \exp(-jm\theta) r dr d\theta, \end{aligned} \quad (6.8)$$

and

$$Z_{nm}^{Rotated} = \frac{\pi}{n+1} \int_0^{2\pi} \int_0^1 f(r, \theta - \alpha) R_{nm}(r) \exp(-jm\theta) r dr d\theta. \quad (6.9)$$

By a change of variable $\theta_1 = \theta - \alpha$, we have

$$\begin{aligned}
Z_{nm}^{Rotated} &= \frac{\pi}{n+1} \int_{-\alpha}^{2\pi-\alpha} \int_0^1 f(r, \theta_1) R_{nm}(r) \exp(-jm(\theta_1 + \alpha)) r dr d\theta_1 \\
&= \frac{\pi}{n+1} \int_0^{2\pi} \int_0^1 f(r, \theta_1) R_{nm}(r) \exp(-jm(\theta_1 + \alpha)) r dr d\theta_1 \\
&= \left[\frac{\pi}{n+1} \int_0^{2\pi} \int_0^1 f(r, \theta_1) R_{nm}(r) \exp(-jm\theta_1) r dr d\theta_1 \right] \exp(-jm\alpha) \\
&= Z_{nm} \exp(-jm\alpha),
\end{aligned} \tag{6.10}$$

and

$$|Z_{nm}^{Rotated}| = |Z_{nm}|. \tag{6.11}$$

6.3.2.2 Reflection invariant

Since a reflection against the y axis can be decomposed into a rotation, and a reflection against the x axis, we only have to prove that the Zernike moments are reflection invariant against the x axis.

If the reflected image against the x axis is denoted by f^{RF} , the relationship between the reflected image and the original one in the same polar coordinates is:

$$f^{RF}(r, \theta) = f(r, -\theta). \tag{6.12}$$

By a change of variable $\theta_1 = -\theta$, we have

$$\begin{aligned}
Z_{nm}^{RF} &= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r, -\theta) R_{nm}(\rho) \exp(-jm\theta) r dr d\theta \\
&= -\frac{n+1}{\pi} \int_0^{-2\pi} \int_0^1 f(r, \theta_1) R_{nm}(r) \exp(jm\theta_1) r dr d\theta_1,
\end{aligned} \tag{6.13}$$

and

$$\begin{aligned}
|Z_{nm}^{RF}| &= \frac{n+1}{\pi} \int_0^{-2\pi} \int_0^1 |f(r, \theta_1) R_{nm}(r) r| dr d\theta_1 \\
&= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 |f(r, \theta_1) R_{nm}(r) r| dr d\theta_1 \\
&= |Z_{nm}|.
\end{aligned} \tag{6.14}$$

6.3.3 Zernike moments extraction from the ICA-canonicalized shape

To extract the Zernike moments from the ICA-canonicalized shape, we need to first shrink the size of the shape so that most of its pixels are inside the unit disc, where the zernike moments are defined. Because ICA transforms the pixels to unit variance, most of the shape pixels fall inside the circle that is centered at the origin of the axis and has a radius of 3. That is in accordance with the "three-sigma" rule that 99.7% of the zero-mean Gaussian distributed data lies within three standard deviations of the mean [130]. So, the canonicalized shapes are shrunk to one third of their original sizes, before the Zernike moments are extracted from them. The magnitudes of the first thirty-six components of the ZMs are the proposed ICAZMSD. Its extraction diagram is shown in Figure 6.12

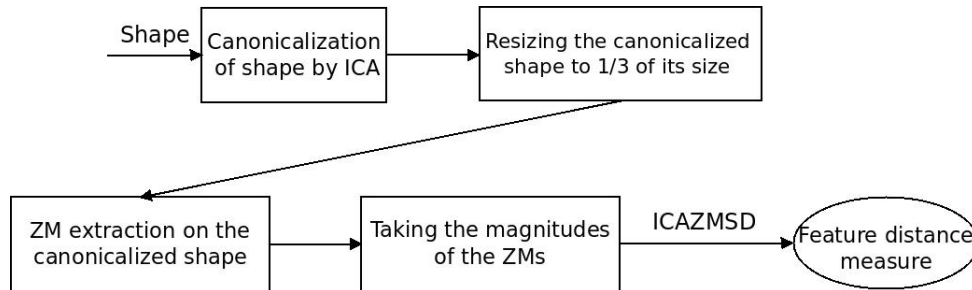
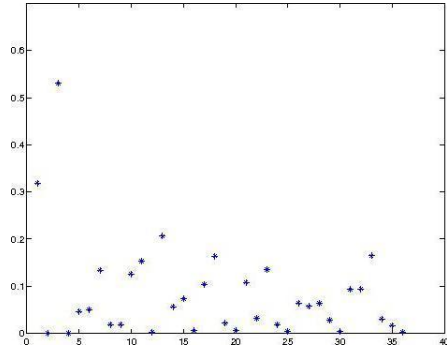
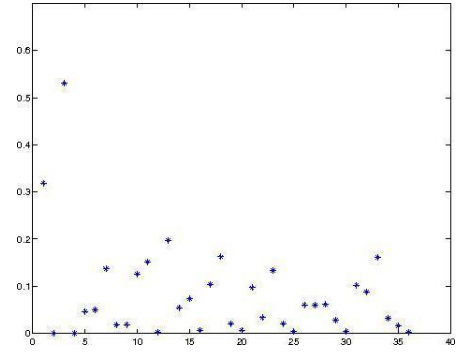


Figure 6.12: Diagram of ICAZMSD extraction

Because of the rotation and reflection invariant properties of the Zernike moments, there is only one single ICAZMSD for each shape, although each shape can have eight possible ICA-canonicalized shapes. Figure 6.13 (a) and (b) show the ICAZMSDs of the two affine related eagle shapes shown in Figures 6.1 (a) and (b), respectively.



(a)



(b)

Figure 6.13: ICAZMSDs of the two affine related eagle shapes shown in Figures 6.1

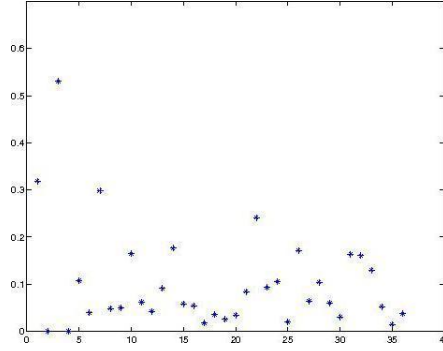


Figure 6.14: ICAZMSD of the pepper shape in Figure 6.4

We know from Section 6.2, the ICA-canonicalized shapes in Figure 6.2 are related to those in Figure 6.3, by a rotation and/or flips. Because of the rotation and reflection invariant properties of the Zernike moments, the magnitudes of their ZMs (ICAZMSDs) are also the same. We can see that the ICAZMSDs (Figures 6.13 (a) and (b)) of the two affine-related eagle shapes (Figures 6.1 (a) and (b)) are the same. On the other hand, they are different from the ICAZMSD (Figure 6.14) of the unrelated pepper shape (Figure 6.4). This can be seen more clearly, when we compare those three ICAZMSDs in the same figure (Figure 6.15). So, the ICAZMSD can be used to discriminate between affine related shapes and unrelated ones.

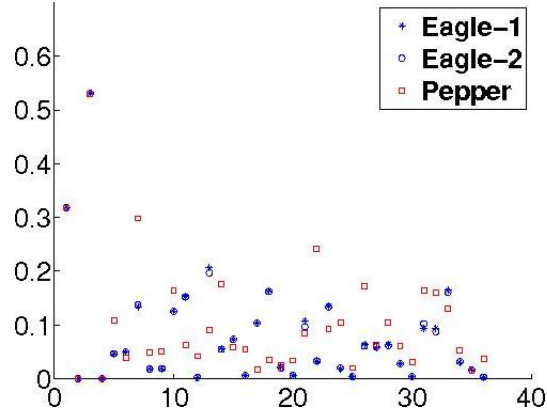


Figure 6.15: Comparison of the ICAZMSDs

6.4 ICA orthogonal Fourier-Mellin moment shape descriptor

The orthogonal Fourier-Mellin moments (OFMMs) [115, 131, 132, 133, 134] are relatively new type of moments. In [115], it is argued that the OFMMs have better image reconstruction performance than the ZMs. Because of this, the ICAOFMMSD, which utilizes both ICA and OFMMs, is proposed and investigated.

6.4.1 Orthogonal Fourier-Mellin moments

The orthogonal Fourier-Mellin moments [115] are defined as:

$$\phi_{mn} = \frac{1}{2\pi a_n} \int_0^{2\pi} \int_0^1 f(r, \theta) Q_n(r) \exp(-jm\theta) r dr d\theta, \quad (6.15)$$

where

n is a positive integer or zero that defines the order.

m is a positive or negative integers constrained by the conditions: $n - |m| = \text{even}$ and $|m| \leq n$.

r is the length of the vector from the origin, in the polar coordinate system

θ is the angle between the vector and the x axis counterclockwise.

The polynomial set of $Q_n(r)$ is defined as:

$$Q_n(r) = \sum_{s=0}^n a_{ns} r^s, \quad (6.16)$$

with

$$a_{ns} = (-1)^{n+s} \frac{(n+s+1)!}{(n-s)!s!(s+1)!}. \quad (6.17)$$

Table 6.2 shows $Q_n(r)$ from order 0 to 8.

Figure 6.16 shows the plots of $Q_n(r)$ from the order of 0 to 6.

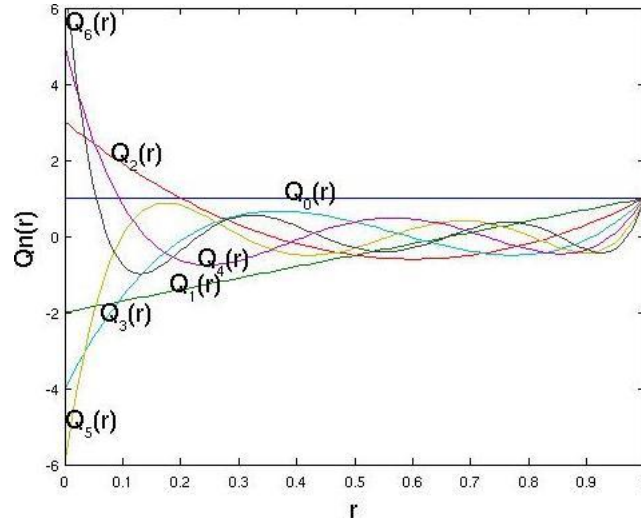


Figure 6.16: Orthogonal Mellin radial functions, n=1-6.

The set of the OFMM bases $\{P_{nm}(r, \theta) = Q_n(r) \exp(jm\theta)\}$ of order $n = 1$ to 3, are shown in Figure 6.17- Figure 6.22. A more complete plots of $\{P_{nm}(r, \theta) = Q_n(r) \exp(jm\theta)\}$ of order $n = 1$ to 5, are shown in Appendix C.

n	$Q_n(r)$	Number of moments in each order n
0	$Q_0(r) = 1$	1
1	$Q_1(r) = -2 + 3r$	1
2	$Q_2(r) = 3 - 12r + 10r^2$	2
3	$Q_3(r) = -4 + 30r - 60r^2 + 35r^3$	2
4	$Q_4(r) = 5 - 60r + 210r^2 - 280r^3 + 126r^4$	3
5	$Q_5(r) = -6 + 105r - 560r^2 + 1260r^3 - 1260r^4 + 462r^5$	3
6	$Q_6(r) = 7 - 168r + 1260r^2 - 4200r^3 + 6930r^4 - 5544r^5 + 1716r^6$	4
7	$Q_7(r) = -8 + 252r - 2520r^2 + 11550r^3 - 27720r^4 + 36036r^5 - 24024r^6 + 6435r^7$	4
8	$Q_8(r) = 9 - 360r + 4620r^2 - 27720r^3 + 90090r^4 - 168168r^5 + 180180r^6 - 102960r^7 + 24310r^8$	5

Table 6.2: The polynomial set of $Q_n(r)$

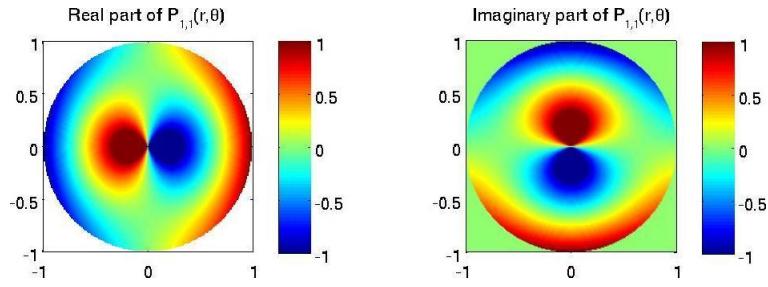


Figure 6.17: Real and imaginary parts of OFMM base, (n=1,m=1)

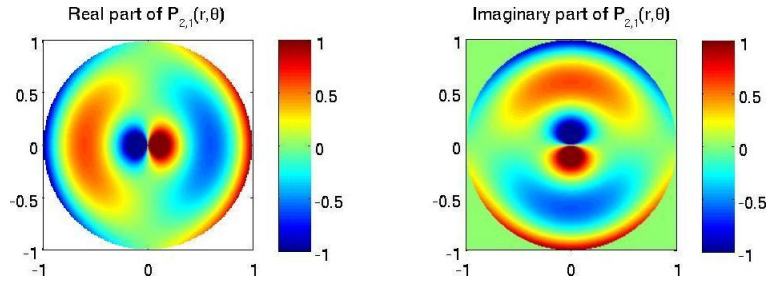


Figure 6.18: Real and imaginary parts of OFMM base, (n=2,m=1)

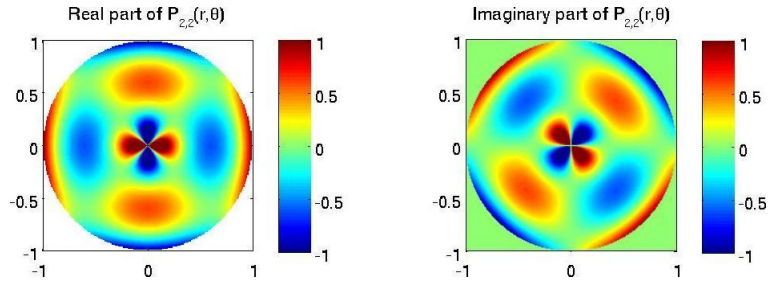


Figure 6.19: Real and imaginary parts of OFMM base, (n=2,m=2)

6.4.2 The invariant properties of the orthogonal Fourier-Mellin moments

6.4.2.1 Rotation invariant

If the rotated image against the x axis is denoted by $f^{Rotated}$, the relationship between the rotated image and the original one in the same polar coordinates is:

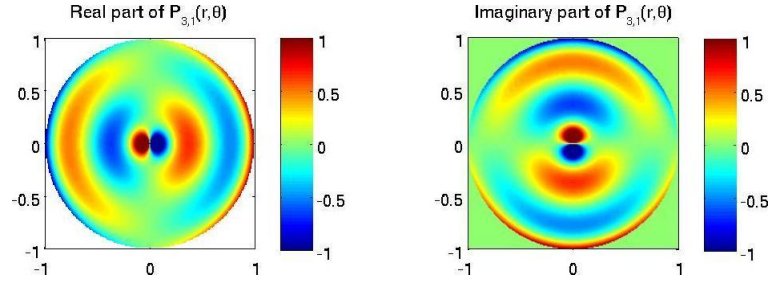


Figure 6.20: Real and imaginary parts of OFMM base, ($n=3, m=1$)

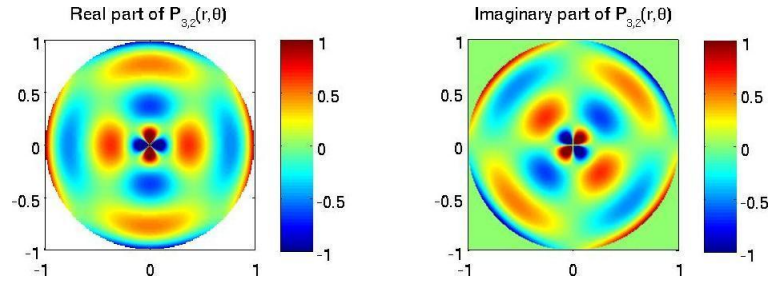


Figure 6.21: Real and imaginary parts of OFMM base, ($n=3, m=2$)

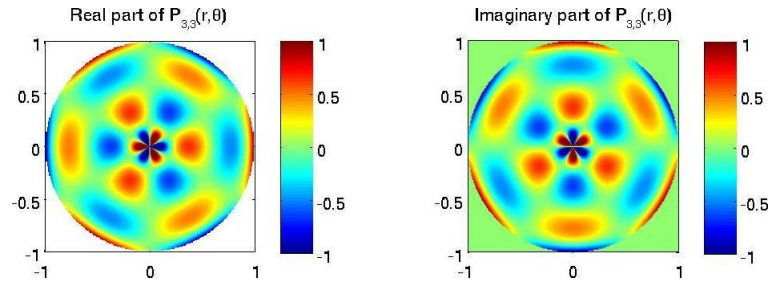


Figure 6.22: Real and imaginary parts of OFMM base, ($n=3, m=3$)

$$f^{Rotated}(r, \theta) = f(r, \theta - \beta). \quad (6.18)$$

We have

$$\phi_{mn} = \frac{1}{2\pi a_n} \int_0^{2\pi} \int_0^1 f(r, \theta) Q_n(r) \exp(-jm\theta) r \, dr d\theta, \quad (6.19)$$

and

$$\begin{aligned} \phi_{mn}^{Rotated} &= \frac{1}{2\pi a_n} \int_0^{2\pi} \int_0^1 f^{Rotated}(r, \theta) Q_n(r) \exp(-jm\theta) r \, dr d\theta \\ &= \frac{1}{2\pi a_n} \int_0^{2\pi} \int_0^1 f(r, \theta - \beta) Q_n(r) \exp(-jm\theta) r \, dr d\theta. \end{aligned} \quad (6.20)$$

By a change of variable $\theta_1 = \theta - \beta$, we have

$$\begin{aligned} \phi_{mn}^{Rotated} &= \frac{1}{2\pi a_n} \int_{-\beta}^{2\pi-\beta} \int_0^1 f(r, \theta_1) Q_n(r) \exp(-jm(\theta_1 + \beta)) r \, dr d\theta_1 \\ &= \frac{1}{2\pi a_n} \int_0^{2\pi} \int_0^1 f(r, \theta_1) Q_n(r) \exp(-jm(\theta_1 + \beta)) r \, dr d\theta_1 \end{aligned} \quad (6.21)$$

$$\begin{aligned} &= \left[\frac{1}{2\pi a_n} \int_0^{2\pi} \int_0^1 f(r, \theta_1) Q_n(r) \exp(-jm\theta_1) r \, dr d\theta_1 \right] \exp(-jm\beta) \\ &= \phi_{mn} \exp(-jm\beta), \end{aligned} \quad (6.22)$$

and

$$\left| \phi_{mn}^{Rotated} \right| = |\phi_{mn}|. \quad (6.23)$$

6.4.2.2 Reflection invariant

Here, we prove the reflection invariant property of the OFMMs, in an identical manner to the ZMs. Since a reflection against the y axis can be decomposed into a rotation, and a reflection against the x axis, we only have to prove the OFMMs are reflection invariant against the x axis.

If the reflected image against the x axis is denoted by f^{RF} , the relationship between the reflected image and the original one in the same polar coordinates is:

$$f^{RF}(r, \theta) = f(r, -\theta). \quad (6.24)$$

By a change of variable $\theta_1 = -\theta$, we have

$$\begin{aligned} \phi_{mn}^{RF} &= \frac{1}{2\pi a_n} \int_0^{2\pi} \int_0^1 f^{RF}(r, \theta) Q_n(r) \exp(-jm\theta) r dr d\theta \\ &= \frac{1}{2\pi a_n} \int_0^{2\pi} \int_0^1 f(r, -\theta) Q_n(r) \exp(-jm\theta) r dr d\theta \\ &= - \frac{1}{2\pi a_n} \int_0^{-2\pi} \int_0^1 f(r, \theta_1) Q_n(r) \exp(jm\theta_1) r dr d\theta_1, \end{aligned} \quad (6.25)$$

and

$$\begin{aligned} |\phi_{mn}^{RF}| &= \frac{1}{2\pi a_n} \int_0^{-2\pi} \int_0^1 |f(r, \theta_1) Q_n(r) r| dr d\theta_1 \\ &= \frac{1}{2\pi a_n} \int_0^{2\pi} \int_0^1 |f(r, \theta_1) Q_n(r) r| dr d\theta_1 \\ &= |\phi_{mn}|. \end{aligned} \quad (6.26)$$

6.4.3 Orthogonal Fourier-Mellin moments extraction from the ICA-canonicalized shape

Similar to extracting the ICAZMSDs, we need to first shrink the size of the shape so that most of its pixels are inside the unit disc, where the OFMMs are defined. The canonicalized shapes are first shrunk to one third of their original sizes, before the OFMMs are extracted from them. The diagram of ICAOFMMSD extraction is shown in Figure 6.23.

Because of the rotation and reflection invariant properties of the OFMMs, there is only one single ICAOFMMSD for each shape, although each shape can have eight possible ICA-canonicalized shapes. Figure 6.24 (a) and (b) show the ICAOFMMSDs of the two affine related eagle shapes shown in Figures 6.1 (a) and (b), respectively.

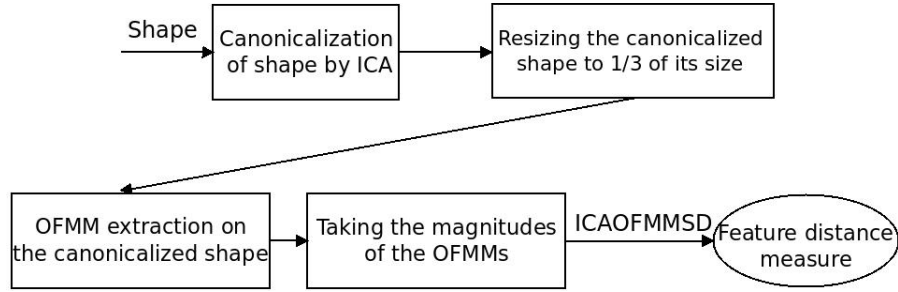


Figure 6.23: Diagram of ICAOFMMSD extraction

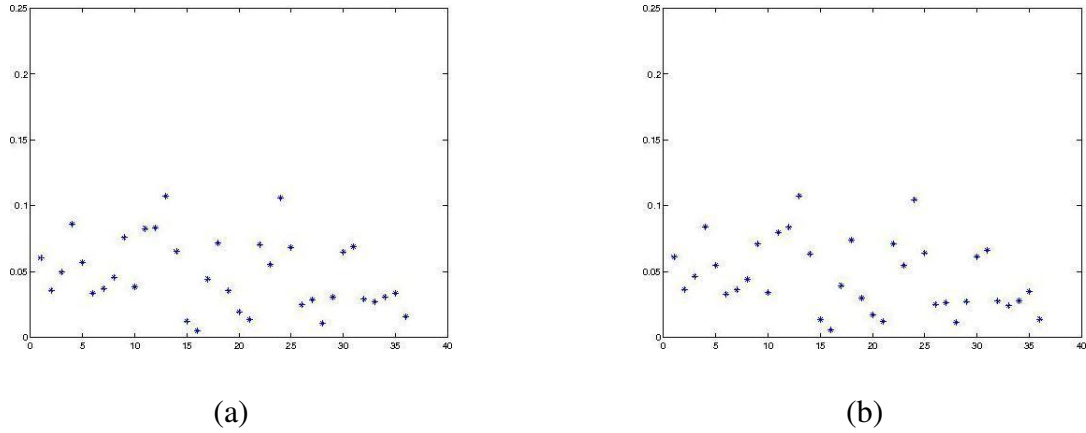


Figure 6.24: ICAOFMMSDs of the two affine related eagle shapes shown in Figures 6.1

We know from Section 6.2, the ICA-canonicalized shapes in Figure 6.2 are related to those in Figure 6.3, by a rotation and/or flips. Because of the rotation and reflection invariant properties of the OFMMs, the magnitudes of their OFMMs (ICAOFMMSDs) are also the same. We can see that the ICAOFMMSDs (Figures 6.24 (a) and (b)) of the two affine-related eagle shapes (Figures 6.1 (a) and (b)) are the same. On the other hand, they are different from the ICAOFMMSD (Figure 6.25) of the unrelated pepper shape (Figure 6.4). This can be seen more clearly, when we compare those three ICAOFMMSDs in the same figure (Figure 6.26). So, the ICAOFMMSD can be used to discriminate between affine related shapes and unrelated ones.

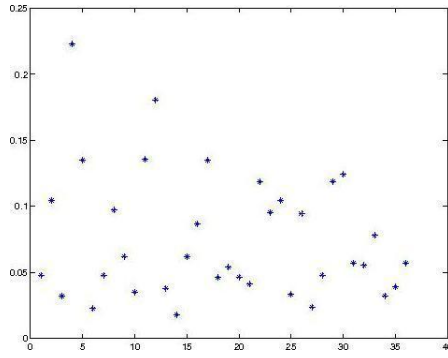


Figure 6.25: ICAOFMMSD of the pepper shape in Figure 6.4

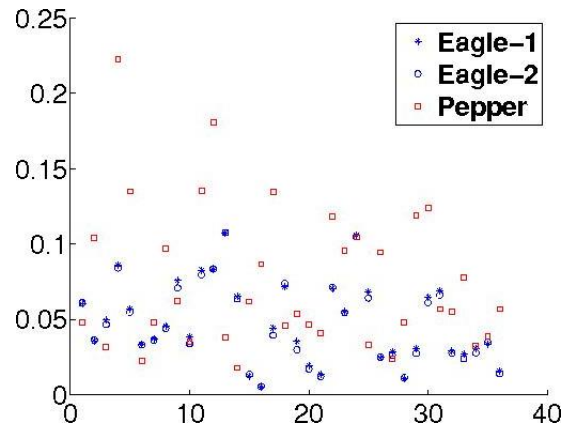


Figure 6.26: Comparison of the ICAOFMMSDs

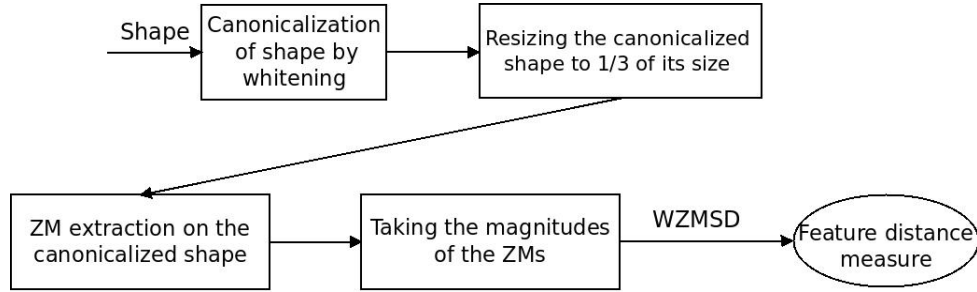


Figure 6.27: Diagram of WZMSD extraction

6.5 Whitening Zernike moment shape descriptor and whitening orthogonal Fourier-Mellin moment shape descriptor

As we have reviewed from Chapter 4, the difference between the whitened variables and the final estimated independent components are either only a rotation or a rotation followed by a flip. Given the relationship between whitening and ICA, we further proposed the whitening Zernike moment shape descriptor (WZMSD) and the Whitening orthogonal Fourier-Mellin moment shape descriptor (WOFMMSD). Their extraction steps are shown in Figure 6.27 and Figure 6.28, respectively. The only difference between them and their ICA-based counterparts is that only whitening will be performed on the shape pixels and the whitened data are not further rotated and/or flipped.

Figure 6.29 compares the WZMSDs and the WOFMMSDs of the two affine related eagle shapes and the unrelated pepper shape. We see that the shape descriptors of the two affine related eagle shapes looks the same while they are different from the shape descriptors of the unrelated pepper shape. Comparing Figures 6.13, 6.14, 6.24, 6.25, and 6.29, we found that the WZMSD and the WOFMMSD are almost the same to their ICA based counterparts. The proposed WZMSD and WOFMMSD avoid the calculation of the rotation, which is the most computationally costly step in ICA, while still having the similar performance as its ICA based counterparts.

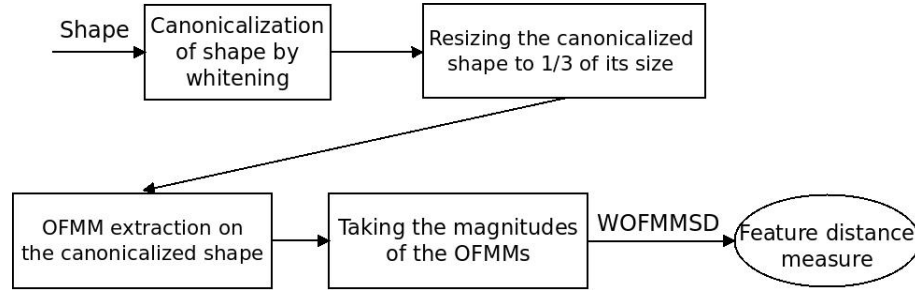


Figure 6.28: Diagram of WOFMMSD extraction

6.6 Experimental results

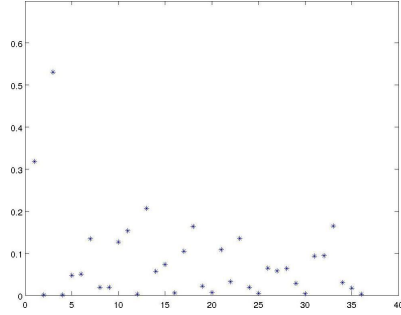
The four newly developed affine invariant shape descriptors are tested and compared with those previously known affine invariant shape representations, studied before. The AIWSR and the AMI-TC are not included in the comparison of retrieval accuracy, because of their poor retrieval performances.

6.6.1 Experimental data

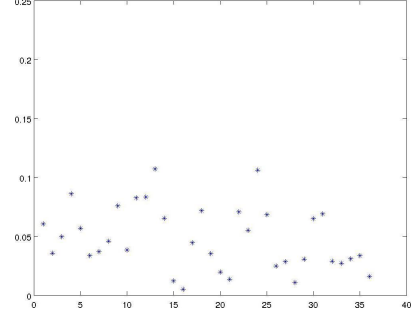
Both the simple and the complex shape database seen in Chapter 3 was used in the retrieval experiments.

6.6.2 Retrieval accuracy

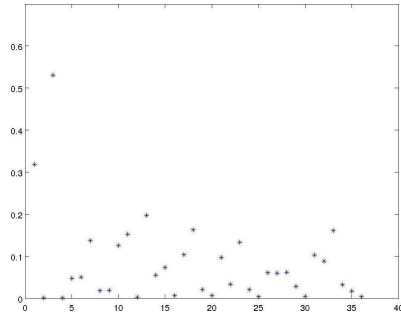
From Figure 6.30, we see that the average precision-recall curves of the ICAZMSD and those of the ICAOFMMSD are much closer to the up-right corner of the average precision-recall graph than those of the AMI-FS are. That means, the ICAZMSD and the ICAOFMMSD perform far better than the AMI-FS on complex shape database. From Figure 6.31, we see that the ICAZMSD and the ICAOFMMSD also perform far better than any other existing affine invariant shape representations on simple shape database. The average precision-recall curves of those two new shape descriptors are almost horizontal lines as the average precision rate nearly equals one hundred percent. That means that their retrieval performances are almost perfect on shape databases without noise.



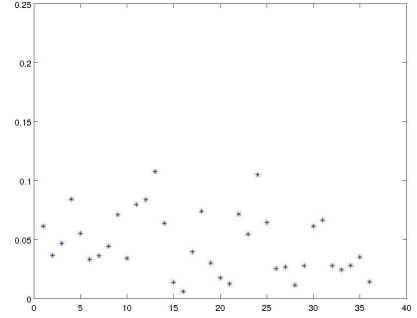
(a)WZMSD of the eagle shape in Figure 6.1 (a)



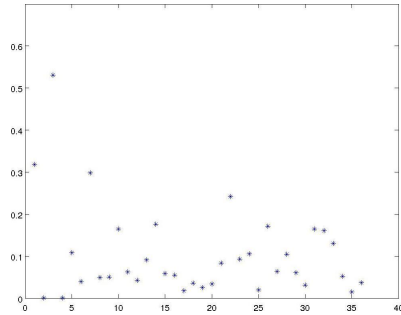
(b)WOFMMSD of the eagle shape in Figure 6.1 (a)



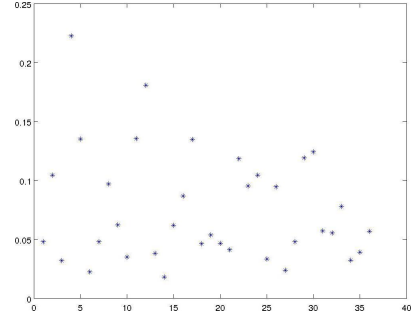
(c)WZMSD of the eagle shape in Figure 6.1 (b)



(d)WOFMMSD of the eagle shape in Figure 6.1 (b)



(e)WZMSD of the pepper shape in Figure 6.4



(f)WOFMMSD of the pepper shape in Figure 6.4

Figure 6.29: (a)(b):WZMSD and WOFMMSD of the eagle shape in Figure 6.1 (a); (c)(d):WZMSD and WOFMMSD of the eagle shape in Figure 6.1 (b); (e)(f):WZMSD and WOFMMSD of the pepper shape in Figure 6.4.

The robustness of the ICAZMSD and the ICAOFMMSD are even more apparent, when the retrieval experiments are done on simple shape database with added boundary noise (Figures 6.32 - 6.37) . Comparing Figure 6.31 with Figures 6.32 - 6.37, we see that the performances of the two newly developed affine-invariant shape descriptors degrade much more slowly than those of the existing ones, when noise is added to the shape boundary. The ICAZMSD and the ICAOFMMSD are very robust on shape databases with or without added boundary noise.

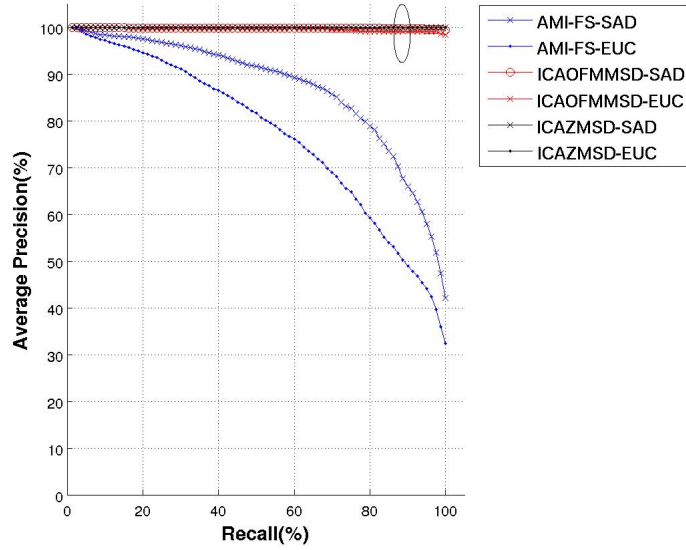


Figure 6.30: Average precision-recall graphs of 4000 retrievals using different affine invariant shape descriptors on complex shape database.

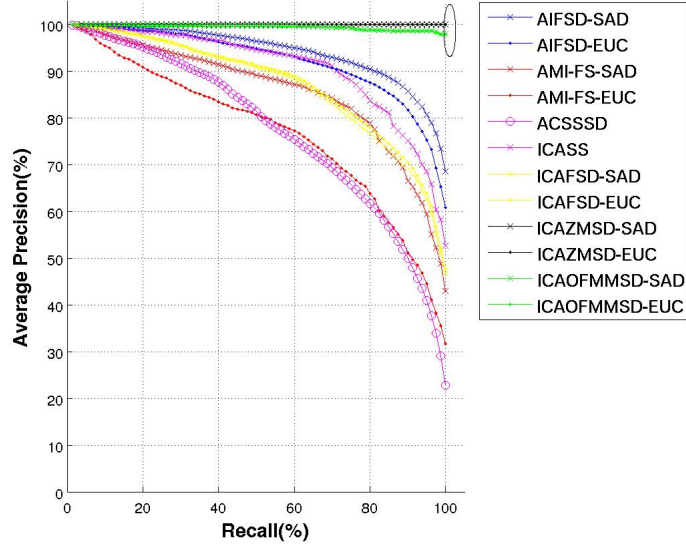


Figure 6.31: Average precision-recall graphs of 5600 retrievals using different affine invariant shape descriptors on simple shape database with no noise.

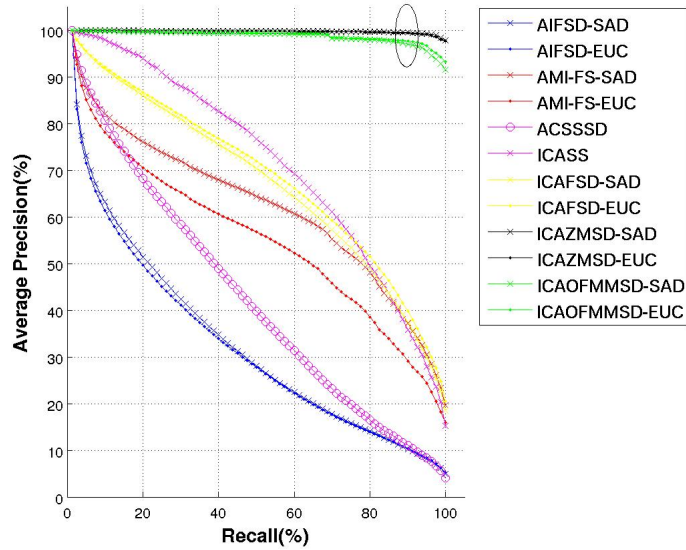


Figure 6.32: Average precision-recall graphs of 5600 retrievals using different affine invariant shape descriptors on simple shape database with Gaussian noise at $SNR_{dB}=30dB$.

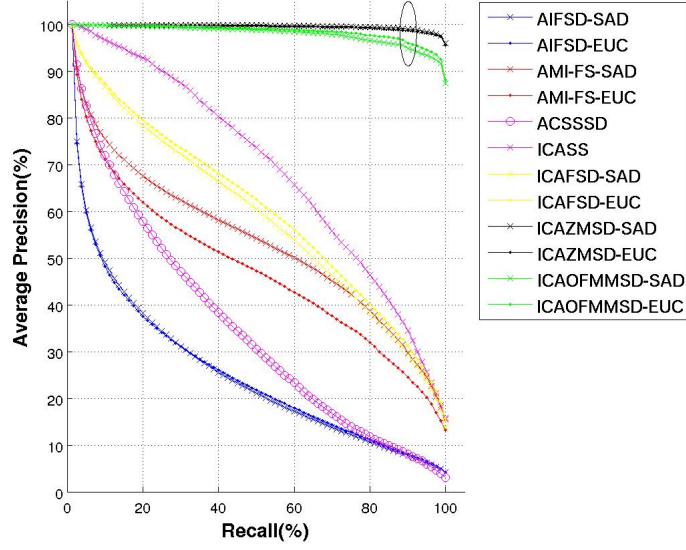


Figure 6.33: Average precision-recall graphs of 5600 retrievals using different affine invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$.

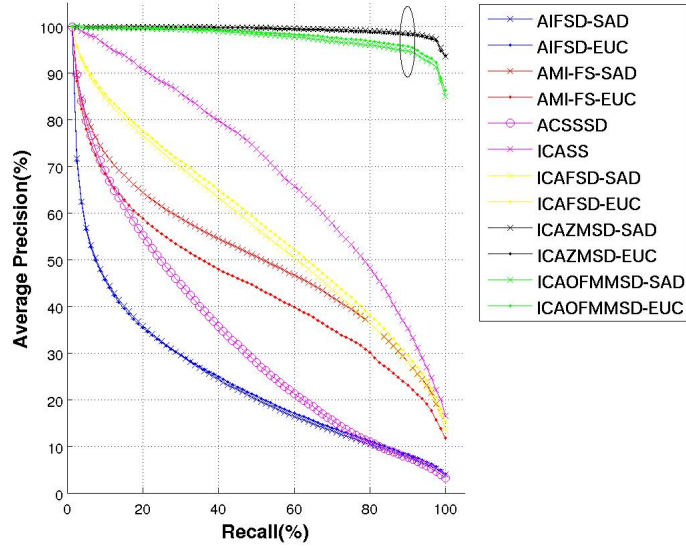


Figure 6.34: Average precision-recall graphs of 5600 retrievals using different affine invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.0206\text{dB}$.

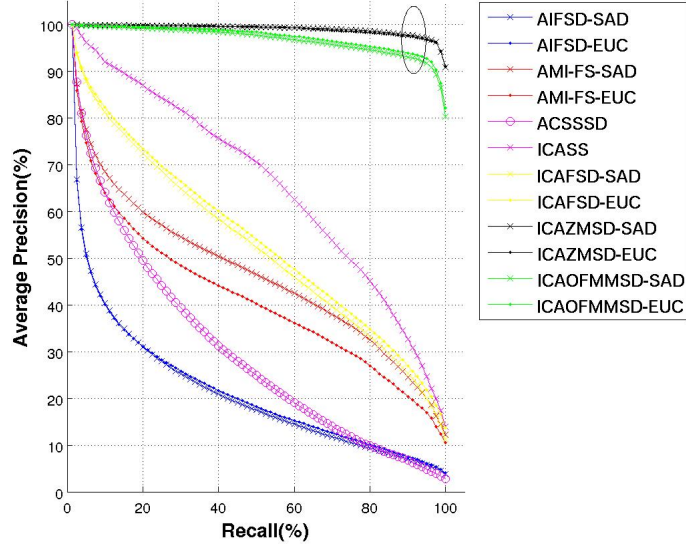


Figure 6.35: Average precision-recall graphs of 5600 retrievals using different affine invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=24.7712\text{dB}$.

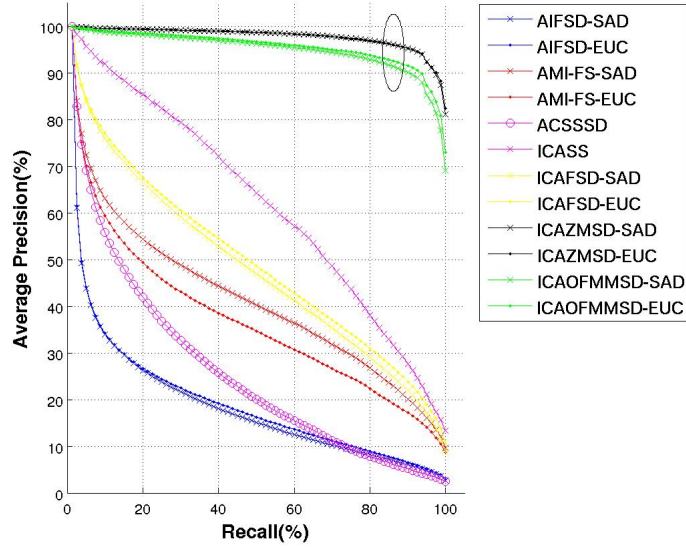


Figure 6.36: Average precision-recall graphs of 5600 retrievals using different affine invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=23.0103\text{dB}$.

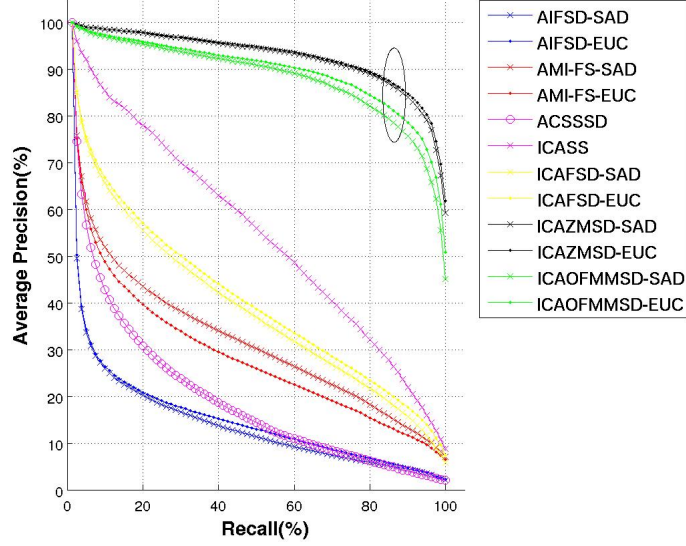


Figure 6.37: Average precision-recall graphs of 5600 retrievals using different affine invariant shape descriptors on simple shape database with Gaussian noise at $\text{SNR}_{dB}=20\text{dB}$.

The precision-recall curves of the two whitening-based shape descriptors, the WZMSD and the WOFMMSD, are not included in the above precision-recall graphs, as they will overlap with those of their ICA-based counterparts. Figure 6.38 shows the average precision-recall curve of the ICAZMSD and that of the WZMSD, tested on the same shape database. The two average precision-recall curves overlap each others, as we expected. Similarly, Figure 6.39 shows the average precision-recall curve of the ICAZMSD and that of the WZMSD, tested on the same shape database. The two average precision-recall curves also overlap each others, as we expected.

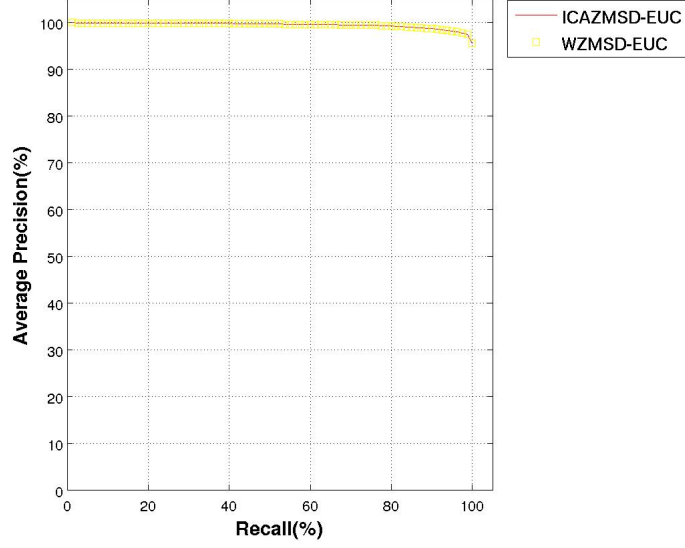


Figure 6.38: Overlapping of the average precision-recall curve of the ICAZMSD and that of the WZMSD (on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$).

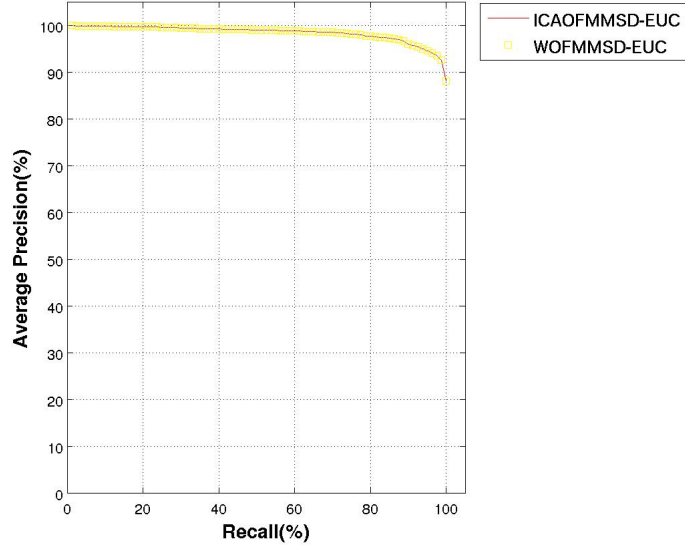


Figure 6.39: Overlapping of the average precision-recall curve of the ICAOFMMSD and that of the WOFMMSD (on the 5600 shape simple shape database with Gaussian noise at $\text{SNR}_{dB}=26.9897\text{dB}$).

6.6.3 Comparison of extraction time, distance calculation time, and compactness

The extraction time, distance calculation time, and compactness of the affine invariant shape representations are compared in Tables 6.3 - 6.5 and in Figures 6.40 - 6.42.

From Table 6.3 and Figure 6.40, we see that the four newly developed shape descriptors require a little bit more extraction time than previously known ones. From Table 6.4 and Figure 6.41, we see that the four newly developed shape descriptors require little distance calculation time. From Table 6.5 and and Figure 6.40 show their feature sizes are compact.

Shape representation	Extraction time(s)	Shape representation	Extraction time(s)	Shape representation	Extraction time(s)
AIFSD	0.0136	AICSSSD	13.9972	AMI-FS	0.1835
AIWSR	0.1377	AMI-TC	0.0521	ICASS	0.0178
ICAFSD	0.1390	<i>ICAZMSD</i>	<i>0.7976</i>	<i>ICAOFMMSD</i>	<i>2.0178</i>
WFSD	0.0164	<i>WZMSD</i>	<i>0.5416</i>	<i>WOFMMSD</i>	<i>1.8155</i>

Table 6.3: Feature extraction time(seconds)

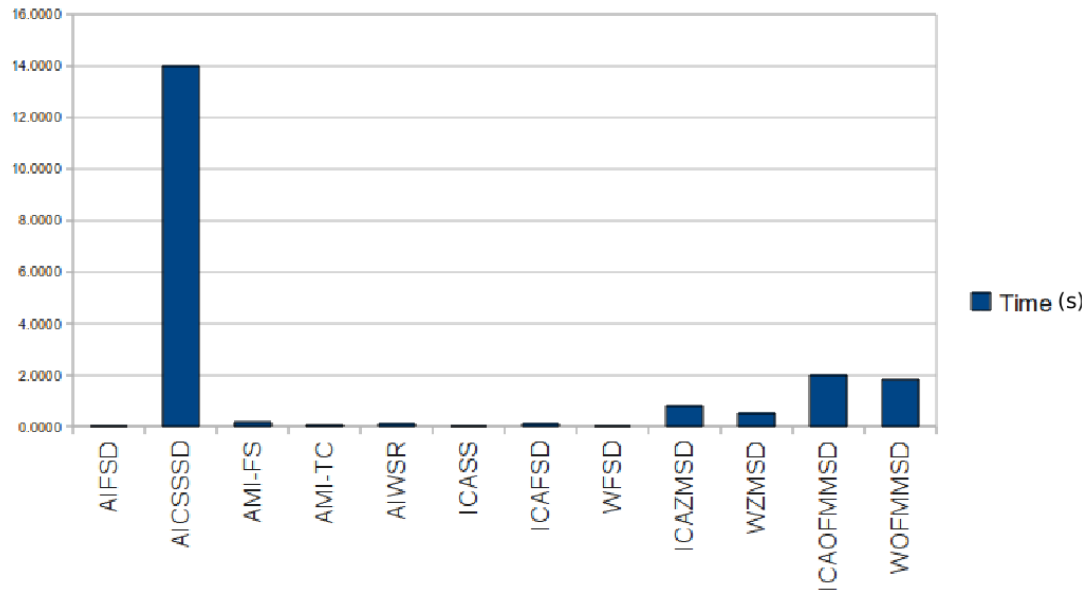


Figure 6.40: Comparison of feature extraction time (seconds)

Shape representation	Extraction time(ms)	Shape representation	Extraction time(ms)
AMI-TC-SAD	0.021	AMI-TC-EUC	0.006
AMI-FS-SAD	0.013	AMI-FS-EUC	0.005
AIFSD-SAD	0.009	AIFSD-EUC	0.006
ICAFSD-SAD	0.009	ICAFSD-EUC	0.009
WFSD-SAD	0.009	WFSD-EUC	0.009
<i>ICAZMSD-SAD</i>	<i>0.013</i>	<i>ICAZMSD-EUC</i>	<i>0.008</i>
<i>ICAOFMMSD-SAD</i>	<i>0.010</i>	<i>ICAOFMMSD-EUC</i>	<i>0.008</i>
<i>WZMSD-SAD</i>	<i>0.009</i>	<i>WZMSD-EUC</i>	<i>0.008</i>
<i>WOFMMSD-SAD</i>	<i>0.009</i>	<i>WOFMMSD-EUC</i>	<i>0.008</i>
AICSSSD	18.999	ICASS	148.886
AIWSR	120.478		

Table 6.4: Feature distance calculation time(milliseconds)

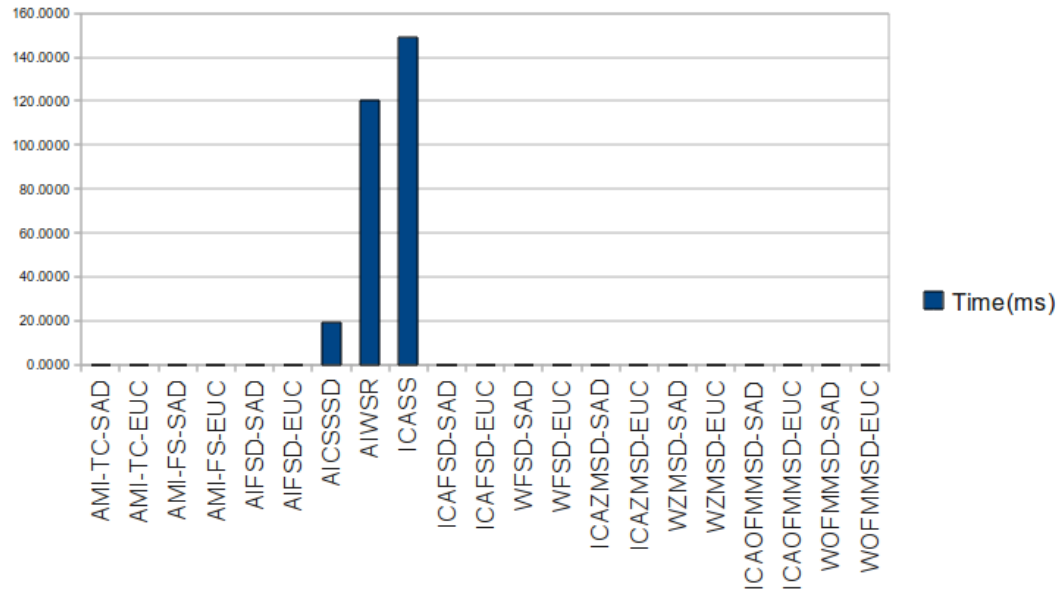


Figure 6.41: Comparison of feature distance calculation time (milliseconds)

Shape representation	Size	Shape representation	Size	Shape representation	Size
AIFSD	36	AICSSSD	(average) 5	AMI-FS	4
AIWSR	512	AMI-TC	8	ICASS	512
ICAFSD	8	<i>ICAZMSD</i>	<i>32</i>	<i>ICAOFMMSD</i>	<i>32</i>
WFSD	8	<i>WZMSD</i>	<i>32</i>	<i>WOFMMSD</i>	<i>32</i>

Table 6.5: Size of the shape representations

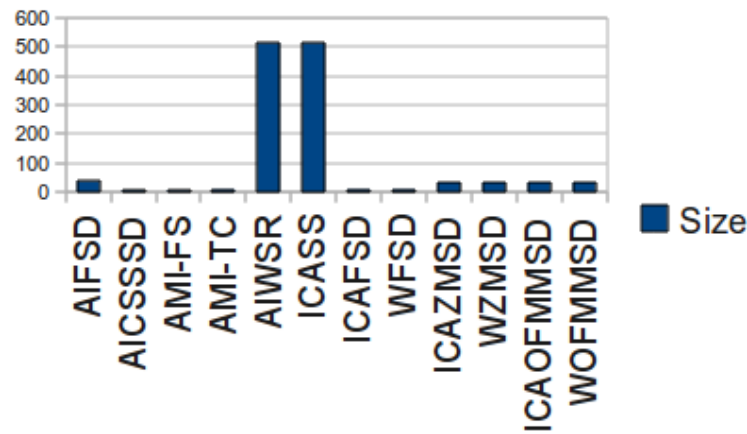


Figure 6.42: Comparison of the size of the shape representations

6.7 Application in traffic sign retrieval

In this experiment, six pictures of a “Stop” sign taken from different viewpoints and distances, and another thirty-one pictures of other traffic signs were used as test database for shape-based retrieval. The shapes of the traffic signs were then extracted using the SIOX algorithm [135] and their ICAZMSDs were further extracted. As we can observe from Figure 6.43, although the shapes of the “Stop” sign taken from different viewing angles and distances are different, their corresponding ICAZMSDs are almost the same. At the same time, they are different from the ICAZMSD extracted from the “No Left Turn” sign. Differences were also observed in comparison with the ICAZMSDs of other traffic signs

in our experiment. Using any one of the six pictures of the “Stop” sign, taken from different viewpoints and distances, the ICAZMSD-based retrieval system was able to retrieve all the other five related pictures in the top five matches without any error from the picture database. Figure 6.44 shows the retrieval results, using Figure 6.43(a) as the query image. All the five “Stop” sign pictures are in the top matches. Using the WZMSD, the ICAOFMMSD, or the WOFMMSD as shape feature descriptor, all the related pictures were also retrieved in the top matches without any error.

6.8 Summary

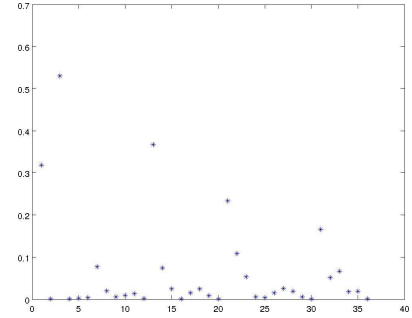
In this chapter, four newly developed region-based affine invariant shape descriptors, the ICAZMSD, the ICAOFMMSD, the WZMSD, and the WOFMMSD, have been introduced. Either ICA or whitening is used in transforming shapes into canonicalized shapes, before ZM or OFMM is applied to them to extract the descriptors. The newly developed shape descriptors show far better performance than any existing ones and can be applied to both simple and complex shapes. Their retrieval performance on shape database without noise, are close to perfect. The robustness of those affine-invariant shape descriptors are also apparent, when the retrieval experiments are done on simple shape database with added boundary noise, as their performances degrade much more slowly than those of the existing ones. Those affine-invariant shape descriptors have also been applied successfully in retrieving photos of traffic signs taken from different viewpoints and distances. Those descriptors have compact sizes and acceptable computational time requirements. The whitening based descriptors require less computational time than their ICA-based counterparts while have the same robust retrieval performance.



(a1)



(a2)



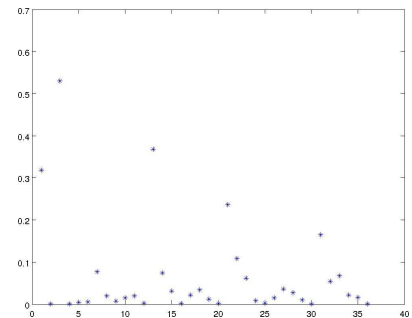
(a3)



(b1)



(b2)



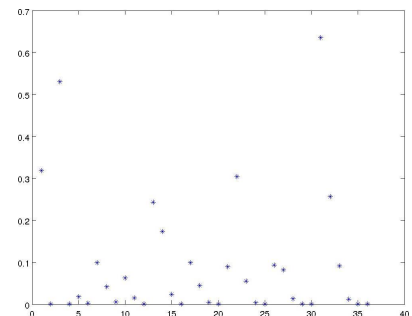
(a3)



(c1)



(c2)



(a3)

Figure 6.43: (a)Traffic signs, (b)their shape images, and (c)their ICAZMSDs

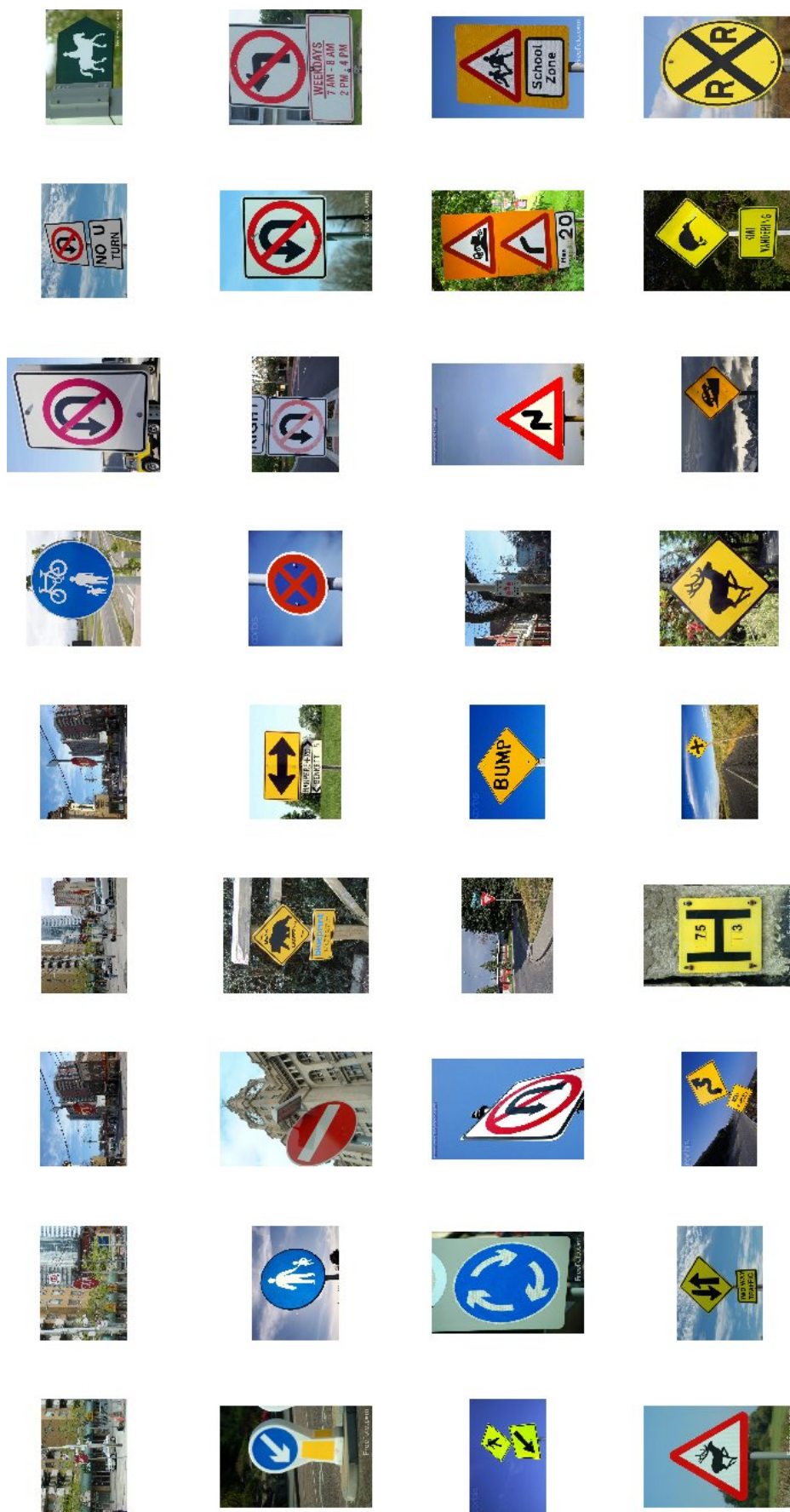


Figure 6.44: Retrieved traffic sign pictures, from the most related to the least related (left to right, top to bottom). The five pictures with 'Stop' signs are all in the top five matches.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this work, existing shape representations have been reviewed and existing affine invariant shape representations have been studied and compared using retrieval experiments on simple and complex shape databases. Experimental results show that the performance of the existing affine invariant shape representations is not satisfactory and has room to improve.

The goal of this work was to find robust affine invariant shape representations that have high retrieval accuracy, compact size, and low computational requirement. To achieve that goal, two contour-based and four region-based affine invariant shape descriptors have been developed. The two contour-based shape descriptors, the ICAFSD and the WFSD, can be applied to simple shapes. They out perform most of the existing affine invariant shape representations, have compact sizes and require low computational time. The four region-based affine invariant shape descriptors, the ICZMSD, the WZMSD, the ICAOFMMSD, and the WOFMMSD, can be applied to both simple and complex shapes, which means less restrictions on their applications. They perform far better than the existing affine invariant shape representations, have compact sizes and acceptable computational time requirements.

The contributions of this dissertation are summarized as below:

- Two contour-based affine-invariant shape descriptors are presented:

1. *ICAFSD*: In extracting the ICAFSD, ICA is used to transform a shape contour into one of eight possible canonical shape contours, which are different only by a rotation of 90, 180, or 270 degrees, and/or a reflection. The DFT is then applied on the centroid distance of the canonical shape contour. Since a rotation or reflection of the canonical shape contour will not change the magnitudes of the DFT coefficients, they are used as the newly proposed ICAFSD. The ICAFSD has a compact size, and low computational time requirement, and out performs most of the existing affine-invariant shape descriptors in retrieval experiments. Unlike some of the existing contour-based shape representations, which is vulnerable to the shape boundary noise, the ICAFSD is much robust under noisy condition.
 2. *WFSD*: Exploiting the relationship between the whitened and the ICA-ed data, the WFSD was further proposed. It also first transforms a shape contour into its canonical one, but doesn't further estimate that rotation and/or the reflection of the shape contour, as the DFT will anyway be used later to extract rotation and reflection invariant features. Therefore, the WFSD has even lower computational time requirement than the ICAFSD, while maintaining the same compactness and retrieval accuracy as the ICAFSD.
- Four region-based shape descriptors are presented:
1. *ICAZMSD*: Unlike contour-based shape descriptors, which can only be applied to simple shapes, the region-based ICAZMSD has no such limitation. ICA is applied to the coordinates of the shape pixels, instead of the coordinates of the shape contour, to transform the shape into a canonical shape. ZMs are then extracted from the canonical shape, and the magnitudes of the ZMs are the newly proposed ICAZMSD. Because the magnitudes of the ZMs are rotation and reflection invariant, the value of the ICAZMSD will be the same, regardless which of the eight possible positions the canonical shape takes. The ICAZMSD has a compact size, and acceptable computational time requirement. It performs far better than the existing affine-invariant shape descriptors and is very robust even under noisy condition.
 2. *WZMSD*: Exploiting the relationship between the whitened and the ICA-ed data, and the invariant properties of the ZMs, the WFSD was further proposed. The WZMSD

uses whitening to transform a shape into its canonical shape and extracts the magnitudes of the ZMs from the canonical shape, as the affine-invariant shape descriptor. As whitening is used, instead of ICA, the WZMSD has lower computational time requirement than the ICAZMSD, but maintains the same compactness and high retrieval accuracy as the ICAZMSD.

3. *ICAOFMMSD*: The OFMMs are relatively new type of orthogonal moments, which has been found to have better performance than the ZMs in image reconstruction. In this work, the OFMMs are used together with ICA to develop the ICAOFMMSD. The magnitudes of the OFMMs are extracted from the ICA-canonicalized shape as the proposed ICAOFMMSD. The ICAOFMMSD, can be applied to both simple shape and complex shapes. It has a compact size and acceptable computational time requirement. Its retrieval perform is far better than those of the existing affine-invariant shape descriptors.
4. *WOFMMSD*: It has similar feature extraction steps as the ICAOFMMSD. But, whitening, instead of ICA, is used to transform shapes into their canonical shapes. The WOFMMSD requires less computational time than the ICAOFMMSD, while still having the same compactness and robust retrieval performance as the ICAOFMMSD.

The six proposed affine invariant shape descriptors have shown very good retrieval performance. Among them, the four proposed region-based ones perform especially well. Their performances are close to perfect in retrieval experiments on both simple shape and complex shape databases without boundary noise. Their performances also do not deteriorate as much as those of the existing affine invariant shape representations, when they are tested on shapes with boundary noise.

7.2 Future work

The goal of developing robust affine invariant shape representations have been met. The four proposed region-based shape descriptors, which have close to perfect retrieval performance, are especially successful. At the same time, they also provide a way to describe

and link two sets of affine-related *two* dimensional data. It would be interesting to extend the methods to describe two sets of affine-related *three* dimensional data. For example, colour under illumination change (Figure 7.1) can be modeled as three dimensional affine transform [136, 137].

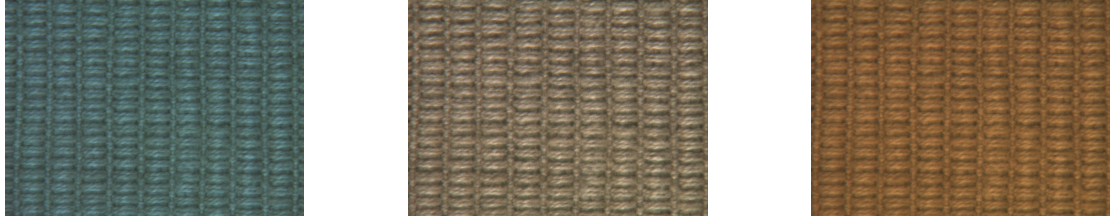


Figure 7.1: Images of a colour texture under different illumination condition (Image source: Outex colour texture database [2])

So, the idea of orthogonal spherical-harmonics-mellin moments (OSHMM) has also been investigated in my research. The OFMM can be decomposed into two parts: the radial portion $R(r)$ and the angular portion $e^{im\theta}$. We know from mathematics, that the angular portion $e^{im\theta}$ is also the angular portion of the solution in solving Laplace's equation in two dimensions [138]. The angular portion of the solution in solving Laplace's equation in three dimensions is the spherical harmonics, which can be used to describe a function define on a sphere (Figure 7.2) [138]. Given the relationship between $e^{im\theta}$ and the spherical harmonics, the *three* dimensional OSHMMs are extended from the *two* dimensional OFMMs by substituting $e^{im\theta}$ with the spherical harmonics. The OSHMMs are three dimensional moments that can be used to describe a function defined in a unit ball. Figure 7.3 shows one of the OSHMM bases. The proposed OSHMMs also have their rotation and reflection invariant properties as their two dimensional counterpart, the OFMM. They can be used, together with ICA or whitening in three dimensions, to extract affine invariant descriptors of three dimensional data. In the future, more investigation could be done to exploit the use of OSHMM and ICA in solving affine invariant problems in 3D.

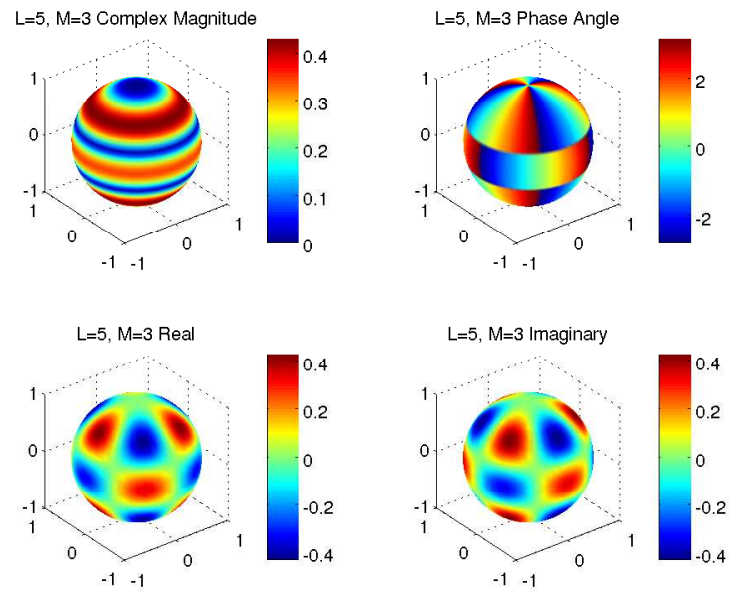
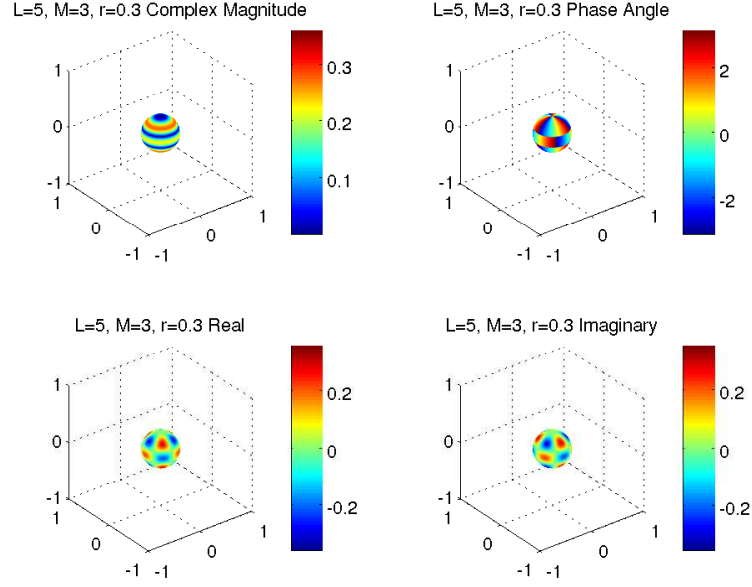
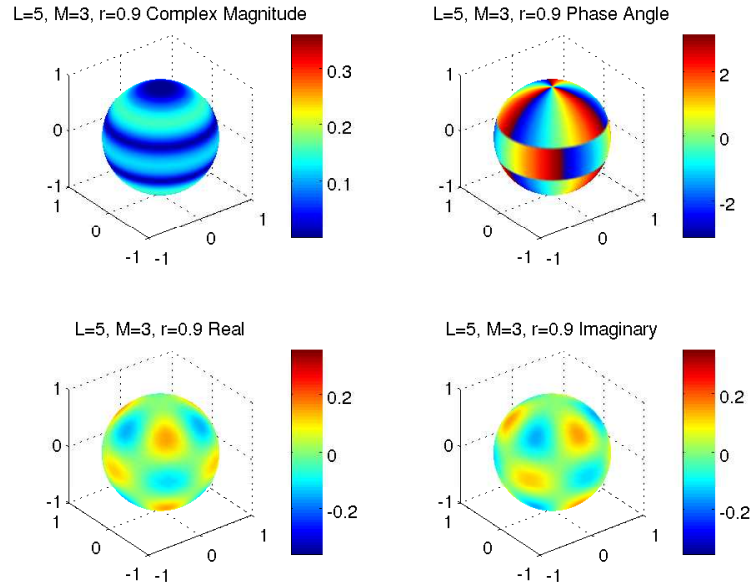


Figure 7.2: The magnitude, phase angle, real, and imaginary parts of one of the spherical harmonics ($L = 5, M = 3$), where L and M are the indexes.



(a) $r=0.3$



(b) $r=0.9$

Figure 7.3: The magnitude, phase angle, real, and imaginary parts of one of the OSHMMs ($L = 5, M = 3$) at radius, $r=0.3$ and $r=0.9$

Appendix A

Plots of Zernike Radial Polynomials

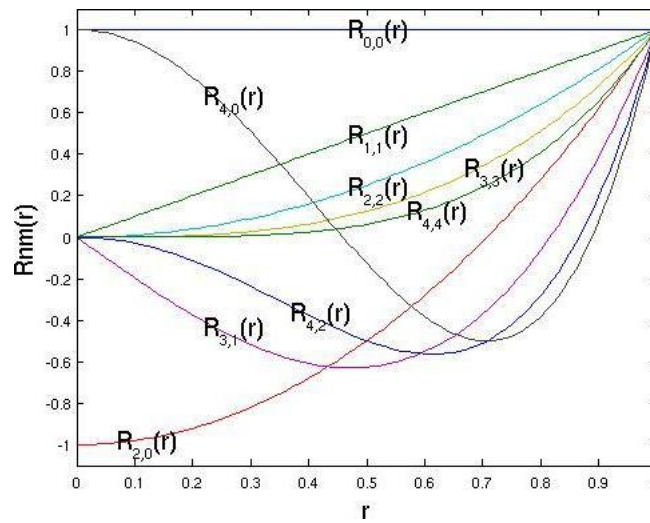


Figure A.1: Zernike Radial Polynomials $R_{nm}(r)$, ($n = 0, 1, \dots, 4$)

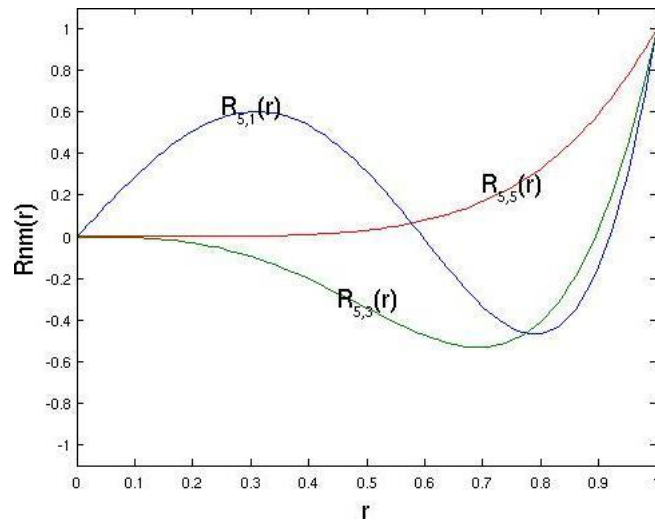


Figure A.2: Zernike Radial Polynomials $R_{nm}(r)$, ($n = 5$)

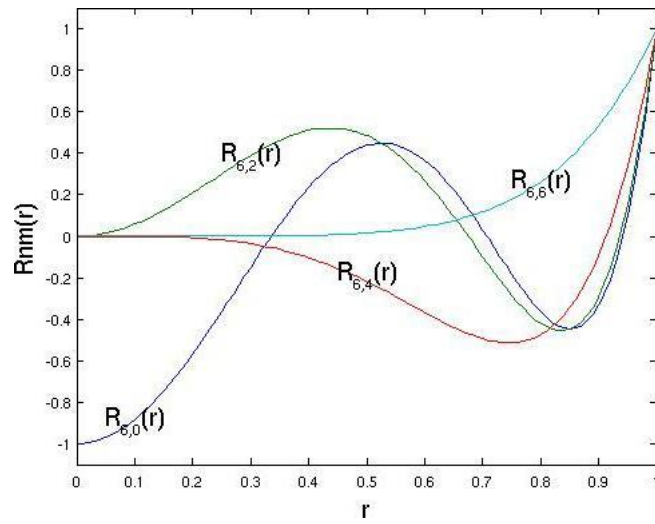


Figure A.3: Zernike Radial Polynomials $R_{nm}(r)$, ($n = 6$)

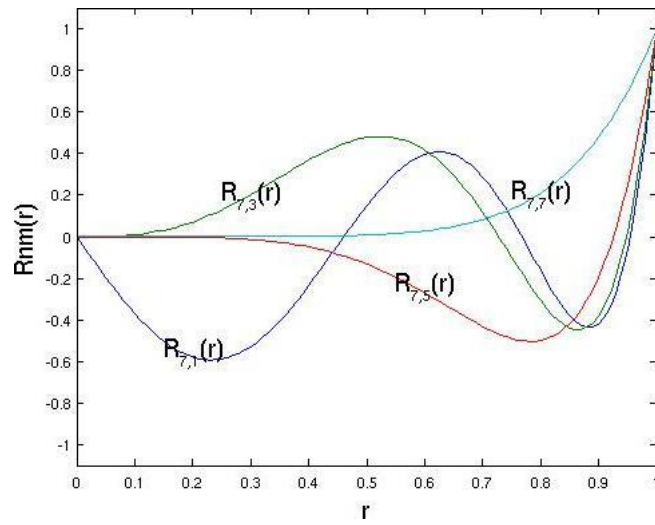


Figure A.4: Zernike Radial Polynomials $R_{nm}(r)$, ($n = 7$)

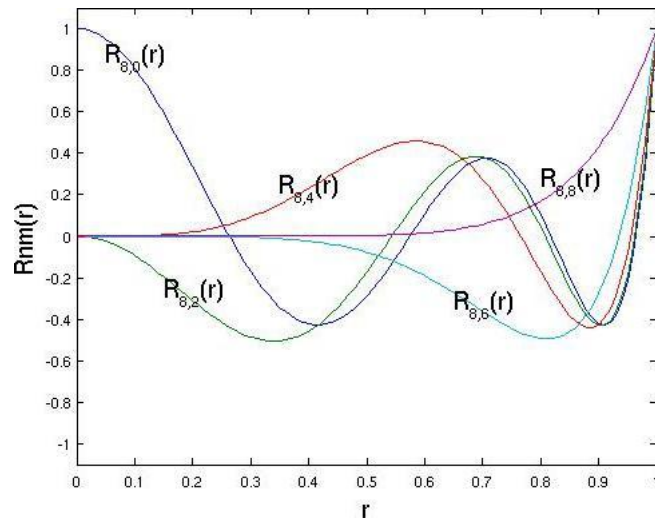


Figure A.5: Zernike Radial Polynomials $R_{nm}(r)$, ($n = 8$)

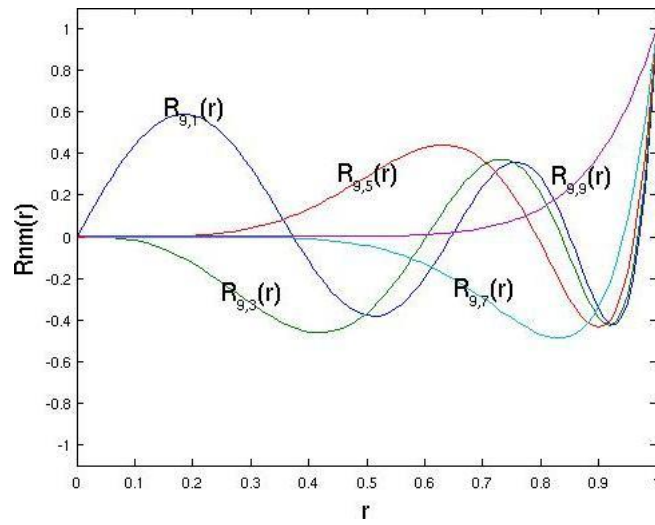


Figure A.6: Zernike Radial Polynomials $R_{nm}(r)$, ($n = 9$)

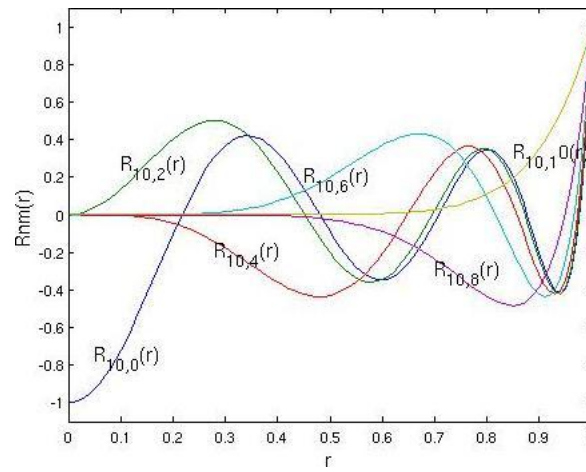


Figure A.7: Zernike Radial Polynomials $R_{nm}(r)$, ($n = 10$)

Appendix B

Lists of Zernike Moment Bases

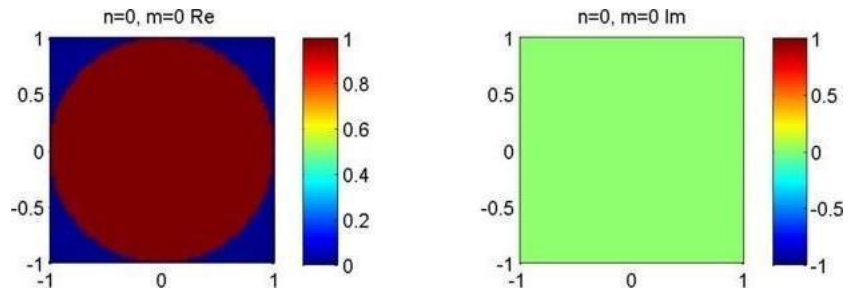


Figure B.1: Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 0, m = 0$)

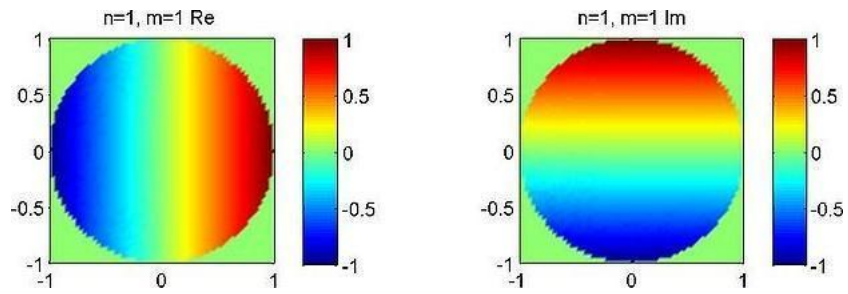


Figure B.2: Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 1, m = 1$)

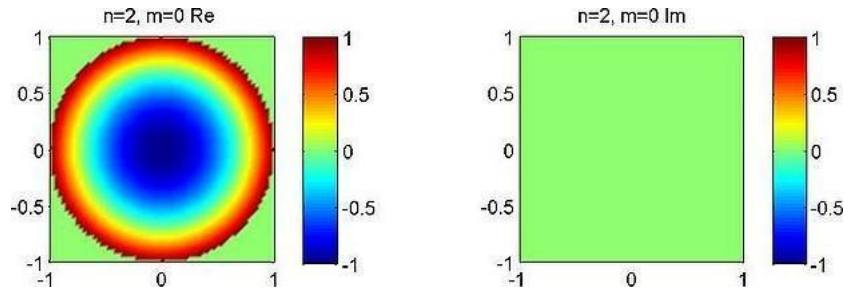


Figure B.3: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 2, m = 0)$

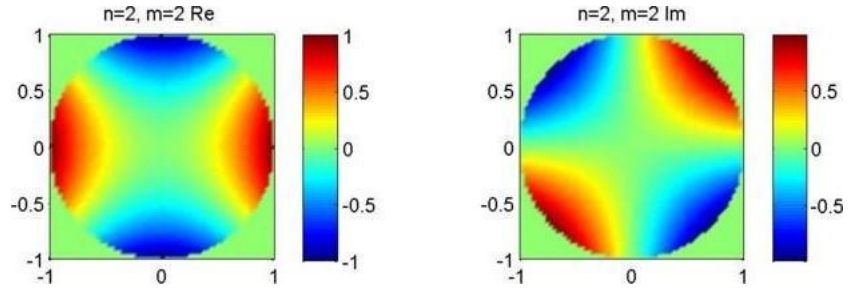


Figure B.4: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 2, m = 2)$

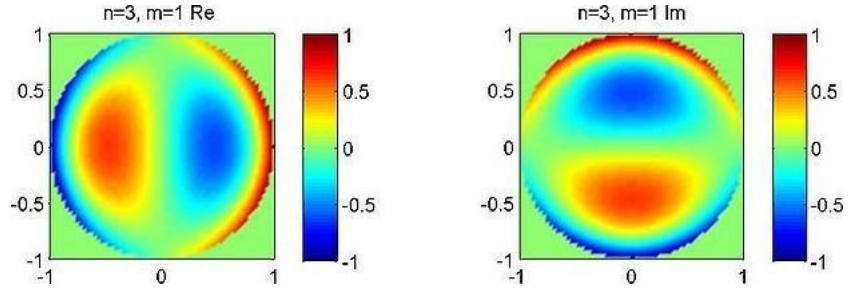


Figure B.5: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 3, m = 1)$

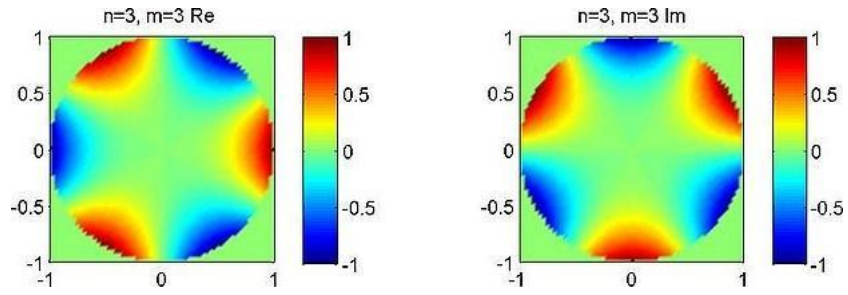


Figure B.6: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 3, m = 3)$

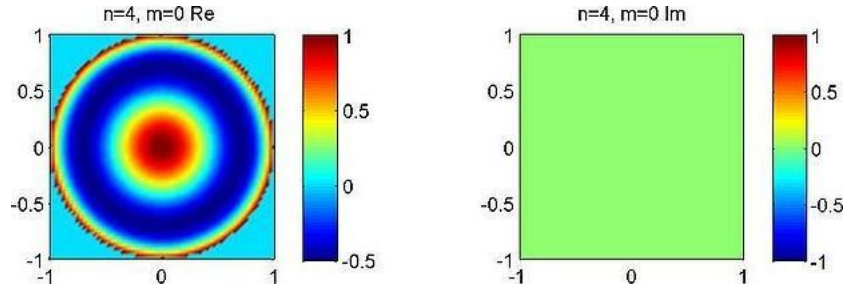


Figure B.7: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 4, m = 0)$

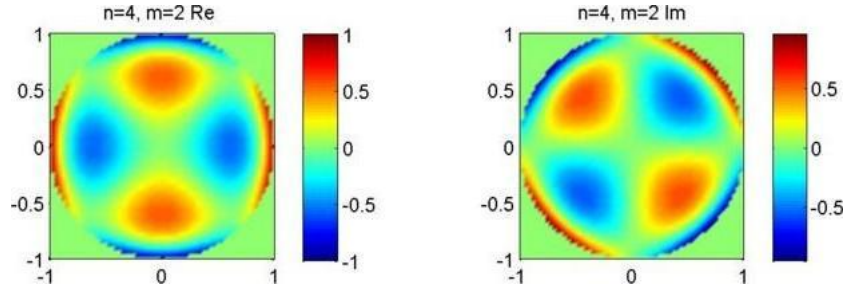


Figure B.8: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 4, m = 2)$

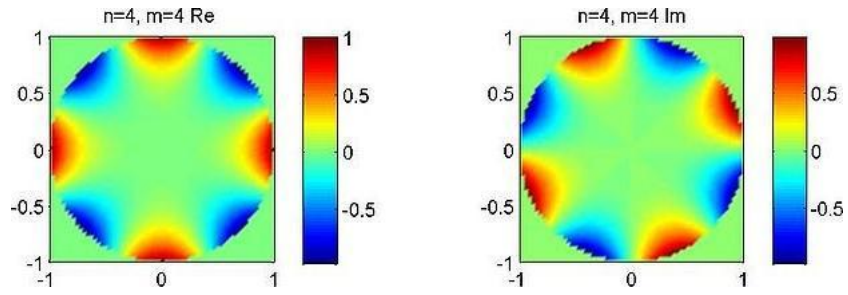


Figure B.9: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 4, m = 4)$

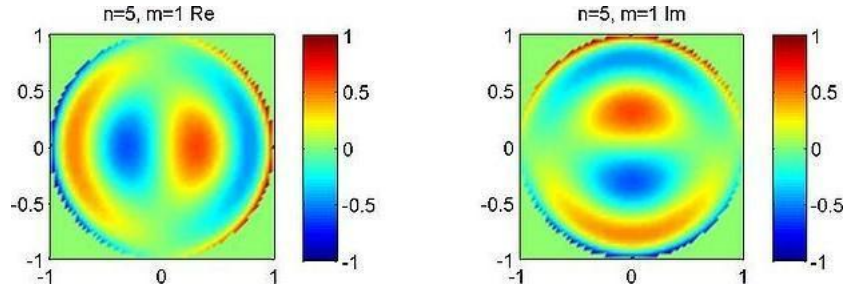


Figure B.10: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 5, m = 1)$

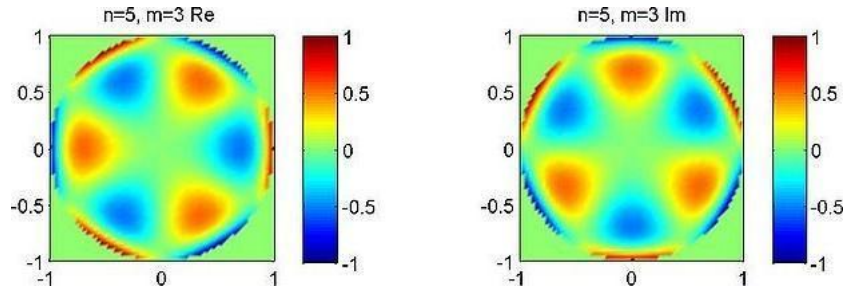


Figure B.11: Real and imaginary parts of the ZM base $V_{nm}(r)$, $(n = 5, m = 3)$

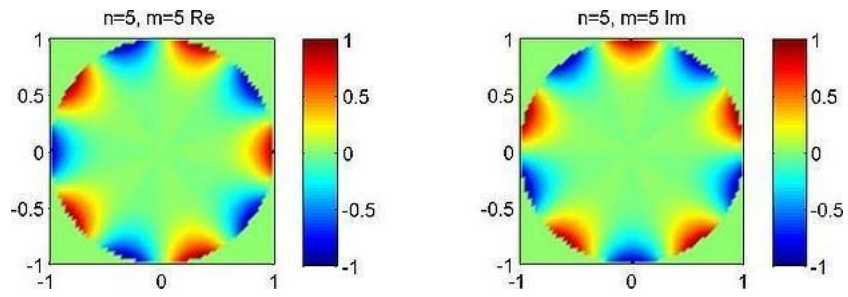


Figure B.12: Real and imaginary parts of the ZM base $V_{nm}(r)$, ($n = 5, m = 5$)

Appendix C

Lists of Orthogonal Fourier-Mellin Moment Bases

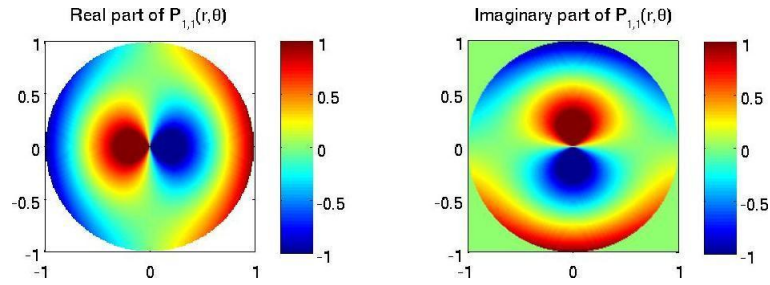


Figure C.1: Real and imaginary parts of OFMM base, ($n=1, m=1$)

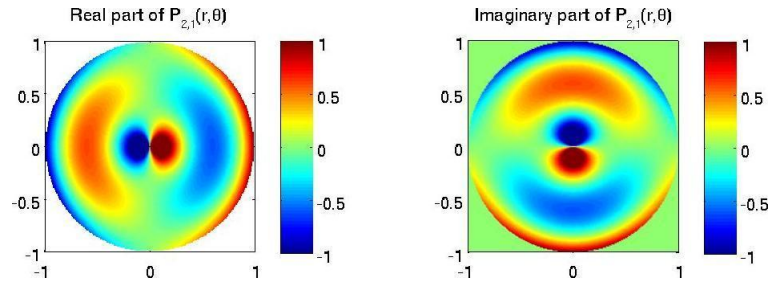


Figure C.2: Real and imaginary parts of OFMM base, ($n=2, m=1$)

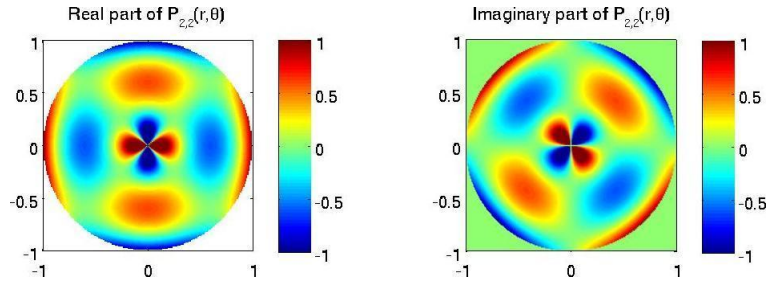


Figure C.3: Real and imaginary parts of OFMM base, (n=2,m=2)

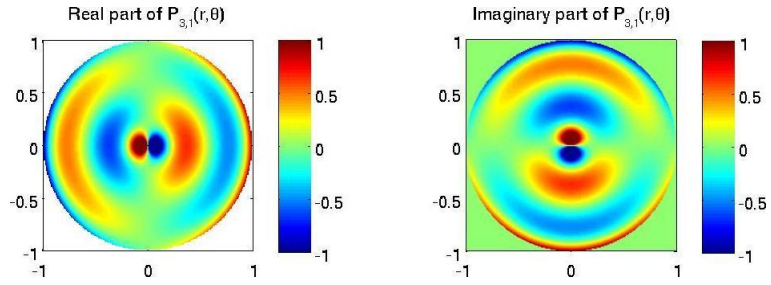


Figure C.4: Real and imaginary parts of OFMM base, (n=3,m=1)

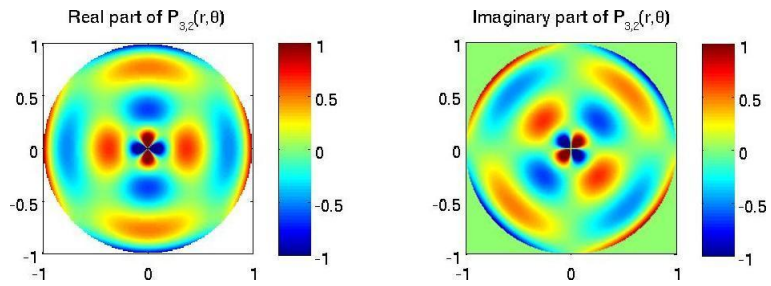


Figure C.5: Real and imaginary parts of OFMM base, (n=3,m=2)

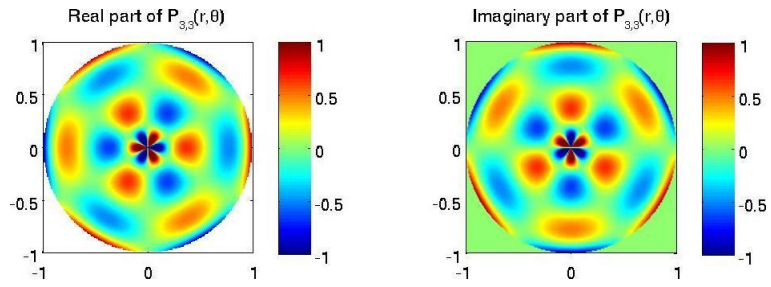


Figure C.6: Real and imaginary parts of OFMM base, (n=3,m=3)

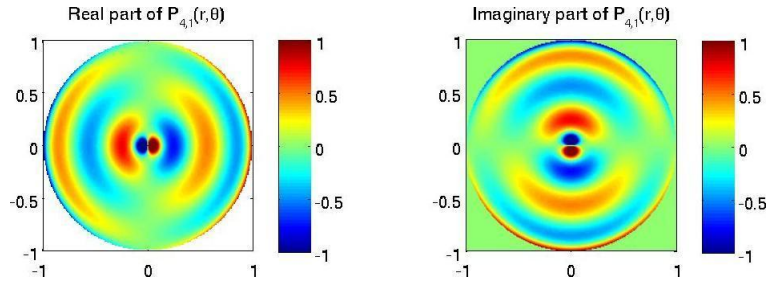


Figure C.7: Real and imaginary parts of OFMM base, (n=4,m=1)

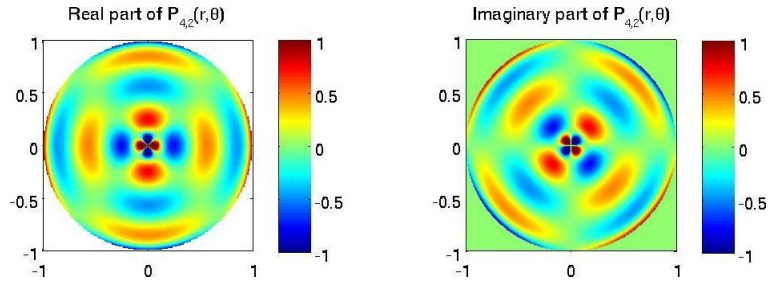


Figure C.8: Real and imaginary parts of OFMM base, (n=4,m=2)

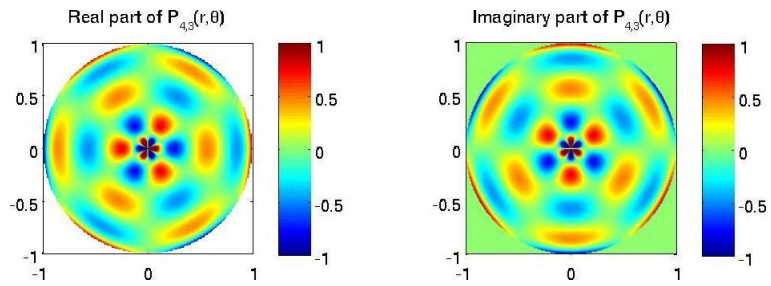


Figure C.9: Real and imaginary parts of OFMM base, (n=4,m=3)

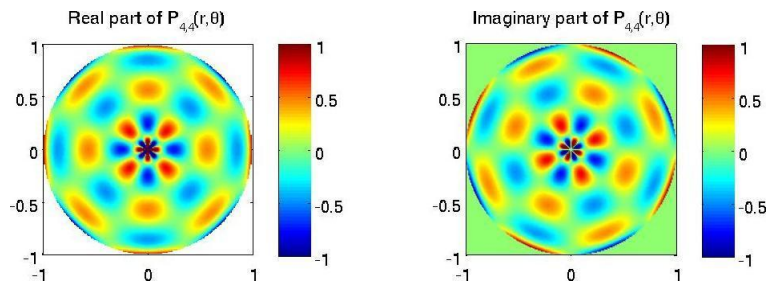


Figure C.10: Real and imaginary parts of OFMM base, (n=4,m=4)

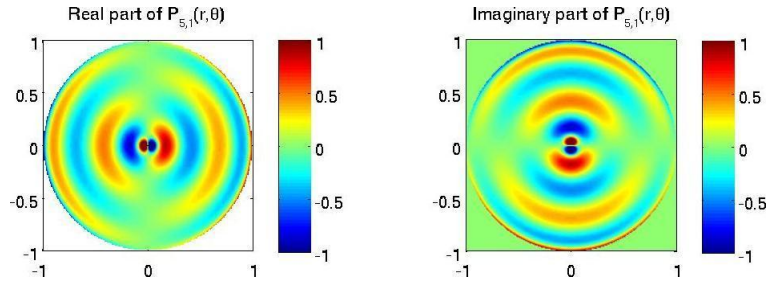


Figure C.11: Real and imaginary parts of OFMM base, (n=5,m=1)

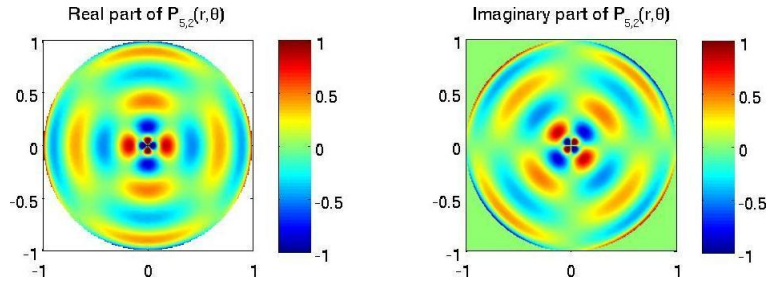


Figure C.12: Real and imaginary parts of OFMM base, (n=5,m=2)

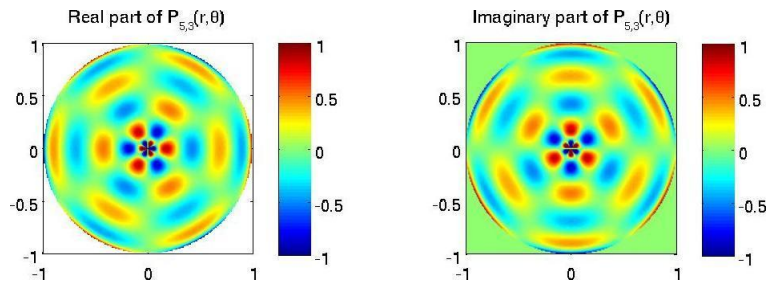


Figure C.13: Real and imaginary parts of OFMM base, (n=5,m=3)

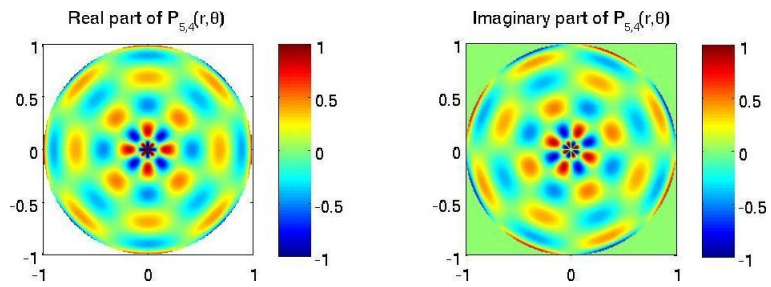


Figure C.14: Real and imaginary parts of OFMM base, (n=5,m=4)

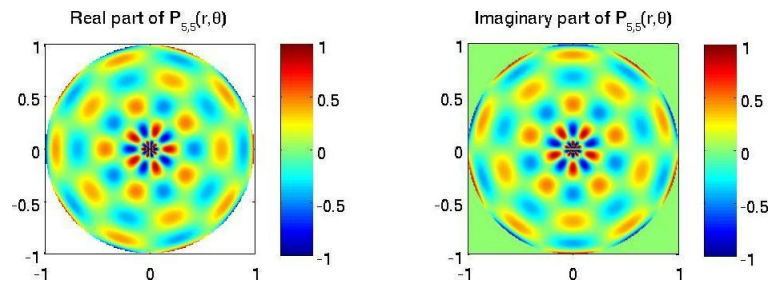


Figure C.15: Real and imaginary parts of OFMM base, ($n=5, m=5$)

Bibliography

- [1] Q. M. Tieng and W. W. Boles, “Wavelet-based affine invariant representation: A tool for recognizing planar objects in 3D space,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, pp. 846–857, August 1997.
- [2] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, and S. Huovinen, “Outex - new framework for empirical evaluation of texture analysis algorithms,” in *Pattern Recognition, 2002. 16th IAPR International Conference on*, pp. 701 – 706, 2002.
- [3] Google, “Google images.” <http://images.google.com/>.
- [4] T. Kato, “Database architecture for content-based image retrieval,” *Image Storage and Retrieval Systems*, vol. 1662, no. 1, pp. 112–123, 1992.
- [5] C. W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, “QBIC project: querying images by content, using color, texture, and shape,” *Storage and Retrieval for Image and Video Databases*, vol. 1908, no. 1, pp. 173–187, 1993.
- [6] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petkovic, and R. Barber, “Efficient and effective querying by image content,” *Journal of Intelligent Information Systems*, vol. 3, pp. 231–262, 1994.
- [7] M. K. Mandal, F. Idris, and S. Panchanathan, “A critical evaluation of image and video indexing techniques in the compressed domain,” *Image and Vision Computing*, vol. 17, pp. 513–529, 1999.

- [8] Y. A. Aslandogan and C. T. Yu, "Techniques and systems for image and video retrieval," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 11, no. 1, pp. 56–63, 1999.
- [9] A. Yoshitaka and T. Ichikawa, "A survey on content-based retrieval for multimedia databases," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 11, pp. 81–93, January/February 1999.
- [10] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, pp. 99–121, November 2000.
- [11] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, pp. 1349–1380, December 2000.
- [12] R. Datta, J. Li, and J. Z. Wang, "Content-based image retrieval: approaches and trends of the new age," in *Multimedia Information Retrieval, 2005. 7th ACM SIGMM International Workshop on*, pp. 253–262, 2005.
- [13] R. Marée, P. Geurts, and L. Wehenkel, "Content-based image retrieval by indexing random subwindows with randomized trees," *IPSI Transactions on Computer Vision and Applications*, vol. 1, pp. 46–57, January 2009.
- [14] Google, "Use pictures to search the web." <http://www.google.com/mobile/goggles/#landmark>.
- [15] PCWorld, "A hands on tour:google goggles visual search." http://www.pcworld.com/article/183933/a_handson_tour_google_goggles_visual_search.html.
- [16] X. Wan and C.-C. Kuo, "A new approach to image retrieval with hierarchical colour clustering," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, pp. 628–643, September 1998.

- [17] S.-C. Pei and C.-M. Cheng, “Extracting colour features and dynamic matching for image database retrieval,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 9, pp. 501–512, April 1999.
- [18] N. Sebe and M. S. Lew, “Colour-based retrieval,” *Pattern Recognition Letters*, vol. 22, no. 2, pp. 223–230, 2001.
- [19] A. Ferman, A. Tekalp, and R. Mehrotra, “Robust colour histogram descriptors for video segment retrieval and identification,” *Image Processing, IEEE Transactions on*, vol. 11, pp. 497–508, May 2002.
- [20] Y. Deng, B. Manjunath, C. Kenney, M. Moore, and H. Shin, “An efficient colour representation for image retrieval,” *Image Processing, IEEE Transactions on*, vol. 10, pp. 140–147, January 2001.
- [21] S. Jeong, C. S. Won, and R. M. Gray, “Image retrieval using colour histograms generated by gauss mixture vector quantization,” *Computer Vision and Image Understanding*, vol. 94, no. 1-3, pp. 44–66, 2004.
- [22] B. S. Manjunath and W. Ma, *Texture Features for Image Retrieval*. Image Databases: Search and Retrieval of Digital Imagery, V. Castelli and L. D. Bergman (editors), 2002.
- [23] M. Do and M. Vetterli, “Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance,” *Image Processing, IEEE Transactions on*, vol. 11, pp. 146–158, February 2002.
- [24] Y. Rubner and C. Tomasi, “Texture-based image retrieval without segmentation,” in *Computer Vision, 1999. 7th IEEE International Conference on*, vol. 2, pp. 1018–1024, 1999.
- [25] S. Choy and C. Tong, “Statistical wavelet subband characterization based on generalized gamma density and its application in texture retrieval,” *Image Processing, IEEE Transactions on*, vol. 19, pp. 281–289, February 2010.

- [26] A. Jain and A. Vailaya, "Shape-based retrieval: A case study with trademark image databases," *Pattern Recognition*, vol. 31, pp. 1369–1390, 1998.
- [27] I. Bartolini, P. Ciaccia, and M. Patella, "Warp: accurate retrieval of shapes using phase of fourier descriptors and time warping distance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, pp. 142–147, January 2005.
- [28] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 509–522, April 2002.
- [29] B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada, "colour and texture descriptors," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, pp. 703–715, June 2001.
- [30] S. Liapis and G. Tziritas, "colour and texture image retrieval using chromaticity histograms and wavelet frames," *Multimedia, IEEE Transactions on*, vol. 6, pp. 676–686, October 2004.
- [31] Y. Mei and D. Androutsos, "Colour texture retrieval using wavelet decomposition on the hue/saturation plane," in *Multimedia and Expo, 2008. IEEE International Conference on*, pp. 877–880, July 2008.
- [32] Y. Mei and D. Androutsos, "Wavelet-based colour texture retrieval using the independent component colour space," in *Image Processing, 2008. 15th IEEE International Conference on*, pp. 165–168, October 2008.
- [33] Y. D. Chun, N. C. Kim, and I. H. Jang, "Content-based image retrieval using multi-resolution colour and texture features," *Multimedia, IEEE Transactions on*, vol. 10, pp. 1073–1084, October 2008.
- [34] J. L. Mundy and A. Zisserman, eds., *Geometric invariance in computer vision*. Cambridge, MA, USA: MIT Press, 1992.

- [35] Y. Mei and D. Androutsos, “Affine invariant shape descriptors: The ICA-Fourier descriptor and the PCA-Fourier descriptor,” in *Pattern Recognition, 2008. 19th IAPR International Conference on*, pp. 1–4, December 2008.
- [36] A. Hyvärinen and U. Köster, “A fast fixed-point algorithm for independent component analysis,” *Neural Computation*, vol. 9, pp. 1483–1492, 1997.
- [37] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-time signal processing (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.
- [38] Y. Mei and D. Androutsos, “Robust affine invariant shape image retrieval using the ICA Zernike moment shape descriptor,” in *Image Processing, 2009. 16th IEEE International Conference on*, pp. 1065–1068, November 2009.
- [39] Y. Mei and D. Androutsos, “Robust affine invariant region-based shape descriptors: The ICA Zernike moment shape descriptor and the whitening Zernike moment shape descriptor,” *Signal Processing Letters, IEEE*, vol. 16, pp. 877–880, October 2009.
- [40] D. Zhang and G. Lu, “A comparative study on shape retrieval using Fourier descriptors with different shape signatures,” *Journal of Visual Communication and Image Representation*, no. 14 (1), pp. 41–60, 2003.
- [41] H. Freeman, “On the encoding of arbitrary geometric configurations,” *Electronic Computers, IEEE Transactions on*, vol. 10, pp. 260–268, June 1961.
- [42] J. Saghri and H. Freeman, “Analysis of the precision of generalized chain codes for the representation of planar curves,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 3, pp. 533–539, September 1981.
- [43] J. Iivarinen and A. J. E. Visa, “Shape recognition of irregular objects,” *Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling*, vol. 2904, no. 1, pp. 25–32, 1996.
- [44] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007.

- [45] M. Peura and J. Iivarinen, "Efficiency of simple shape descriptors," in *Advances in Visual Form Analysis*, 1997.
- [46] S. Abbasi, F. Mokhtarian, and J. Kittler, "Curvature scale space image in shape similarity retrieval," *Multimedia System*, vol. 7, no. 6, pp. 467–476, 1999.
- [47] H. Kauppinen, T. Seppänen, and M. Pietikäinen, "An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, pp. 201–207, February 1995.
- [48] Y. K. Liu and B. Zalik, "An efficient chain code with Huffman coding," *Pattern Recognition*, vol. 38, no. 4, pp. 553–557, 2005.
- [49] T. D. Bui and G. Chen, "Invariant Fourier-wavelet descriptor for pattern recognition," *Pattern Recognition*, vol. 32, pp. 1083–1088, 1999.
- [50] D. Zhang and G. Lu, "Shape-based image retrieval using generic Fourier descriptor," *Signal Processing: Image Communication*, vol. 17, no. 10, pp. 825–848, 2002.
- [51] S. Tabbone, L. Wendling, and J.-P. Salmon, "A new shape descriptor defined on the Radon transform," *Computer Vision and Image Understanding*, vol. 102, no. 1, pp. 42–51, 2006.
- [52] S. R. Deans, *The Radon Transform and Some of Its Applications*. Krieger Publishing Company.
- [53] M. K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. IT-8, pp. 179–187, February 1962.
- [54] A. M. Abo Zaid and E. Horne, "Generalised complex moment descriptors," in *Circuits and Systems, 1991. 34th Midwest Symposium on*, pp. 151–154, 1991.
- [55] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, pp. 489–497, May 1990.

- [56] Å. Wallin and O. Kübler, "Complete sets of complex Zernike moment invariants and the role of the pseudoinvariants," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, pp. 1106–1110, November 1995.
- [57] K. Arbter, W. Snyder, H. Burkhardt, and G. Hirzinger, "Application of affine-invariant Fourier descriptors to recognition of 3-D objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, pp. 640–647, July 1990.
- [58] Q. Jin and P. Yan, "A new method of extracting invariants under affine transform," in *Pattern Recognition, 1992. 11th IAPR International Conference on*, pp. 742–745, August 1992.
- [59] A. E. Oirrak, M. Daoudi, and D. Aboutajdine, "Affine invariant descriptors using Fourier series," *Pattern Recognition Letters*, vol. 23, no. 10, pp. 1109–1118, 2002.
- [60] M. I. Khalil and M. M. Bayoumi, "A dyadic wavelet affine invariant function for 2D shape recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, pp. 1152–1164, October 2001.
- [61] I. E. Rube', M. Ahmed, and M. Kamel, "Wavelet approximation-based affine invariant shape representation functions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, pp. 323–327, February 2006.
- [62] S. Abbasi and F. Mokhtarian, "Curvature scale space with affine length parametrisation," in *Scale-Space Theories in Computer Vision, 1999. 2nd International Conference on*, pp. 435–440, 1999.
- [63] F. Mokhtarian and S. Abbasi, "Shape similarity retrieval under affine transforms," *Pattern Recognition*, vol. 35, no. 1, pp. 31–41, 2002.
- [64] X. Huang, B. Wang, and L. Zhang, "A new scheme for extraction of affine invariant descriptor and affine motion estimation based on independent component analysis," *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1244–1255, 2005.

- [65] N. Guney and A. Ertuzun, “Undoing the affine transformation using blind source separation,” in *Independent Component Analysis and Blind Signal Separation*, vol. 3889 of *Lecture Notes in Computer Science*, pp. 360–367, Springer, 2006.
- [66] J. Flusser and T. Suk, “Pattern recognition by affine moment invariants,” *Pattern Recognition*, vol. 26, no. 1, pp. 167–174, 1993.
- [67] J. Flusser and T. Suk, “Affine moment invariants: a new tool for character recognition,” *Pattern Recognition Letters*, vol. 15, no. 4, pp. 433–436, 1994.
- [68] J. Flusser and T. Suk, “A moment-based approach to registration of images with affine geometric distortion,” *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 32, pp. 382–387, March 1994.
- [69] G. Taubin and D. Cooper, “Object recognition based on moment (or algebraic) invariants,” in *Geometric Invariance in Computer Vision 1992*, p. Chapter 19, 1992.
- [70] D. Zhao and J. Chen, “Affine curve moment invariants for shape recognition,” *Pattern Recognition*, vol. 30, no. 6, pp. 895–901, 1997.
- [71] H. W. Guggenheimer, *Differential Geometry*. New York, NY, USA: McGraw-Hill, 1963.
- [72] P. Modenov and A. Parkhomenko, *Geometric Transformations*, vol. 1. New York: Academic Press, 1965.
- [73] S. Mallat, “Zero-crossings of a wavelet transform,” *Information Theory, IEEE Transactions on*, vol. 37, pp. 1019–1033, July 1991.
- [74] S. Mallat and S. Zhong, “Characterization of signals from multiscale edges,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, pp. 710–732, July 1992.
- [75] M. Shensa, “The discrete wavelet transform: wedding the a trous and mallat algorithms,” *Signal Processing, IEEE Transactions on*, vol. 40, pp. 2464–2482, October 1992.

- [76] J. Fowler, “The redundant discrete wavelet transform and additive noise,” *Signal Processing Letters, IEEE*, vol. 12, pp. 629 – 632, September 2005.
- [77] S. Abbasi and F. Mokhtarian, “Affine-similar shape retrieval: application to multiview 3-D object recognition,” *Image Processing, IEEE Transactions on*, vol. 10, no. 1, pp. 131–139, 2001.
- [78] F. Mokhtarian, S. Abbasi, and J. Kittler, “Robust and efficient shape indexing through curvature scale space,” in *British Machine Vision Conference*, pp. 53–62, 1996.
- [79] F. Mokhtarian and M. Bober, *Curvature Scale Space Representation: Theory, Applications, and MPEG-7 Standardization*. Norwell, MA, USA: Kluwer Academic Publishers, 2003.
- [80] G. Stewart, “The decompositional approach to matrix computation,” in *Computing in Science and Engineering*, vol. 2, pp. 50–59, Piscataway, NJ, USA: IEEE Educational Activities Department, 2000.
- [81] L. J. Latecki, R. Lakamper, and T. Eckhardt, “Shape descriptors for non-rigid shapes with a single closed contour,” in *Computer Vision and Pattern Recognition, 2000. IEEE Conference on*, vol. 1, pp. 424–429, 2000.
- [82] B. Ans, J. Herault, and C. Jutten, “Adaptive neural architectures: detection of primitives,” in *Proceedings of COGNITIVA’85*, pp. 593–597, 1985.
- [83] B. Arons, “A review of the cocktail party effect,” *Journal of the American Voice I/O Society*, vol. 12, pp. 35–50, 1992.
- [84] T.-W. Lee, A. Ziehe, R. Orglmeister, and T. Sejnowski, “Combining time-delayed decorrelation and ICA: towards solving the cocktail party problem,” in *Acoustics, Speech and Signal Processing, 1998. IEEE International Conference on*, vol. 2, pp. 1249–1252, 1998.

- [85] D. A. Peterson and C. W. Anderson, "EEG-based cognitive task classification with ICA and neural networks," in *Artificial Neural Networks, International Workshop on*, pp. 265–272, 1999.
- [86] J. E. Moran, C. L. Drake, and N. Tepley, "ICA methods for MEG imaging.," *Neurol Clin Neurophysiol*, vol. 2004, 2004.
- [87] S.-N. Yu and K.-T. Chou, "Integration of independent component analysis and neural networks for ECG beat classification," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2841–2846, 2008.
- [88] S.-N. Yu and K.-T. Chou, "Selection of significant independent components for ECG beat classification," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2088–2096, 2009.
- [89] C. Peng, X. Qian, and D. Ye, "Electrogastrogram extraction using independent component analysis with references," *Neural Computing and Applications*, vol. 16, no. 6, pp. 581–587, 2007.
- [90] A. Hyvärinen, "Sparse code shrinkage: denoising of nongaussian data by maximum likelihood estimation," *Neural Computation*, vol. 11, no. 7, pp. 1739–1768, 1999.
- [91] A. Hyvärinen, P. Hoyer, and E. Oja, "Sparse code shrinkage for image denoising," in *Neural Networks Proceedings, 1998. IEEE International Joint Conference on*, vol. 2, pp. 859–864, 1998.
- [92] H. Li, G. Ren, and B. Xiao, "Image denoising algorithm based on independent component analysis," in *Software Engineering, 2009. WRI World Congress on*, vol. 4, pp. 465–469, May 2009.
- [93] A. D. Back and A. S. Weigend, "A first application of independent component analysis to extracting structure from stock returns," *International Journal of Neural Systems*, vol. 8, pp. 473–484, 1997.

- [94] C.-J. Lu, T.-S. Lee, and C.-C. Chiu, “Financial time series forecasting using independent component analysis and support vector regression,” *Decision Support Systems*, vol. 47, no. 2, pp. 115–125, 2009.
- [95] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. Wiley & Sons, 2001.
- [96] J. Shlens, “A tutorial on principal component analysis,” December 2005.
- [97] T. Rowland, “Orthogonal transformation.” <http://mathworld.wolfram.com/OrthogonalTransformation.html>.
- [98] J. A. Rice, *Mathematical Statistics and Data Analysis*. Duxbury Press, April 2001.
- [99] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 1991.
- [100] A. Hyvärinen, “Fast and robust fixed-point algorithms for independent component analysis,” *Neural Networks, IEEE Transactions on*, vol. 10, pp. 626–634, May 1999.
- [101] A. Hyvärinen, “New approximations of differential entropy for independent component analysis and projection pursuit,” in *Advances in neural information processing systems, 1997. 1997 conference on*, pp. 273–279, 1998.
- [102] P. Comon, “Independent component analysis, a new concept?,” *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.
- [103] A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications,” *Neural Networks, IEEE Transactions on*, vol. 13, no. 4-5, pp. 411–430, 2000.
- [104] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals & systems (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.
- [105] C.-H. Teh and R. Chin, “On image analysis by the methods of moments,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, pp. 496–513, July 1988.

- [106] S. Liao and M. Pawlak, "On image analysis by moments," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, pp. 254–266, March 1996.
- [107] M. Zhenjiang, "Zernike moment-based image shape analysis and its application," *Pattern Recognition Letters*, vol. 21, no. 2, pp. 169–177, 2000.
- [108] R. Mukundan, S. Ong, and P. Lee, "Image analysis by Tchebichef moments," *Image Processing, IEEE Transactions on*, vol. 10, pp. 1357–1364, September 2001.
- [109] H. S. Kim and H.-K. Lee, "Invariant image watermark using Zernike moments," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, pp. 766–775, August 2003.
- [110] L. Kotoulas and I. Andreadis, "Real-time computation of Zernike moments," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, pp. 801–809, June 2005.
- [111] W. Wang, J. E. Mottershead, and C. Mares, "Mode-shape recognition and finite element model updating using the Zernike moment descriptor," *Mechanical Systems and Signal Processing*, vol. 23, no. 7, pp. 2088–2112, 2009.
- [112] Z. Iscan, Z. Dokur, and T. Oelmez, "Tumor detection by using Zernike moments on segmented magnetic resonance brain images," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2540–2549, 2010.
- [113] P.-T. Yap, R. Paramesran, and S.-H. Ong, "Image analysis by Krawtchouk moments," *Image Processing, IEEE Transactions on*, vol. 12, pp. 1367–1377, November 2003.
- [114] N. Singhal, Y.-Y. Lee, C.-S. Kim, and S.-U. Lee, "Robust image watermarking using local Zernike moments," *Journal of Visual Communication and Image Representation*, vol. 20, no. 6, pp. 408–419, 2009.
- [115] Y. Sheng and L. Shen, "Orthogonal Fourier-Mellin moments for invariant pattern recognition," *Journal of the Optical Society of America A*, vol. 11, no. 6, pp. 1748–1757, 1994.

- [116] Z. Ping, H. Ren, J. Zou, Y. Sheng, and W. Bo, “Generic orthogonal moments: Jacobi-Fourier moments for invariant image description,” *Pattern Recognition*, vol. 40, no. 4, pp. 1245 – 1254, 2007.
- [117] Z. Ping, R. Wu, and Y. Sheng, “Image description with Chebyshev-Fourier moments,” *Journal of the Optical Society of America A*, vol. 19, no. 9, pp. 1748–1754, 2002.
- [118] H. Ren, Z. Ping, W. Bo, W. Wu, and Y. Sheng, “Multidistortion-invariant image recognition with radial harmonic Fourier moments,” *Journal of the Optical Society of America A*, vol. 20, no. 4, pp. 631–637, 2003.
- [119] G. Papakostas, Y. Boutalis, C. Papaodysseus, and D. Fragoulis, “Numerical stability of fast computation algorithms of Zernike moments,” *Applied Mathematics and Computation*, vol. 195, no. 1, pp. 326–345, 2008.
- [120] J. Revaud, G. Lavoue, and A. Baskurt, “Improving Zernike moments comparison for optimal similarity and rotation angle retrieval,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, pp. 627–636, April 2009.
- [121] J. Turuwheua, “Corneal surface reconstruction algorithm using zernike polynomial representation: improvements,” *Journal of the Optical Society of America A*, vol. 24, no. 6, pp. 1551–1561, 2007.
- [122] D. Iskander, M. Collins, and B. Davis, “Optimal modeling of corneal surfaces with zernike polynomials,” *Biomedical Engineering, IEEE Transactions on*, vol. 48, pp. 87 –95, January 2001.
- [123] S. Ghosal and R. Mehrotra, “Segmentation of range images: an orthogonal moment-based integrated approach,” *Robotics and Automation, IEEE Transactions on*, vol. 9, pp. 385 –399, August 1993.
- [124] S. Ghosal and R. Mehrotra, “Edge detection using orthogonal moment-based operators,” in *Pattern Recognition, 1992. 11th IAPR International Conference on*, pp. 413 –416, 1992.

- [125] M. Pawlak, "On the reconstruction aspects of moment descriptors," *Information Theory, IEEE Transactions on*, vol. 38, pp. 1698–1708, November 1992.
- [126] N. Singhal, Y.-Y. Lee, C.-S. Kim, and S.-U. Lee, "Robust image watermarking using local zernike moments," *Journal of Visual Communication and Image Representation*, vol. 20, no. 6, pp. 408–419, 2009.
- [127] W. Kim and Y. Kim, "A region-based shape descriptor using Zernike moments," *Signal Processing: Image Communication*, vol. 16, pp. 95–102, September 2000.
- [128] S. Li, M.-C. Lee, and C.-M. Pun, "Complex Zernike moments features for shape-based image retrieval," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 39, pp. 227–237, January 2009.
- [129] F. Zernike, "Diffraction theory of the cut procedure and its improved form, the phase contrast method," *Physica*, vol. 1, pp. 689–704, 1934.
- [130] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.
- [131] J. Wang, Y. Sun, and Q. Chen, "Recognition of digital annotation with invariant HONN based on orthogonal Fourier-Mellin moments," in *Machine Learning and Cybernetics, 2003 International Conference on*, vol. 4, pp. 2261–2264 Vol.4, November 2003.
- [132] G. Papakostas, Y. Boutalis, D. Karras, and B. Mertzios, "Fast numerically stable computation of orthogonal Fourier-Mellin moments," *Computer Vision, IET*, vol. 1, pp. 11–16, March 2007.
- [133] K. M. Hosny, M. A. Shouman, and H. M. A. Salam, "Fast computation of orthogonal Fourier-Mellin moments in polar coordinates," *Journal of Real-Time Image Processing*, 2009.
- [134] H. Zhang, H. Z. Shu, P. Haigron, B. S. Li, and L. M. Luo, "Construction of a complete set of orthogonal Fourier-Mellin moment invariants for pattern recognition applications," *Image and Vision Computing*, vol. 28, no. 1, pp. 38–44, 2010.

- [135] G. Friedland, K. Jantz, and R. Rojas, “Siox: Simple interactive object extraction in still images,” in *Multimedia, International Symposium on*, pp. 253–260, 2005.
- [136] B. Thai and G. Healey, “Spatial filter selection for illumination-invariant colour texture discrimination,” in *Computer Vision and Pattern Recognition, 1999. IEEE Conference on*, vol. 2, p. 2154, 1999.
- [137] G. Healey and D. Slater, “Computing illumination-invariant descriptors of spatially filtered colour image regions,” *Image Processing, IEEE Transactions on*, vol. 6, pp. 1002–1013, July 1997.
- [138] R. Haberman, *Applied partial differential equations : with Fourier series and boundary value problems*. Upper Saddle River, N.J.: Pearson/Prentice Hall, 4th ed., 2004.