

NONLINEAR ADAPTIVE NOISE CANCELLING FOR IMAGE WITH ADAPTIVE NETWORK-BASED FUZZY INFERENCE SYSTEMS

A Thesis

Presented to

The Faculty of Graduate Studies

of

The University of Guelph

by

Hao Qin

In partial fulfillment of requirements

for the degree of

Master of Science

June, 2015

© Hao Qin, 2015

Abstract

Image Denoising with Adaptive Neuro Fuzzy Inference Systems

Hao Qin

University of Guelph, 2004

Advisor:

Professor Simon X. Yang

Neuro-fuzzy systems are capable of inducting rules from observations, where the adaptive neuro-fuzzy inference system (ANFIS) is an effective method that can be applied to a variety of domains such as pattern recognition, robotics, nonlinear regression, nonlinear system identification and adaptive signal processing. However, the signals processed by a typical ANFIS are in one dimension, e.g., acoustic signals. In this thesis, we extend the ANFIS method to two-dimensional signals, images. First, the image restoration contaminated with Gaussian noise is investigated in nonlinear passage dynamics of orders 2 and 3. We inspect 8 types of membership functions. In addition, several issues, such as the training epochs, the number of membership functions for each input, the optimization method, the type of output membrane functions and the over-fitting problem, are investigated. Secondly, we use ANFIS to restore the image corrupted by salt and pepper noise. Finally, besides the experiment of gray image, the RGB color image corrupted by the different type of noise at different intensity is studied to remove noise and restore. We compare the effect of the image restoration by ANFIS with the conventional filters including spatial filters, frequency domain filters, adaptive optimal filter, Wiener filter, and wavelet and wavelet packet. Spatial filters include arithmetic mean filter, geometric mean filter, harmonic mean filter, contraharmonic mean filter, median filter, max and min filters, midpoint filter, alpha-trimmed mean filter, and signal dependent rank order mean filter. For frequency domain filters, we investigate ideal filter, Butterworth lowpass filters and Gaussian lowpass filter and their design methods, such as frequency transformation method, frequency sampling method and windowing method. For wavelet methods,

we discuss the hard threshold and soft threshold selection rules for denoising.

Dedication

First to the glory of Almighty God;

Secondly to

*my mother who teaches me loving life,
my father who teaches me loving knowledge,
my wife who encourages me bravely facing frustration,
and my son who makes me delight and gives me hope.*

Acknowledgements

I would like to thank Dr. Simon Yang, my supervisor, for his many suggestions and constant support during this research. I can not reach the light of the day without his support. He has devoted so much time and effort to teaching me both in this research and in writing it that my labours will never be able to match. He provided a motivating, enthusiastic, and critical atmosphere during the many discussions we had. His overly enthusiasm and integral view on research and his mission for providing only high-quality work, has made a deep impression on me. I am also thankful to Dr. Robert Dony for his guidance through the early years of chaos and confusion.

I am grateful to my friends, Zhemin Wang, Aifang Wu and Wenlu Feng, Huaishi Xu and my cousin, Xiaoling Qin for their help in my life when I started my life in Canada.

Above all, thank you my wife and my parents for your love, support and taking over my share of household chores, a huge incentive to thesis writing. Also thanks my sister, Wen Qin and my brother-in-law, Jingsong for their financial and sprite support.

Last but by no means least, I wish to thank the following: Yu Li and Xiaojun Bao for his friendship and help. I also faithfully appreciate Mr. Robert Sharpe for the literal advices.

I acknowledge that all the original pictures used in this thesis obtained from MATLAB issued by Mathworks Company or download free from Webshots Company for academic research objective.

Contents

List of Tables	x
List of Figures	xiii
List of Symbols	xxx
Preface	xxxii
1 Introduction	1
1.1 Problem Statement	1
1.2 Objectives of this Thesis	2
1.3 Contributions of this Thesis	3
1.4 Organization of this Thesis	5
2 Background and Literature Survey	8
2.1 Literature Survey	8
2.1.1 Image Processing	10
2.1.2 Image Processing with Soft Computing	12
2.1.3 Restoration of Images Corrupted by Noise with Soft Computing	14
2.2 Image Restoration Models	15
2.2.1 Mathematical Expression of Imaging System	16
2.2.2 Image Degradation Models	20
2.2.3 Summary of Image Restoration Methods	22
2.2.4 Algebraic Methods for Image Restoration	23

2.3	Restoration with Spatial Filtering, Frequency Domain Filtering and Wavelet Filtering	25
2.3.1	Spatial Filtering	25
2.3.2	Frequency Domain Filtering	31
2.3.3	Wiener Filter	37
2.3.4	Denoising with Wavelet Analysis	40
2.4	Denoising Images with Adaptive Neuro-Fuzzy Inference Systems	50
2.4.1	Introduction	52
2.4.2	Adaptive Network-based Fuzzy Inference Systems	58
3	Proposed Method for Two Demensional Image Denoising with ANFIS Algorithm	66
3.1	Transformation from 2-D Image Matrix to 1-D Sequence Vector	66
3.2	Generating and Adding Noises into the Processed Images	69
3.3	Nonlinear Adaptive Noise Cancellation with ANFIS	69
3.4	Restoration of Images Corrupted by the Noise of the Nonlinear Passage Dynamics with ANFIS	75
3.4.1	Restoration of Images Corrupted by the Noise of the Nonlinear Passage Dynamics of Order 2 with ANFIS	75
3.4.2	Restoration of Images Corrupted by the Noise of the Nonlinear Passage Dynamics of Order 3 with ANFIS	75
3.4.3	Other Modifications for 2-D Image Denoising with ANFIS Algorithm	76
3.4.4	Other Factors Discussed in the Proposed Method	76
3.5	Transformation from 1-D Sequence Vector Back to 2-D Image Matrix	77
4	Results of Image Denoising with ANFIS to Different Types of Nonlinear Passage Dynamics and of Noises	78
4.1	Application of ANFIS to the Nonlinear Passage Dynamics of Order 2 to Restore an Image Corrupted by Noise	78

4.1.1	Application of ANFIS to Restore an Image Corrupted by Gaussian Noise	79
4.1.2	Application of ANFIS with Different MFs	83
4.1.3	Discussion of Parameters of ANFIS	107
4.2	Application of ANFIS to the Nonlinear Passage Dynamics of Order 3 to Restore an Image Corrupted by Gaussian Noise	127
4.2.1	Application of ANFIS with the Default Bell MFs	128
4.2.2	Summary of All Membership Functions	130
4.2.3	Discussion of Parameters of ANFIS	132
4.3	Application of ANFIS to the Nonlinear Passage Dynamics of Order 2 to Restore an Image Corrupted by Salt and Pepper Noise	134
4.3.1	Application of ANFIS with the Default Bell MFs	138
4.3.2	Summary of All Membership Functions	139
4.3.3	Discussion on Parameters of ANFIS	139
4.4	Application of ANFIS to the Nonlinear Passage Dynamics of Order 3 to Restore an Image Corrupted by Salt & Pepper Noise	142
4.4.1	Application of ANFIS with the Default Bell MFs	143
4.4.2	Discussion on Parameters of ANFIS	145
5	Comparisons between ANFIS and Conventional Filters	147
5.1	Spatial Filtering	147
5.1.1	Arithmetic Mean Filter	148
5.1.2	Geometric Mean Filter	148
5.1.3	Harmonic Mean Filter	148
5.1.4	Contraharmonic Mean Filter	148
5.1.5	Median Filter	152
5.1.6	Max Filter	152
5.1.7	Min Filter	152
5.1.8	Midpoint Filter	157
5.1.9	Alpha-trimmed Mean Filter	157

5.1.10	SD-ROM Filter	157
5.2	Frequency Domain Filtering	157
5.2.1	Frequency Domain Filter Designed with Frequency Transformation Method	161
5.2.2	Frequency Domain Filter Designed with Frequency Sampling Method	161
5.2.3	Frequency Domain Filter Designed with Windowing Method .	161
5.2.4	Gaussian Lowpass Filter	164
5.2.5	Butterworth Lowpass Filter	164
5.3	Wiener Lowpass Filter	164
5.4	Wavelet Denoising	169
5.5	Wavelet Packet Denoising	169
5.6	Summary	169
6	Conclusions and Future Works	175
6.1	Conclusions	175
6.2	Further Work	179
A.1	Application of ANFIS to a Nonlinear Passage Dynamics of Order 3 to Restore an Image Corrupted by Gaussian Noise	182
A.1.1	Application of ANFIS with Different MFs	183
A.1.2	Discussion on Parameters of ANFIS	195
A.2	Application of ANFIS to a Nonlinear Passage Dynamics of Order 2 to Restore an Image Corrupted by Salt and Pepper Noise	207
A.2.1	Application of ANFIS with Different MFs	211
A.2.2	Discussion on Parameters of ANFIS	219
A.2.3	The Training Data and Checking Data	224
A.3	Application of ANFIS to a Nonlinear Passage Dynamics of Order 3 to Restore an Image Corrupted by Gaussian Noise	224
A.3.1	Discussion on Parameters of ANFIS	232
A.3.2	The Training Data and Checking Data	234

Appendix A—Part of results of image processing with ANFIS	182
Appendix B—Index	237
Appendix B—Index	240
Vita	242

List of Tables

4.1	<i>MSE between the original image and the restoration image with all types of MFs.</i>	110
4.2	<i>MSE between the original image and the restoration image with the bell MFs in the different training epochs number.</i>	113
4.3	<i>MSE of filtered noise images contaminated with the different number of bell MFs for each input.</i>	113
4.4	<i>MSE of filtered noise images by the backpropagation method and by using the hybrid learning algorithm contaminated with different intensity Gaussian noises.</i>	115
4.5	<i>MSE of filtered noise images by the output MF type as linear and constant contaminated with different intensity Gaussian noise.</i>	117
4.6	<i>MSE between the original image and the restoration image with all types of MFs for the nonlinear passage dynamics of order 3.</i>	132
4.7	<i>MSE between the original image and the restoration image with the bell MFs in the different training epoch numbers.</i>	132

4.8	<i>MSE of filtered noise images contaminated with the different number of bell MFs for each input.</i>	132
4.9	<i>MSE of filtered noise images by the backpropagation method and by using the hybrid learning algorithm contaminated with different intensity Gaussian noise.</i>	133
4.10	<i>MSE of filtered noise images by the output MF type as linear and constant contaminated with different intensity Gaussian noise.</i>	133
4.11	<i>MSE between the original image and the restoration image with all types of MFs.</i>	141
4.12	<i>MSE between the original image and the restoration image with the bell MFs in different training epoch numbers.</i>	141
4.13	<i>MSE of filtered noise images contaminated with the different number of bell MFs for each input.</i>	141
4.14	<i>MSE between the backpropagation method and the hybrid learning algorithm.</i>	142
4.15	<i>MSE of filtered noise images by the output MF type as linear and constant contaminated with salt & pepper noise.</i>	142
4.16	<i>MSE between the original image and the restoration image with all types of MFs in a nonlinear passive dynamics of order 3.</i>	145

4.17 <i>MSE between the original image and the restoration image with the bell MFs in the different training epoch numbers.</i>	145
4.18 <i>MSE of filtered noise images contaminated with the different number of bell MFs for each input.</i>	145
4.19 <i>MSE between the backpropagation method and the hybrid learning algorithm.</i>	146
4.20 <i>MSE of filtered noise images by the output MF type as linear and constant contaminated with salt & pepper noise.</i>	146
5.1 <i>MSE of noisy image contaminated with different types of noise filtered by the conventional techniques.</i>	173
5.2 <i>The elapsed time of filtered noisy image contaminated with different types of noise by ANFIS (bell MFs) and by the conventional techniques (Unit: second).</i>	174

List of Figures

2.1	<i>Decomposition of image function (redrawn from Pratt (2001)).</i>	17
2.2	<i>Point-source imaging system (redrawn from Pratt (2001)).</i>	18
2.3	<i>Point-source imaging system (redrawn from Pratt (2001)).</i>	19
2.4	<i>Effect of point spread function on image blurring.</i>	20
2.5	<i>Model of image degradation and restoration process (redrawn from Gonzalez and Woods (2001)).</i>	21
2.6	<i>A square subimage window R_{xy} of size 3×3 (redrawn from MATLAB manual).</i>	26
2.7	<i>An Alpha-trimmed filter: (a) Image corrupted by a combination of Gaussian and salt & pepper noise. (b) Image filtered with an alpha-trimmed mean filter size 3×3 and $\alpha=1/3$.</i>	29
2.8	<i>A SD-ROM filter: (a) An image corrupted by salt & pepper noise. (b) The result of filtering with a SD-ROM filter of size 3×3.</i>	32
2.9	<i>Lowpass filter transfer functions: (a) An ideal lowpass filter transfer function. (b) A Butterworth lowpass filter transfer function. (c) A Gaussian lowpass filter transfer function.</i>	34
2.10	<i>The 14th order lowpass filter frequency response with truncation frequency of 0.5 designed by frequency transformation method: (a) A one-dimensional frequency response . (b) A two-dimensional frequency response.</i>	35
2.11	<i>Lowpass filter frequency response designed by frequency sampling method: (a) Desired two-dimensional frequency response. (b) Actual two-dimensional frequency response.</i>	37

2.12 Lowpass filter frequency responses designed by windowing method: (a) Bartlett window; (b) Hamming window; (c) Hanning window; (d) Blackman window.	38
2.13 Perspective plot of Fourier transform (redrawn from MATLAB manual).	41
2.14 Perspective plot of STFT.	41
2.15 Perspective plot of Wavelet.	42
2.16 The contrast of different signal analysis methods.	43
2.17 Four types of wavelet functions: (a) Mexican hat wavelet function. (b) Gaussian wavelet function. (c) Meyer wavelet function. (d) Morlet wavelet function.	44
2.18 The different scales of wavelet function.	45
2.19 Calculating the coefficient by scaled, shifted versions of the wavelet function.	46
2.20 The perspective plot of a 2 level image decomposition (redrawn from Gonzalez and Woods (2001)).	47
2.21 Multiple level decomposition trees: (a) Three level decomposition tree. (b) Four level decomposition tree in a practical case (redrawn (a) from MATLAB manual).	47
2.22 Two level decomposition and reconstruction of an image (redrawn from MATLAB manual).	47
2.23 The original image and its decomposition at different levels: (a) Original image; (b) Decomposition at level 1; (c) Decomposition at level 2; (d) Decomposition at level 3.	48
2.24 A two level decomposition tree with wavelet packet (redrawn from MATLAB manual).	49
2.25 A three level decomposition and reconstruction for an image.	51
2.26 The curves of (a) original signal; (b) hard threshold; (c) soft threshold.	52
2.27 Block diagram for pure FIS (redrawn from Wu and Ling (2002)).	53
2.28 Block diagram for Mamdani FIS (redrawn from Wu and Ling (2002)).	54
2.29 Block diagram for Sugeno FIS (redrawn from Wu and Ling (2002)).	56

2.30	<i>The architecture of ANFIS for two-input first order Sugeno fuzzy model with two rules (redrawn from Jang, Sun and Mizutani (1997)).</i>	61
3.1	<i>A 2 dimensional image matrix with M rows and N columns (redrawn from Bose (2003)).</i>	67
3.2	<i>A 1 dimensional sequence vector after a 2 dimensional image matrix is connected with the row (or columns) (redrawn from Bose (2003)).</i>	68
3.3	<i>Schematic diagram of adaptive noise cancellation without ANFIS (redrawn from Jang, Sun and Mizutani (1997)).</i>	71
3.4	<i>The architecture of ANFIS for noise cancellation (redrawn from Jang, Sun and Mizutani (1997)).</i>	72
4.1	<i>The original image: (a) The original color image; (b) The original gray image.</i>	79
4.2	<i>The signals being used in ANFIS: (a) The measurable source noise $v_1(i)$; (b) The distorted noise $v_0(i)$; (c) The estimated distorted noise $y(i)$ by ANFIS; (d) The error between the estimated distorted noise $y(i)$ by ANFIS and the distorted noise $v_0(i)$.</i>	81
4.3	<i>Spectral density distributions.</i>	82
4.4	<i>The characteristics of ANFIS function \hat{f}.</i>	82
4.5	<i>The estimated errors with ANFIS corrupted by low, medium and high Gaussian noise.</i>	83
4.6	<i>RMSE curves and training step size of bell MFs: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.</i>	84
4.7	<i>The changes in bell MFs before and after training with low Gaussian noise.</i>	85
4.8	<i>The changes of bell MFs before and after training with medium Gaussian noise.</i>	86
4.9	<i>The changes of bell MFs before and after training with high Gaussian noise.</i>	87

4.10 The image corrupted by different intensity noises and the results of removing noises with ANFIS of bell MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	88
4.11 The changes in triangle MFs before and after training.	89
4.12 RMSE curves and training step size of triangle MFs: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.	90
4.13 The image corrupted by different intensity noises and the results of removing noises with ANFIS of triangle MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	91
4.14 The changes in Gaussian MFs before and after training.	92
4.15 RMSE curves and training step size of Gaussian MFs: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.	93
4.16 The image corrupted by different intensity noises and the results of removing noises with ANFIS of Gaussian MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	94
4.17 The changes in the two-sided Gaussian MFs before and after training.	95
4.18 RMSE curves and training step size of two-sided Gaussian MFs: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.	96

4.19 The image corrupted by different intensity noises and the results of removing noises with ANFIS of the two sided Gaussian MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	97
4.20 The changes in product of two sigmoid MFs before and after training.	98
4.21 RMSE curves and training step size of product of two sigmoid MFs: (a), (c) and (e) RMSE curves to low, medium and high noises; (b), (d) and (f) The training step size for low, medium and high noises.	99
4.22 The image corrupted by different intensity noises and the results of removing noises with ANFIS of the product of two sigmoid membership functions: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	100
4.23 The changes in the trapezoidal MFs before and after training.	101
4.24 RMSE curves and training step size of trapezoidal MFs: (a), (c) and (e) RMSE curves to low, medium and high noises; (b), (d) and (f) The training step size for low, medium and high noises.	102
4.25 The image corrupted by different intensity noises and the results of removing noises with ANFIS of the trapezoidal MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	103
4.26 The changes in the difference between two sigmoidal MFs before and after training.	104
4.27 RMSE curves and training step size of difference between two sigmoidal MFs: (a), (c) and (e) RMSE curves to low, medium and high noises; (b), (d) and (f) The training step size for low, medium and high noises.	105

4.28 The image corrupted by different intensity noises and the results of removing noises with ANFIS of the difference between two sigmoidal MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	106
4.29 The changes in the pi-shaped MFs before and after training.	107
4.30 RMSE curves and training step size of pi-shaped MFs: (a), (c) and (e) RMSE curves to low, medium and high noises; (b), (d) and (f) The training step size for low, medium and high noises.	108
4.31 The image corrupted by different intensity noises and the filtered images with ANFIS of the pi-shaped MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	109
4.32 The RMSE of ANFIS removing high Gaussian noise with the bell MFs for the different training epoch numbers: (a) 50 epochs; (b) 100 epochs; (c) 200 epochs; (d) 400 epochs.	111
4.33 The change of step size curve in the training processing: (a) 50 epochs; (b) 100 epochs; (c) 200 epochs; (d) 400 epochs.	112
4.34 The image corrupted by the high Gaussian noise and the results of removing noise with ANFIS of the bell MFs for the different training epoch numbers: (a) The image with high noise; (b) 20 epochs; (c) 50 epochs; (d) 100 epochs; (e) 200 epochs; (f) 400 epochs.	114
4.35 The changes in 3 MFs for each input before and after training.	115
4.36 The changes in 4 MFs for each input before and after training.	116
4.37 The changes in 5 MFs for each input before and after training.	117
4.38 The changes in 6 MFs for each input before and after training.	118

4.39 The RMSE of ANFIS removing high Gaussian noise with the different number of bell MFs for each input: (a) 3 MFs; (b) 4 MFs ; (c) 5 MFs; (d) 6 MFs.	119
4.40 The images corrupted by the high Gaussian noise and the results of removing noise with ANFIS in the different number of the bell MFs for each input: (a) The image with high noise; (b) 2MFs; (c) 3MFs; (d) 4MFs; (e) 5MFs; (f) 6MFs.	120
4.41 RMSE curves and training step size of backpropagation method: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.	121
4.42 The images corrupted by different intensity noises restored by the back-propagation method and by a hybrid learning algorithm with ANFIS of the bell MFs: (a), (c) and (e) Backpropagation method restored from low, medium and high noise respectively; (b), (d) and (f) Hybrid learning algorithm restored from low, medium and high noise respectively. .	122
4.43 RMSE curves and training step size of the output MF type as constant: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.	123
4.44 The images corrupted by the different intensity noise restored by the output MF type of constant and linear with ANFIS of the bell MFs: (a), (c) and (e) using the output MF type of constant restored from low, medium and high noise respectively; (b), (d) and (f) using the output MF type of linear restored from low, medium and high noise respectively.	124
4.45 The changes in RMSEs of the training data and the checking data. . .	125
4.46 The changes in the MFs of the training data and checking data before and after training.	126
4.47 Spectral density distributions: (a) Information signal $s(i)$; (b) Source noise signal $v_1(i)$; (c) Distorted noise signal $v_0(i)$; (d) Measurable output signal $x(i)$	129

4.48	<i>The changes in bell MFs before and after training.</i>	130
4.49	<i>The image corrupted by different intensity noises and the results of removing noises with ANFIS of bell MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.</i>	131
4.50	<i>The original image: (a) The original color image; (b) The original gray image.</i>	135
4.51	<i>The signals being used in ANFIS: (a) The measurable source noise $v_1(i)$; (b) The distorted noise $v_0(i)$; (c) The estimated distorted noise $y(i)$ by ANFIS; (d) The error between the estimated distorted noise $y(i)$ by ANFIS and the distorted noise $v_0(i)$.</i>	136
4.52	<i>Spectral density distributions.</i>	137
4.53	<i>The characteristics of ANFIS function \hat{f}.</i>	137
4.54	<i>The estimated errors with ANFIS corrupted by high salt & pepper noise.</i>	138
4.55	<i>The image corrupted by salt & pepper noises and the results of removing noises with ANFIS of bell MFs: (a) The image contaminated with low salt & pepper noise ($PDF=0.1008$); (b) The restored image; (c) The image contaminated with medium salt & pepper noise ($PDF=0.1812$); (d) The restored image; (e) The image contaminated with heavy salt & pepper noise ($PDF=0.3017$); (f) The restored image.</i>	140
4.56	<i>Spectral density distributions.</i>	143
4.57	<i>The image corrupted by salt & pepper noise and the results of removing noise with ANFIS of bell MFs: (a) The image contaminated with salt & pepper noise ($PDF=0.2793$); (b) The restored image.</i>	144

5.1	<i>The images filtered with arithmetic mean filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by: (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.</i>	149
5.2	<i>The images filtered with geometric mean filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.</i>	150
5.3	<i>The images filtered with harmonic mean filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.</i>	151
5.4	<i>The images filtered with contraharmonic mean filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.</i>	153
5.5	<i>The images filtered with median filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.</i>	154
5.6	<i>The images filtered with max filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.</i>	155

5.7 The images filtered with min filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with min filter corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (g) Salt & pepper noise.	156
5.8 The images filtered with midpoint filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.	158
5.9 The images filtered with alpha-trimmed mean filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.	159
5.10 The images filtered with SD-ROM filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.	160
5.11 The images filtered with frequency transformation method corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.	162
5.12 The images filtered with frequency sampling method corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.	163

5.13 The images filtered with windowing method corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.	165
5.14 The images filtered with Gaussian lowpass filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.	166
5.15 The images filtered with Butterworth lowpass filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.	167
5.16 The images filtered with Wiener lowpass filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.	168
5.17 The images filtered with soft threshold in wavelet corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.	170
5.18 The images filtered with soft threshold in wavelet packet corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.	171

A.1	Spectral density distributions. (a) Information signal $s(i)$; (b) Source noise signal $v_1(i)$; (c) Distorted noise signal $v_0(i)$; (d) Measurable output signal $x(i)$	184
A.2	The changes of bell MFs before and after training.	185
A.3	The images corrupted by the different intensity noise and the results of removing noise with ANFIS of bell MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	186
A.4	The changes of triangle MFs before and after training.	187
A.5	The images corrupted by the different intensity noise and the results of removing noise with ANFIS of triangle MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	188
A.6	The changes of Gaussian MFs before and after training.	189
A.7	The images corrupted by the different intensity noise and the results of removing noise with ANFIS of Gaussian MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	190
A.8	The changes of the two-sided Gaussian MFs before and after training.	191
A.9	The images corrupted by the different intensity noise and the results of removing noise with ANFIS of the two sided Gaussian MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise. . . .	192
A.10	The changes of the trapezoidal MFs before and after training. . . .	193

A.11 The images corrupted by the different intensity noise and the results of removing noise with ANFIS of the trapezoidal MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	194
A.12 The changes of the product of two sigmoid MFs before and after training.	195
A.13 The images corrupted by the different intensity noise and the results of removing noise with ANFIS of the product of two sigmoid membership functions. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	196
A.14 The changes of the difference between two sigmoidal MFs before and after training.	197
A.15 The images corrupted by the different intensity noise and the results of removing noise with ANFIS of the difference between two sigmoidal MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	198
A.16 The changes of the pi-shaped MFs before and after training.	199
A.17 The images corrupted by the different intensity noise and the filtered images with ANFIS of the pi-shaped MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.	200
A.18 The images corrupted by the high Gaussian noise and the results of removing noise with ANFIS of the bell MFs in the different training epoch number. (a) The image with high noise; (b) 20 epochs; (c) 50 epochs; (d) 100 epochs; (e) 200 epochs; (f) 400 epochs.	201
A.19 The changes of 3 MFs for each input before and after training.	202

A.20 The changes of 4 MFs for each input before and after training.	203
A.21 The images corrupted by the high Gaussian noise and the results of removing noise with ANFIS in the different number of the bell MFs for each input. (a) The image with high noise; (b) 2MFs; (c) 3MFs; (d) 4MFs.	204
A.22 The restoration images corrupted by the different intensity noise by the backpropagation method and by a hybrid learning algorithm with ANFIS of the bell MFs. (a), (c) and (e) Backpropagation method restored from low , medium and high noise respectively; (b), (d) and (f) Hybrid learning algorithm restored from low, medium and high noise respectively.	205
A.23 The restoration images corrupted by the different intensity noise by the output MF type of <i>constant</i> and <i>linear</i> with ANFIS of the bell MFs. (a), (c) and (e) using the output MF type of <i>constant</i> restored from low, medium and high noise respectively; (b), (d) and (f) using the output MF type of <i>linear</i> restored from low, medium and high noise respectively.	206
A.24 The changes of RMSEs of the training data and the checking data. .	207
A.25 The changes of the MFs of the training data and checking data before and after training.	208
A.26 The original image. (a) The original color image; (b) The original gray image.	209
A.27 The signals being used in ANFIS. (a) The measurable source noise $v_1(i)$; (b) The distorted noise $v_0(i)$; (c) The estimated distorted noise $y(i)$ by ANFIS; (d) The error between the estimated distorted noise $y(i)$ by ANFIS and the distorted noise $v_0(i)$	210
A.28 Spectral density distributions.	211
A.29 The characteristics of ANFIS function \hat{f}	212
A.30 The estimated errors with ANFIS corrupted by high salt & pepper noise.	213

A.31 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of bell MFs. (a) The image contaminated with low salt & pepper noise (PDF=0.1008); (b) The restored image; (c) The image contaminated with medium salt & pepper noise (PDF=0.1812); (d) The restored image; (e) The image contaminated with heavy salt & pepper noise (PDF=0.3017); (f) The restored image.	214
A.32 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the triangle MFs. (a) The image contaminated with salt & pepper noise; (b) The restored image.	215
A.33 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of Gaussian MFs. (a) The image contaminated with salt & pepper noise; (b) The restored image.	216
A.34 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the two-sided Gaussian MFs. (a) The image contaminated with salt & pepper noise (b) The restored image. . .	216
A.35 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the product of two sigmoidal MFs. (a) The image contaminated with salt & pepper noise (b) The restored image.	217
A.36 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the trapezoidal MFs. (a) The image contaminated with salt & pepper noise (b) The restored image. . .	218
A.37 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the difference of two sigmoidal MFs. (a) The image contaminated with salt & pepper noise (b) The restored image.	218
A.38 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the pi-shaped MFs. (a) The image contaminated with salt & pepper noise (b) The restored image.	219
A.39 The changes of 3 MFs for each input before and after training.	220
A.40 The changes of 4 MFs for each input before and after training.	221
A.41 The changes of 3 MFs for each input before and after training.	222

A.42 The changes of 4 MFs for each input before and after training.	223
A.43 The restoration images corrupted by salt & pepper noise by the back-propagation method and by a hybrid learning algorithm with ANFIS of the bell MFs. (a) Backpropagation method; (b) Hybrid learning algorithm.	223
A.44 The changes of RMSEs of the training data and the checking data. .	225
A.45 The changes of the MFs of the training data and checking data before and after training.	226
A.46 Spectral density distributions.	227
A.47 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of bell MFs. (a) The image contaminated with salt & pepper noise (PDF=0.2793); (b) The restored image.	228
A.48 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of Gaussian MFs. (a) The image contaminated with salt & pepper noise (PDF=0.2793); (b) The restored image. . .	229
A.49 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of triangle MFs. (a) The image contaminated with salt & pepper noise (PDF=0.2793); (b) The restored image. . .	229
A.50 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of two-sided Gaussian MFs. (a) The image contaminated with salt & pepper noise (PDF=0.2793); (b) The restored image.	230
A.51 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of trapezoidal MFs. (a) The image contaminated with salt & pepper noise (PDF=0.2793); (b) The restored image. . .	231
A.52 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the difference between two sigmoidal MFs. (a) The image contaminated with salt & pepper noise (PDF=0.2793); (b) The restored image.	231

A.53 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the product of two sigmoidal MFs. (a) The image contaminated with salt & pepper noise (PDF=0.2793); (b) The restored image.	232
A.54 The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the pi-shaped MFs. (a) The image contaminated with salt & pepper noise (PDF=0.2793); (b) The restored image.	233
A.55 The restoration images corrupted by salt & pepper noise by the backpropagation method and by a hybrid learning algorithm with ANFIS of the bell MFs. (a) Backpropagation method; (b) Hybrid learning algorithm.	234
A.56 The changes of RMSEs of the training data and the checking data. .	235
A.57 The changes of the MFs of the training data and checking data before and after training.	236

List of Symbols

$\delta(x, y)$	The two dimensional Dirac delta function
ε	An infinitesimally small limit of integration
(x, y)	Spatial coordinates
(ξ, η)	Coordinates in the $x - y$ plane
$f(x, y)$	A system input function
$f(\xi, \eta)$	The weighting factor of the impulse located at coordinates (ξ, η)
$\sigma(\cdot)$	The transfer function of imaging system
$h(x, y, \xi, \eta)$	The system response to the function σ or called the point spread function
$g(x, y)$	The convolution integral
$n(x, y)$	System noise
$\hat{f}(x, y)$	The optimal approximation for $f(x, y)$
H	$MN \times MN$ dimension matrix in discrete digital images
λ	Lagrange's multiplier
$m \times n$	The size of rectangular subimage window
R_{xy}	The set of coordinates in a rectangular subimage window of size
P	The order of the filter
α	The trimmed ratio in alpha-trimmed mean filter
$x(\mathbf{n})$	A 3×3 window center in Signal dependent rank order mean filter
$\mathbf{w}(\mathbf{n})$	An eight element observation vector of the neighboring pixels of $x(\mathbf{n})$ excluding $x(\mathbf{n})$ in the window
$\mathbf{r}(\mathbf{n})$	The elements of $\mathbf{w}(\mathbf{n})$ arranged in ascending order
$m(\mathbf{n})$	The rank-ordered mean

$\mathbf{d}(\mathbf{n})$	The rank-ordered differences
T_i	threshold values within ascending order ($i = 1, 2, 3, 4$)
(u, v)	Coordinates in frequency domain
$F(u, v)$	Fourier transform of the image to be denoised
$H(u, v)$	A filter transform function in frequency domain
$G(u, v)$	Image restoration function in frequency domain
$D(u, v)$	The distance from a point (u, v) to the origin of Fourier transform
$\epsilon(x(t))$	Expected operator for handling these stochastic variables
R_n	Autocorrelation of signal $x(t)$
P_n	Power spectrum of $x(t)$
Ψ	Wavelet function
s	Scale of the wavelet function
p	Shifted position of the wavelet function
$C(s, p)$	Wavelet coefficients
T_h	The hard threshold for denoising with wavelet
T_s	The soft threshold for denoising with wavelet
$O_{l,i}$	The output of the i th node in layer l in ANFIS
$s(i)$	The measurable information signal
$v_1(i)$	The unmeasurable noise source signal
$v_0(i)$	The unmeasurable distorted noise
$x(i)$	The output signal
$y(i)$	The output of ANFIS
$e(i)$	The recovered information signal $s(i)$

Preface

Images are an important source of information. The ultimate objective of image processing is to aid human beings understanding for the connotation of information. The research of digital image processing technique which is a comprehensive edge science concerns to the areas of optional system, micro-electronic technique, computer science and mathematical analysis. The ongoing increasing in scale of integration of electronics makes storage and computational power affordable to many applications. Image processing system also benefit from this trend and a variety of algorithm for image processing tasks becomes close at hand. This technique obtains significant achievement in recent 30 years.

Fuzzy logic, neural networks and genetic algorithm are the newest theoretical fundamental of artificial intelligence. A particular useful property of fuzzy logic technique is its capacity of represent knowledge in a way which is comprehensible to human interpretation. A distinct characteristic of neural networks is its ability of self study. The combination of these two techniques generates the adaptive neuro-fuzzy inference system (ANFIS). The fuzzy membership functions and fuzzy rules of ANFIS are obtained by the study of the given data rather than based on the experience or intuition. It is particularly important for the specially complicated system which properties are not full known.

An important application of ANFIS is cancelling noise. In signal processing and control, the measured signal often includes noise. When the noisy source is measurable, modelling for the noise is an effective method to cancel the noise and improve the precision of measured data. Therefore, we can use the learning property of ANFIS to grasp knowledge about the noise and avoid to blindly cancel the noise like

conventional filters. By the experiments, we find ANFIS really displays its more excellent ability for obtaining knowledge of noise and cancelling it than the conventional filters. Of course, we do not say ANFIS can replace the conventional filters to cancel noise in any cases. ANFIS also has some limits while showing its excellent ability for cancelling noise because it need more condition to meet than the conventional filters.

In this thesis, we broad the applications of ANFIS in one dimensional signal to two dimensional matrix signal — images. A simple introduction of the problem statement, objectives of this thesis, contributions and organization are in Chapter 1. Background and Literature Survey including the fundamental theory of image and ANFIS are introduced in Chapter 2. Its capability of adaptive cancelling Gaussian noise in the nonlinear passive dynamics of 2 and 3 in Chapter 4, ?? respectively. Its capability of adaptive cancelling salt & pepper noise in the nonlinear passive dynamics of 2 and 3 in Chapter ???. The image restoration by the conventional filters is shown in Chapter 5 for comparison with ANFIS. Chapter 6 summarizes the conclusions and the future work.

This thesis is a reflection of the significant progress that has been made in this field in just the past decades. Some important figures for comparison have been omitted to limit the size of thesis. We hope the content in this thesis can inspire your interests in this challenge area. The last but not least, welcome to any comments from the readers.

Hao Qin
Guelph, Ontario
April 2004

Chapter 1

Introduction

An image is a picture in which objects are mapped by a light system. If the imaging system is ideal, the mapping result will obtain an ideal image. This image can map all messages on the objects undistortedly. However, defects are present in all imaging systems. Potential image degradations include diffraction in the optical system, sensor nonlinearities, optical system aberrations, film nonlinearities, atmospheric turbulence effects, image motion blur, and geometric distortion. Noise disturbances may be caused by electronic imaging sensor, film granularity, and channel noise.

1.1 Problem Statement

Images are often corrupted by noise especially impulse noise due to a noisy channel or faulty image acquisition device. Much research has been done on removing noise. The aim is to suppress the noise while preserving the integrity of edges and details of image. With the applications and development of soft computing, more and more methods of fuzzy and neural network have been developed to solve the problems at the practical level.

For example, mean filtering is a linear technique to remove random Gaussian noise. However, while it removes the noise, it also removes too much detail and edge sharpness in image processing. More importantly, it is not an effective method to reduce impulse noise like salt & pepper.

Median filtering is a nonlinear technique that is known for its effectiveness in removing noise especially impulse noise while preserving edge sharpness in image

processing [?, ?, ?]. Even though median filters can suppress and remove noise from the contaminated images, too much signal distortion is introduced, and features such as sharp edge are lost. Also performance advantages by these conventional approaches can only be achieved when the occurrence probability of noise is small. Due to lack of adaptability, these variations of median filters cannot perform well when the probability is bigger than 0.2. Other conventional filtering methods all face the same problem.

Recently, adaptive systems based on neural networks [?, ?] or fuzzy theory [?, ?] with data-driven adjustable parameters have emerged as attractive alternatives. The methods such as the optimal detail-restoring stack filters [?], the hybrid FIR weighted order statistics filter [?], and adaptive average iterative filters [?] have been developed.

The proposed noise cancelling in degradation images with adaptive neuro-fuzzy inference system (ANFIS) is inspired by Widrow and Glover's [?] adaptive noise cancellation model for one dimensional signals, in which the objective is to filter out by identifying a nonlinear model between a measurable noise source and the corresponding unmeasurable interference in noise removing in one dimensional signals with J.-S. Jang's ANFIS.

1.2 Objectives of this Thesis

In ANFIS, neural networks exploit their structures with abundant theorems and efficient numerical training algorithms. They inset several input-output mappings on a black-box network of connection weights. During the training process, a neural network is trained with a sequence of randomly selected input samples, and values of connection weights are adjusted in accordance with the training algorithm. On the other hand, fuzzy systems can directly encode structured knowledge. Fuzzy systems may invariably store a series of common-sense rules linguistically articulated by an expert, or a fuzzy system may adaptively reason and change its fuzzy rules from representative symbols as well as numerical samples. Fuzzy systems and neural networks naturally combine. The combination generates an adaptive system in that the neural networks imbed in an overall fuzzy architecture, generating and refining fuzzy rules

from training data.

We extend this ANFIS method from a one dimensional signal to a two dimensional signal images. The objectives of this thesis are to explore if ANFIS can be used to remove noise generated by nonlinear model and what is its advantages and disadvantages in a two dimensional signal.

First, our objective is to investigate the image restoration contaminated with low, medium and high Gaussian noise in different nonlinear passage dynamics. We also inspect 8 types of membership function (MF) which are bell MF, triangle MF, Gaussian MF, two sided MF, pi-shaped MF, product of two sigmoidal MFs, difference of two sigmoidal MFs and trapezoidal MF. Other parameters such as the training epochs, the number of MFs for each input, the optimization method, the type of output MFs and the overfitting between the training data and checking data are also investigated. Second, our objective is to use ANFIS to restore the image corrupted by salt & pepper noise in these passage dynamics.

Our objectives also is to compare the effect of the image restoration by ANFIS with the conventional filters including spatial filters, frequency domain filters, adaptive optimal filter Wiener filter, and wavelet and wavelet packet. Spatial filters include arithmetic mean filter, geometric mean filter, harmonic mean filter, contraharmonic mean filter, median filter, max and min filters, alpha-trimmed mean filter, signal dependent rank order mean filter. In frequency domain filters, we will investigate ideal filter, Butter-worth lowpass filters and Gaussian lowpass filter and the design methods, such as frequency transformation method, frequency sampling method and windowing method. In wavelet, we will discuss the hard threshold and soft threshold selection rules for de-noising.

1.3 Contributions of this Thesis

The contributions of this thesis can be generalized as follows:

1. Adaptive noise cancellation with ANFIS was to process the signals which are one dimension. We broad this ANFIS method from one dimensional signal to

two dimensional signals — images.

2. First we extend linear adaptive noise cancellation into nonlinear area by using nonlinear adaptive systems. ANFIS can be used to recognize an unknown nonlinear passage dynamics which transits a noise source into an interference component in a detected signal. Sometimes the proposed nonlinear adaptive cancelling technique is more suitable to filter some noise component than noise cancelling techniques based on frequency-selective filtering.
3. Secondly we try the image restoration contaminated with noise by ANFIS. An information signal is unmeasurable and a noise source signal is measurable. The noise source goes through an unknown nonlinear function to generate a distorted noise. It is then added to information signal to compose a output signal which is measurable. Our goal is to recover the information signal from the compound output signal, which forms the information signal plus a distorted and delayed version, with ANFIS.
4. The noise generated by nonlinear passage dynamics of order 2 and 3 are investigated. We discuss the nonlinear passive dynamics of order 2 and 3 because they are typical in the corrupted images. For higher order components, their affection dramatically reduce with the increasing of delay time in the whole noise. Therefore, the nonlinear passive dynamics of order higher than 3 does not discussed.
5. We inspect 8 types of membership function (MF) which are bell MF, triangle MF, Gaussian MF, two sided MF, pi-shaped MF, product of two sigmoidal MFs, difference of two sigmoidal MFs and trapezoidal MF. We show the changes in MFs before and after training. RMSE (root mean square error) and the change in step size curve in the training processing are also displayed in figures.
6. Other parameters such as the training epochs, the number of MFs for each input, the optimization method, the type of output MFs and the overfitting of

the training data and checking data are also investigated.

7. We try to use ANFIS to restore the image corrupted by not only Gaussian noise with low, medium and high intensity but also salt & pepper noise with low, medium and high probability in these passage dynamics.
8. The behavior of the information signal, the noise source, the distorted noise, and the restored information signal in the frequency domain is also investigated. We inspect the spectral density distributions of them to know overlap each other in a large frequency area and to see if the common filtering methods can be used to remove these kinds of noise.
9. We compare the effect of the image restoration by ANFIS with the conventional filters including spatial filters, frequency domain filters, adaptive optimal filter Wiener filter, and wavelet and wavelet packet. Spatial filters include arithmetic mean filter, geometric mean filter, harmonic mean filter, contraharmonic mean filter, median filter, max and min filters, alpha-trimmed mean filter, signal dependent rank order mean filter. In frequency domain filters, we investigate ideal filter, Butter-worth lowpass filters and Gaussian lowpass filter and the design methods, such as frequency transformation method, frequency sampling method and windowing method. In wavelet, we discuss the hard threshold and soft threshold selection rules for de-noising.

1.4 Organization of this Thesis

The thesis is composed of six chapters. The structure is listed as follows:

A simple introduction of the problem statement of ANFIS, objectives of this thesis, contributions and organization are in Chapter 1.

In chapter 2, an introduction about the basic knowledge of image restoration models is addressed. we mainly introduce three aspects about image restoration models: Mathematical expression of imaging system, models of image degradation, and summary of image restoration methods. In addition, we introduce the principles

of the conventional filters including spatial filters, frequency domain filters, adaptive optimal filter — Wiener filter and wavelet and wavelet packet denoising. Spatial filters include mean filters, order-statistics filters. In frequency domain filters, we investigate ideal filter, Butterworth lowpass filters and Gaussian lowpass filter and the design methods, such as frequency transformation method, frequency sampling method and windowing method. In wavelet, we discuss the hard threshold and soft threshold selection rules for denoising. Chapter 2 also introduces the basic concepts and principles of fuzzy and neural networks.

In Chapter 3, We give out proposed method for two demensional image denoising with ANFIS algorithm. We introduce transformation from 2-D image matrix to 1-D sequence vector. Then we describe how to generate and add noises into the processed images. After that, the architecture and algorithm of nonlinear adaptive noise cancellation with ANFIS is introduced. Subsequently, we present restoration of images corrupted by the noise of the nonlinear passage dynamics with ANFIS. Finally, transformation from 1-D sequence vector back to 2-D image matrix is illustrated.

In Chapter 4, we present the experiment results and some analysis of nonlinear passage dynamics of order 2 in images corrupted with Gaussian noise. We inspect 8 types of membership function (MF) which are bell MF, triangle MF, Gaussian MF, two sided MF, pi-shaped MF, product of two sigmoidal MFs, difference of two sigmoidal MFs and trapezoidal MF. Other parameters such as the training epochs, the number of MFs for each input, the optimization method, the type of output MFs and the overfitting between the training data and checking data are also investigated. We also simply present the experiment results and some analysis of nonlinear passage dynamics of order 3 in images corrupted with Gaussian noise. Similarly, we inspect 8 types of MFs and other parameters like the discussion in nonlinear passage dynamics of order 2. Finally, we simply present the experiment results and some analysis of two nonlinear passage dynamics of order 2 and 3 in images corrupted with salt & pepper noise. We also inspect the same 8 types of MFs and other parameters as the investigation in two nonlinear passage dynamics of order 2 and 3 in images corrupted with Gaussian noise.

In Chapter 5, we present the experiment results of the image restoration with the conventional filters including spatial filters, frequency domain filters, adaptive optimal filter — Wiener filter and wavelet and wavelet packet denoising. Spatial filters include arithmetic mean filter, geometric mean filter, harmonic mean filter, contraharmonic mean filter, median filter, max and min filters, alpha-trimmed mean filter, signal dependent rank order mean filter. In frequency domain filters, we investigate ideal filter, Butterworth lowpass filters and Gaussian lowpass filter and the design methods, such as frequency transformation method, frequency sampling method and windowing method. In wavelet, we discuss the hard threshold and soft threshold selection rules for denoising.

Conclusions are summarized and recommendations for future work are presented in Chapter 6.

Chapter 2

Background and Literature Survey

In today's increasingly technological society, many new techniques are creatively merged and employed in new areas, such as the application of soft computing technique in image processing.

2.1 Literature Survey

Prof. Lotfi A. Zadeh [?] first presented the concept of soft computing , which is an innovative path for building computationally intelligent system in 1991. The fundamental thought underlying soft computing in its basic soul can be traced back to his many earlier publications, such as his 1965 paper on fuzzy sets [?]; the 1973 paper on the analysis of complex systems and decision processes [?]; and the 1979 report (1981 paper) on possibility theory and soft data analysis [?]. Neural computing and genetic computing in soft computing appear within the recent 2 decades.

Soft computing is an emerging approach to computing which parallels the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision [?].

The reason why soft computing differs from common hard computing is that unlike hard computing, soft computing is tolerant of imprecision, uncertainty, partial truth, and approximation. In fact, the prototype for soft computing inspire from the human intelligence. The basic principle of soft computing is to probe into the tolerance

for imprecision, uncertainty, partial truth, and approximation to realize robustness, tractability and low solution cost [?].

Prof. Zadeh outlines the differences between hard computing which is based on binary logic, crisp systems, numerical analysis and crisp software, and soft computing which is based on fuzzy logic, neural networks and probabilistic reasoning. Hard computing has the properties of precision and categoricity. On the other hand, soft computing has the specialties of approximation and imprecision. Hard computing require the properties of precision and certainty, however, the objective of soft computing is to explore the tolerance for imprecision and uncertainty to obtain high machine intelligence, tractability, lower cost.

Under these guidelines, the principal compositions of soft computing are fuzzy logic, neural computing, evolutionary computation, machine learning and probabilistic reasoning, with the latter including belief networks, chaos theory and parts of learning theory. However, one more thing we need emphasize is that soft computing is not a melange. Contrarily, each element of soft computing is a cooperation relationship and each of the partners gives a distinct and precise methodology for declaring problems in its field. From this perspective, the principal methodologies of the compositions in soft computing are complementary instead of competitive.

The complementary nature of fuzzy systems, neural networks, evolutionary computation, and probabilistic reasoning results in an important fact: in many situations the problem can be solved much more effectively by using the component of soft computing in combination rather than in exclusion. A well-known example of an effective combination is what has been known as neuro-fuzzy systems [?, ?]. Such systems are becoming progressively noticeable in a wide scope of consumer products ranging from washing machines to photocopiers, and from air conditions to video cameras. Neuro-fuzzy systems in industrial application are maybe more important even though they are less noticeable. The applications of soft computing techniques leads high machine intelligence quotient to a dramatic increasing in both consumer products and industrial systems in last two decades. To a large extent, the principal factor that results in the rapid growth in the number and variety of applications of

soft computing is the high machine intelligence quotient of soft computing systems.

Because the soft computing is not to seek perfect solutions but optimal and competitive ones instead, the practical applications can benefit from the combination of various methodologies [?] in soft computing. These interrelated yet distinguishing methodologies are currently attracting a great deal of attention and have already discovered many practical applications from industrial process control to face recognition, and from fault diagnosis to character recognition. The basic contribution of fuzzy logic is used to find a fundament for approximate reasoning, while neural network theory provides an effective methodology for learning from input data, and genetic algorithm systems supply computationally effective techniques for tuning the parameters used in fuzzy logic and neural networks.

We get the inspiration from the structure of soft computing that it should not only be applied in fuzzy logic, neurocomputing and genetic algorithm separately but also in all the cooperative methodologies even though we need not use them in the same degree.

We also get inspire from the successful applications of soft computing and the rapid growth of soft computing that the effects of soft computing will be felt dramatically increasingly in future. Soft computing is not only likely to act as an important role from consumer products to industrial systems but also its influence may finally spread much far-reaching. If we view from many aspects, we can find that soft computing describes a typical change in the objectives of computing, which reflects the fact that the human intelligence has a remarkable ability to store and deal with message which is generally imprecise, uncertain and lacking in categoricity. Viewing from this point, the important fact is that it gives a stage for the advancement of soft computing which reduces obstacles between the components of soft computing.

2.1.1 Image Processing

Since ancient time, vision in general and image in particular have acted as an especially important and essential role in human life. Images are obtained by using

different forms and means to observe the objective world in all kinds of observing systems. They generate the entities of visual perception by acting to eyes. Scientific research and statistics indicate that about 75% information which humans obtain from outside comes from visual system. That means most of message obtains from images.

Image can be classified into two category:

- Analog image is also called continuous image and includes optical image, photo image and video image.
- Digital image change continuous analog image by discrete processing into matrix image which can be identified by computers.

Digital image have the following prominent merits:

- High precision;
- Convenient processing;
- excellent repetition.

Image processing is to change images into a numeral matrix for storing in computers and process them using algorithms. In recently, the techniques of image processing are paid attention in many practical fields and obtain huge achievements. We can divide image processing into different branches according to different application areas.

The important branches have:

- Image digitization: By sampling and numerating, it changes analog images digital form for computer processing.
- Image enhancement and restoration: Its main goal is to enhance the useful information and reduce interruption and noise, making images clear or change their form for fitting to human and machine's analysis.
- Image coding: Under fidelity, it codes image information and compress image data and discard the redundant data

- Image segmentation and characteristic selection: It divide images into nonoverlapping regions and gather the objects from background.
- Image analysis: It segments, classifies, recognizes describes and explains the different objects.
- Image hiding: It hides media message, such as watermark.

The above contents of image processing are correlated. A practical image processing system usually combines several kinds of techniques of image processing for the required results. For example, image digitization is the first stage of image processing. Image coding is to transmit and store images. Image enhancement and restoration is usually the final objective for image processing. It can also acts as the preparation for the further image processing. Image characteristic obtained by image segmentation can be the final results or the fundamant for the further analysis.

Image processing relates to abroad knowledge and is complicated. For example, the basic theory of image coding is the combination of informatics and abstract mathematics. Image recognition need have the knowledge of statistics and signal processing. Some subjects need more professional knowledge, such as wavelet transformation, neural networks and morphologic theory. Also Image processing must suit to the development of computer. For example, because Intel CPU increase MMX technique, it benefits to Fourier transforation which is the common methodology in image processing.

Another property of image processing that also is the difficulty is that the algorithm is highly correlated to the content of processed objectives. It is difficult to find a general methodology for adapting all kinds of cases.

2.1.2 Image Processing with Soft Computing

Currently, the area of image processing has numerous applications, which all result from the interaction between fundamental scientific research on the one hand and the development of new technology on the other hand. The scientific community

became familiar to soft computing techniques in image processing quite recently. Soft Computing consists of several new components, such as fuzzy set theory, neural networks and genetic algorithms. In the past years, these techniques have proven their richness and power in the field of image processing. This fact is evident from the increasing number of publications and patents, and the fact that image processing is definitely mentioned as a topic in most soft computing in international conferences.

Image processing is a very important scientific research field which has a broadening range of applications. The development of new perceptions and applications is inspired by both basic scientific research and the development of new techniques. Soft computing, which is a generic term for a specific collection of tools to model uncertainty, imprecision, evolutionary behavior and complex models, is a facet of one of these emerging techniques. Soft computing is particularly suited to deal with extremely complex problems, in which traditional mathematics leads to cumbersome solutions in computation.

Because soft computing which is an emerging field that consists of complementary elements of fuzzy logic, neural computing, evolutionary computation and machine learning is characterized by a strong learning and cognitive ability and good tolerance of imprecision and uncertainty, soft computing techniques have found wide applications in society. Needless to say, image processing is one of the applications. The areas in which soft computing is a factor in image processing include but are not limited to:

- Noise reduction in images [?, ?, ?, ?];
- Edge detection and segmentation [?, ?, ?, ?, ?];
- Feature extraction;
- Image enhancement [?, ?];
- Image compression [?];
- Similarity between images;

- Image analysis;
- Medical imaging;
- Visualization;
- Fuzzy wavelets.

Viewing both from the theoretical and practical point, the application of soft computing techniques in image processing area has led to numerous technical advances. The deepening of the synergistic cooperation between these two fields will certainly produce even more groundbreaking methods that have the ability to solve complex real world problems. The objective is to exhaust soft computing techniques for diverse aspects of image processing.

2.1.3 Restoration of Images Corrupted by Noise with Soft Computing

Recently, adaptive systems established on neural networks [?, ?] or fuzzy theory [?, ?] with adjustable parameters trained by input data have emerged as attractive alternatives. The methods like the optimal detail-restoring stack filters [?], the hybrid FIR weighted order statistics filter [?], and adaptive average iterative filters [?] have been developed.

Yu and Lee used the adaptive fuzzy median filter [?] with the backpropagation algorithm [?] tuning a set of randomly given initial membership functions even though Currently, hybrid neuro-fuzzy networks do not represent general means in restoring images corrupted by impulsive noise. Their results show that usage of adaptive fuzzy median filter achieves better filtering than traditional median filter when the probability is greater than 0.3.

Tsai et al. [?] present a multichannel filter called neuro-fuzzy hybrid multichannel filter to simultaneously achieve three objectives: noise attenuation, edge or detail preservation and chromaticity retention. This is described by a set of fuzzy rules

such that it is capable of effectively fusing the useful filtering advantages from vector median, vector directional, and identity filter to improve the filtering function of the conventional filters. They exploit the functional equivalence between fuzzy inference systems and radial-basis function networks on the optimal design of neuro-fuzzy hybrid multichannel filters.

Castellanos et al. present a new approach, which is based on an adaptive fuzzy leader clustering network, to nonlinear filtering that can remove speckle noise without noticeable degradation in [?]. It is a neuro-fuzzy clustering algorithm which can be used to cluster noise pixels in the image separately. A search is performed throughout the image to detect the noise pixels and to eliminate them using a spatial technique same as the traditional median filter after the clustering process is completed.

In [?], Wang et al. present an alterative approach by using histogram adaptive fuzzy filtering to recover the images corrupted by noise using fuzzy spatial filtering. It is fulfilled by a fuzzy smoothing filter which is constructed from a set of fuzzy membership functions for the initial parameters derived in according to input histogram. A principle of conservation in histogram potential is combined with input statistics to adjust the initial parameters so that it minimizes the difference between a reference intensity and the output of defuzzification process.

Their simulation results have shown that adaptive neuro-fuzzy filtering successfully achieves better filtering than traditional filtering methods when images are corrupted with heavy noise. However, they all have not explored the case in which an interference component is filtered out by identifying a nonlinear model between a measurable noise source and the corresponding unmeasurable interference in noise removing in images with adaptive neuro-fuzzy inference system.

2.2 Image Restoration Models

Image restoration may be thought as an estimation process in which operations are performed on an observed or measured image field to estimate the ideal image that would be observed if no image degradation were present in an imaging system. In

this chapter we introduce image restoration as follows:

- Mathematical expression of imaging system;
- Models of image degradation;
- Summary of image restoration methods.

2.2.1 Mathematical Expression of Imaging System

Based on research of the causes of image degradation, image restoration is to design an operation depending on the degrading image and pre-knowledge, and finally to estimate the ideal product. In general, when we obtain a digital image, we always deal with it using the techniques of image restoration and then enhance it. Because there are different reasons of degradation in different areas, we should adopt different restoration methods for the same image. The methods should be appropriate for the cause of degradation.

For purposes of representation, the imaging system usually is regarded as a linear system. The general mathematical equations are deduced to the relations between the inputs of objects and the outputs of image. Then the model of degradation is set for imaging system and the techniques of restoration are studies on this foundation. In fact, the imaging system has, to some extent, a nonlinear quality. However, if the nonlinear property does not cause serious error, or when the imaging system satisfies the linear property in small extent, the imaging system is regarded as linear. Because the linear system has an integrate theoretical system, it is convenient and can show the main characteristics of the imaging system.

From the linear property of imaging system, singular operators are widely used in the analysis of an imaging system. The Dirac delta function [?] is a singular operator which has the following properties:

$$\int_{-\varepsilon}^{\varepsilon} \int_{-\varepsilon}^{\varepsilon} \delta(x, y) dx dy = 1 \quad \text{for } \varepsilon > 0 \quad (2.1a)$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \eta) \delta(x - \xi, y - \eta) d\xi d\eta = f(x, y) \quad (2.1b)$$

In Eq.(2.1a), ε is an infinitesimally small limit of integration; Eq.(2.1b) is called the filtering property of the Dirac delta function.

Known by the linear property of imaging system, the input function of system is decomposed into the summation of the infinite Dirac delta functions. The simplest Dirac delta function is δ function (impulse function). Therefore, the input function can be denoted as

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \eta) \delta(x - \xi, y - \eta) d\xi d\eta \quad (2.2)$$

From Eq.(2.2), the input function can be regarded as a sum of amplitude-weighted Dirac delta function.

If the output for a general linear system like one to one is defined as

$$g(x, y) = \sigma\{f(x, y)\} \quad (2.3)$$

then

$$g(x, y) = \sigma \left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \eta) \delta(x - \xi, y - \eta) d\xi d\eta \right\} \quad (2.4a)$$

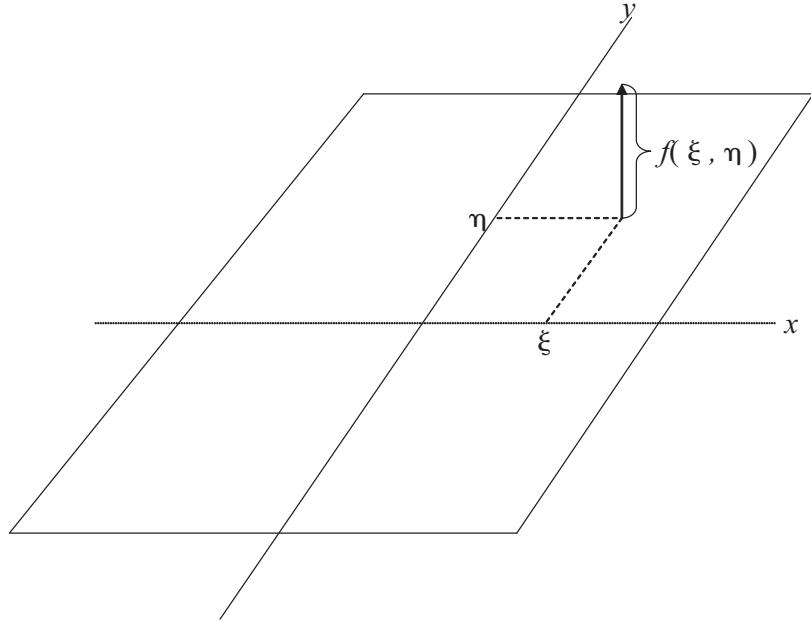


Fig. 2.1: Decomposition of image function (redrawn from Pratt (2001)).

or

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \eta) \sigma \{ \delta(x - \xi, y - \eta) \} d\xi d\eta \quad (2.4b)$$

in Eq.(2.4a) and (2.4b), σ is the transfer function of imaging system.

If the symbol $h(x, y, \xi, \eta)$ is expressed as the system response to σ function, the second term in the function of Eq.(2.4b)

$$h(x, y, \xi, \eta) = \sigma \{ \delta(x - \xi, y - \eta) \} \quad (2.5)$$

is called the impulse response of the imaging system. It is often denoted as the point spread function in an optical system. This is because the point of object will not be a point but an enlarged concentric circle after passing through the imaging system shown in Fig. 2.2. The relationship of input and output of the system is denoted as

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \eta) h(x, y, \xi, \eta) d\xi d\eta \quad (2.6)$$

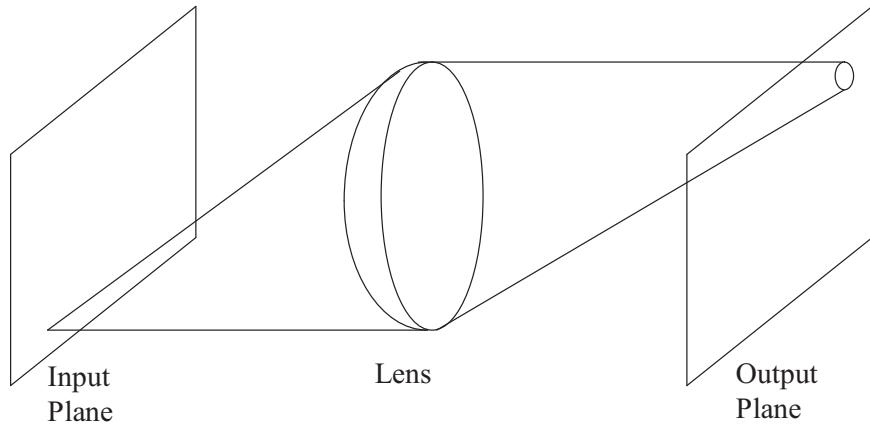


Fig. 2.2: *Point-source imaging system (redrawn from Pratt (2001)).*

Because the impulse response in optical imaging system only depends on $(x - \xi)$ and $(y - \eta)$, that is when point source moves in the object plane, its image only changes in location not in function form in the focal plane. Therefore we have

$$h(x, y, \xi, \eta) = h(x - \xi, y - \eta) \quad (2.7)$$

The function Eq.(2.6) can be transformed as

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \eta) h(x - \xi, y - \eta) d\xi d\eta \equiv f * h \quad (2.8)$$

where $*$ defines the convolution operation. The convolution integral is symmetric and given as follows:

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - \xi, y - \eta) h(\xi, \eta) d\xi d\eta \quad (2.9)$$

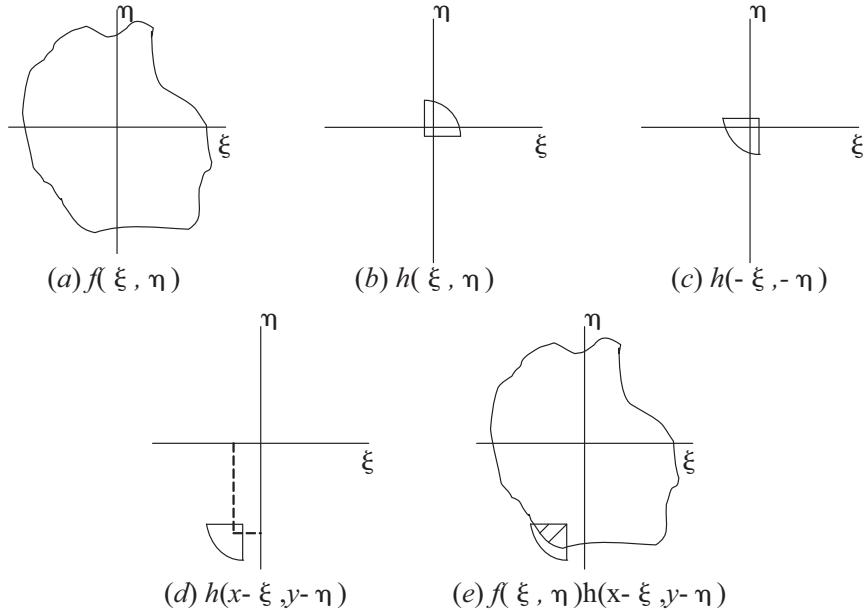


Fig. 2.3: *Point-source imaging system (redrawn from Pratt (2001)).*

Fig. 2.3 provides a symbolic form of the convolution process. The input function $f(x, y)$ and impulse response are drawn in the fake coordinate system (ξ, η) in Fig. 2.3(a) and (b). Then the coordinates of the impulse response are overturned, and the impulse response is compensated by the spatial values (x, y) in Fig. 2.3(c) and (d). The integrand product of the convolution integral of Eq.(2.8) is shown as a hatched region in Fig. 2.3(e).

Eq.(2.8) is the mathematical expression representing imaging system in recent optics. The convolution of the object function $f(x, y)$ and the impulse response is the image after diffraction effect is taken into consideration. The stronger the diffraction effect, the more degradation. Thus the point spread function decides the quality of imaging. This is illustrated in the three images shown in Fig. 2.4. Fig. 2.4(a) is the clear original image in real scene. Fig. 2.4(b) and Fig. 2.4(c) display the images of

the convolutions of the original image with point spread function of vertical motion and horizontal motion respectively.

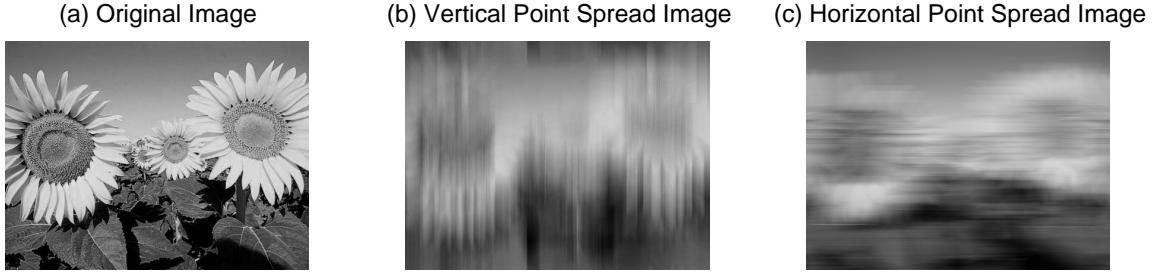


Fig. 2.4: *Effect of point spread function on image blurring.*

We can deduce the models of image degradation, and then recover the original image according to some standard after we obtain the mathematical expression of imaging system. This is an estimation to find an accurate solution of original image from degradation image. There are many standards for estimating effect, such as least square method (LSM). Image restoration is the reverse estimation of the process of image degradation, which is a process of deconvolution.

2.2.2 Image Degradation Models

Image restoration processing is built on the foundation of the mathematical model of image degradation. The model should reflect the reasons of image degradation. The causes of degradation usually are regarded as a factor of linear degradation in image processing because the reasons of image degradation are varied and complex but can be analyzed one by one. Then the approximated system degradation model is built.

Continuous Degradation Model

Assume $g(x, y)$ is a degradation image, $f(\xi, \eta)$ is the original image. The degradation model is presented in Fig. 2.5.

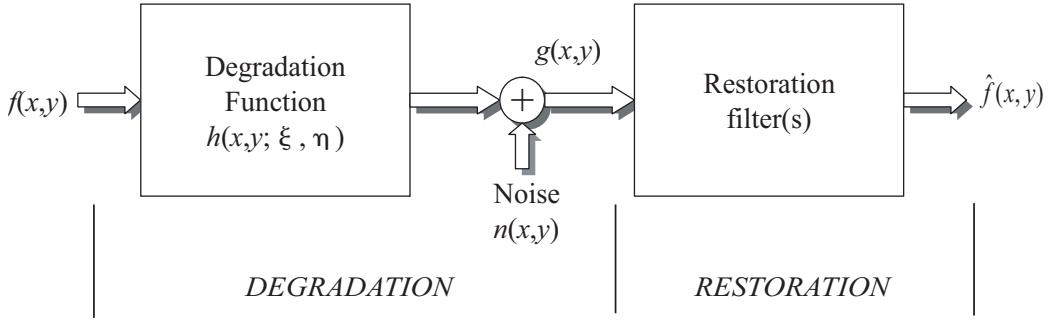


Fig. 2.5: *Model of image degradation and restoration process (redrawn from Gonzalez and Woods (2001)).*

$n(x, y)$ is system noise. The equation of image degradation model is

$$\begin{aligned} g(x, y) &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi, \eta) h(x - \xi, y - \eta) d\xi d\eta + n(x, y) \\ &= f(x, y) * h(x, y) + n(x, y) \end{aligned} \quad (2.10)$$

From Fig. 2.5, image degradation is the degradation of imaging system adding the additive system noise. In this model, image restoration reversely operates to obtain the optimal approximation $\hat{f}(x, y)$ for $f(x, y)$ if $h(x, y)$ and $n(x, y)$ are given from image degradation. There are two reasons why $\hat{f}(x, y)$ is the optimal approximation $f(x, y)$ but not the real $f(x, y)$: One reason is the equation has no solution in reverse operation and this is the most stubborn problem — singularity; Another reason is there exist multiple solutions in reverse operation. Both in two cases are called morbidity of image restoration.

Discrete Degradation Model

Because digital images are discrete, Eq.(2.10) is represented as discrete form in application shown as follows

$$g(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n) + n(x, y) \quad (2.11)$$

where $x=0, 1, 2, \dots, M-1$; $y=0, 1, 2, \dots, N-1$. The functions of $f(x, y)$ and $h(x, y)$ are the periodic functions of M and N . If the period of these two functions

are not M and N , they need extend zero padding to avoid overlapping of convolution period. $g(x, y)$ is the same periodic function as $f(x, y)$ and $h(x, y)$.

We stack the rows of $M \times N$ matrix functions $f(x, y)$, $g(x, y)$ and $n(x, y)$ as MN dimension column vectors and denote as f , g and n respectively.

$$f = \begin{bmatrix} f(0,0) \\ f(0,1) \\ \dots \\ f(0,N-1) \\ \dots \\ f(M-1,0) \\ f(M-1,1) \\ \dots \\ f(M-1,N-1) \end{bmatrix}; g = \begin{bmatrix} g(0,0) \\ g(0,1) \\ \dots \\ g(0,N-1) \\ \dots \\ g(M-1,0) \\ g(M-1,1) \\ \dots \\ g(M-1,N-1) \end{bmatrix}; n = \begin{bmatrix} n(0,0) \\ n(0,1) \\ \dots \\ n(0,N-1) \\ \dots \\ n(M-1,0) \\ n(M-1,1) \\ \dots \\ n(M-1,N-1) \end{bmatrix}$$

Eq.(2.11) can be written as

$$g = Hf + n \quad (2.12)$$

where H is $MN \times MN$ dimension matrix; H includes M^2 submatrix of which the size is $N \times N$ and is shown as

$$H = \begin{bmatrix} H_0 & H_{M-1} & H_{M-2} & \dots & H_1 \\ H_1 & H_0 & H_{M-1} & \dots & H_2 \\ H_2 & H_1 & H_0 & \dots & H_3 \\ \dots & \dots & \dots & \dots & \dots \\ H_{M-1} & H_{M-2} & H_{M-3} & \dots & H_0 \end{bmatrix} \quad (2.13)$$

Every submatrix H_j in Eq.(2.13) consists of j th row of $h(x, y)$

$$H_j = \begin{bmatrix} h(j,0) & h(j,N-1) & h(j,N-2) & \dots & h(j,1) \\ h(j,1) & h(j,0) & h(j,N-1) & \dots & h(j,2) \\ h(j,2) & h(j,1) & h(j,0) & \dots & h(j,3) \\ \dots & \dots & \dots & \dots & \dots \\ h(j,N-1) & h(j,N-2) & h(j,N-3) & \dots & h(j,0) \end{bmatrix} \quad (2.14)$$

2.2.3 Summary of Image Restoration Methods

From Eq.(2.13) and (2.14), we find H is a block cycling matrix. Every element is shown by point spread function. So H is also called the matrix of point spread function. From image degradation model expressed by Eq.(2.12), when g is given and some characteristics of H and n are known, restoration is to estimate f from

Eq.(2.12). Obviously, it is the reverse calculation process of degradation. If the noise is zero, the process of restoration is a process of deconvolution. However, in practice, if the values of M and N are large, then the matrix H will be very tremendous from Eq.(2.13) and (2.14) and the workload of calculation is too huge. Two methods can solve this problem: One method is to simplify diagonal block cycling matrix and then obtain the solution by existing fast algorithm. For example, diagonalize the block cycling matrix and then obtain the solution by discrete Fourier transform with Eq.(2.12). Another method is to analyze the reasons of degradation and find the function H in detail. For example, the function H for x direction motion blurring can be expressed as

$$h(x) = \begin{cases} T/a & 0 \leq x \leq a \\ 0 & otherwise \end{cases}$$

where a is denoted as the whole displacement in exposure time T .

2.2.4 Algebraic Methods for Image Restoration

Image restoration solves reverse transform with LSM if the point spread function and noise are given:

$$\hat{f} = \sigma^{-1}\{g\} \quad (2.15)$$

The goal of algebra image restoration method is to seek a closest approximation of input vector f . This approximation can make the value of preset rule least. The algebra unrestrained restoration method and algebra restrained restoration method is the two main algebra restoration methods.

Unrestrained Restoration Equation

From Eq.(2.12), it is obtained

$$n = g - Hf \quad (2.16)$$

the square of norm n is

$$\|n\|^2 = n^T n = (g - Hf)^T (g - Hf) \quad (2.17)$$

$\|n\|^2$ is regarded as a scale of noise. Unrestraint restoration is to obtain the solution of LSM in Eq.(2.17). That is to seek f and make $\|n\|$ least. Let

$$J(\hat{f}) = (g - Hf)^T(g - Hf) \quad (2.18)$$

unrestrained restoration method is transferred to solve the minimum of Eq.(2.18). Because \hat{f} does not have other restrained conditions except $J(\hat{f})$ being a minimum, this restoration method is called unrestrained restoration method. The minimum of Eq.(2.18) is

$$\frac{\partial J(\hat{f})}{\partial \hat{f}} = -2H^T(g - H\hat{f}) = 0 \quad (2.19)$$

then

$$\hat{f} = (H^T H)^{-1} \cdot H^T g \quad (2.20)$$

where $(H^T H)^{-1}$ is the generalized reverse matrix of H . If H^{-1} exists, Eq.(2.20) can be simplified as

$$\hat{f} = H^{-1}g \quad (2.21)$$

Eq.(2.21) is the restoration solution of unrestrained condition. Even though the form is simple, the calculation is very huge and H^{-1} should be simplified.

Restrained Restoration Equation

According to different areas, sometimes \hat{f} has different restrictions. Processed image need meet these restrictions. In solving Eq.(2.18), we need to use Lagrange's multiplier under these restrictions.

If the image has the same energy before and after restoration under the restorations, the mathematical expression is as follows:

$$\hat{f}^T \hat{f} = g^T g = C \quad (2.22)$$

Now we need solve the extremum of function $J(\hat{f})$ under Eq.(2.22). Lagrange's multiplier λ is introduced

$$J(\hat{f}, \lambda) = J(\hat{f}) + \lambda(\hat{f}^T \hat{f} - C) = (g - Hf)^T(g - Hf) + \lambda(\hat{f}^T \hat{f} - C) \quad (2.23)$$

Let

$$\frac{\partial J(\hat{f}, \lambda)}{\partial \hat{f}} = -2H^T(g - H\hat{f}) + 2\lambda\hat{f} = 0 \quad (2.24)$$

Then

$$\hat{f} = (H^T H + \lambda I)^{-1} \cdot H^T g \quad (2.25)$$

where I is denoted as unit matrix. Eq.(2.25) is the expression of estimated values if image maintains the same energy. In general, we can express the restrained restoration equation as: If Q is a linear operator of F , the problem of restrained restoration is to solve the extremum of the function $\| Q\hat{f} \|$ under condition

$$\| g - H\hat{f} \|^2 = \| n \|^2 \quad (2.26)$$

We use Lagrange's multiplier to solve the solution. Finally we obtain:

$$\hat{f} = \left(H^T H + \frac{1}{\lambda} Q^T Q \right)^{-1} \cdot H^T g \quad (2.27)$$

Eq.(2.27) is the general equation of restrained restoration. We need continuously adjust the value of parameter λ until good effect is obtained.

2.3 Restoration with Spatial Filtering, Frequency Domain Filtering and Wavelet Filtering

When only noise presents as the reason of the degradation image, $h \equiv 1$ and Eq.(2.10) becomes

$$g(x, y) = f(x, y) + n(x, y) \quad (2.28)$$

The noise terms are unknown, so subtracting them from $g(x, y)$ is not a realistic options. In fact, restoration and enhancement become almost indistinguishable disciplines in only additive noise presented.

2.3.1 Spatial Filtering

Spatial filtering [?] is the method of choice in a situation when only additive noise is presented.

Mean Filters

In this section we briefly discuss the mean filter and develop several other filters.

Arithmetic Mean Filter

The simplest of the mean filters is the arithmetic mean filter . Let R_{xy} denotes the pair of coordinates in a rectangular subimage window whose size is $m \times n$, centered at point (x, y) . Fig. 2.6 shows a square subimage window R_{xy} of size 3×3 . The arithmetic mean filtering procedure computes the average value of the contaminated image $g(x, y)$ in the region denoted by R_{xy} . The values of the recovered image \hat{f} at any point (x, y) is simply the arithmetic mean counted using the pixels in the area R_{xy} . It is presented as follows:

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in R_{xy}} g(s, t) \quad (2.29)$$

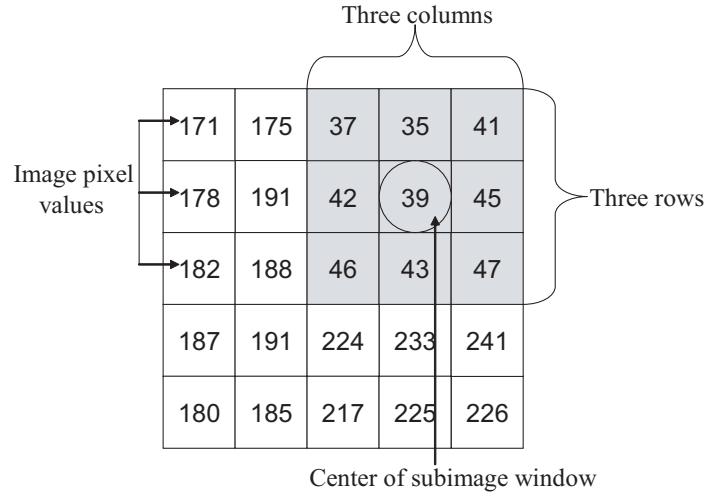


Fig. 2.6: A square subimage window R_{xy} of size 3×3 (redrawn from MATLAB manual).

This operation can be executed by employing a convolution mask in which the coefficients of all pixels is $1/mn$. A mean filter simply smoothes local difference in an image. Noise is reduced and caused of blurring.

Geometric Mean Filter

The function of geometric mean filtering in an image restoration is given by the expression

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in R_{xy}} g(s, t) \right]^{\frac{1}{mn}}. \quad (2.30)$$

Each pixel in image restored is generated by the product of the pixels in the subimage window. It raise the power to $1/mn$. A geometric mean filter gains smoothing compared to the arithmetic mean filter but lose fewer image details in the process.

Harmonic Mean Filter

The following expression presents to the harmonic mean filtering operation

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in R_{xy}} \frac{1}{g(s, t)}} \quad (2.31)$$

The harmonic mean filter eliminates salt noise well, but works poorly to pepper noise. It also work well on other types of noise like Gaussian noise.

Contraharmonic Mean Filter

The following expression shows the contraharmonic mean filtering operation which yields a restored image

$$\hat{f}(x, y) = \frac{\sum_{(s,t) \in R_{xy}} g(s, t)^{P+1}}{\sum_{(s,t) \in R_{xy}} g(s, t)^P} \quad (2.32)$$

where P is called the order of the filter. This filter fits to reduce or actually remove the noise of salt-and-pepper. For positive values of P , the filter removes pepper noise. For negative values of P , it reduces salt noise. It can not eliminate both noise at the same time. The contraharmonic filter becomes the arithmetic mean filter when $P=0$, and the harmonic mean filter when $P=-1$.

Order-Statistics Filters

Order-statistics filters are spatial filters whose response is founded on ordering or ranking the pixels in the image area enclosed by the filter. The response of the filter at any pixel is decided by the ordering result.

Median Filter

The median filter is the best famous order-statistics filter. The value of a pixel is replaced by the median of the gray levels in the subimage window of that pixel:

$$\hat{f}(x, y) = \underset{(s,t) \in R_{xy}}{\text{median}}\{g(s, t)\} \quad (2.33)$$

The computation of the median also includes the original value of that pixel. Because median filtering has excellent denoising ability for some types of random noise and considerably less blurring than linear smoothing filters of same size, it is popularly applied in wide areas. Median filters work especially well in the case of unipolar and bipolar impulse noise.

Max and Min Filters

The median filter is not the only one even though it is the most used order-statistics filter in image processing. The median denotes the 50% of a ranked set of numbers, however, ranking itself causes too many other probabilities in statistics. For example, the so called max filter results from the 100% ranking. It is given by

$$\hat{f}(x, y) = \underset{(s,t) \in R_{xy}}{\max}\{g(s, t)\}. \quad (2.34)$$

This filter is fairly useful for finding the brightest points in the image. Moreover, because pepper noise has very low value — nearly zero, it is reduced by this filter due to the maximum selection procedure in the subimage window R_{xy} .

The 0% filter is the min filter :

$$\hat{f}(x, y) = \underset{(s,t) \in R_{xy}}{\min}\{g(s, t)\}. \quad (2.35)$$

This filter is very useful to find the darkest point in the image, so it decreases salt noise as a result of the minimum operation.

Midpoint Filter

The midpoint filter simply counts the average value with the maximum and minimum values in the subimage window:

$$\hat{f}(x, y) = \frac{1}{2}[\underset{(s,t) \in R_{xy}}{\max}\{g(s, t)\} + \underset{(s,t) \in R_{xy}}{\min}\{g(s, t)\}]. \quad (2.36)$$

Note that this filter combines order statistics and averaging, so it make affection well for randomly distributed noise.

Alpha-trimmed Mean Filter

Alpha-trimmed mean filter [?, ?, ?] is widely used for the restoration of signals and images corrupted by additive symmetric noise. Assume that the $\alpha mn/2$ lowest and the $\alpha mn/2$ highest gray-level values of $g(s, t)$ are deleted in the subimage area R_{xy} . Also suppose that $g_{rm}(s, t)$ denotes the remaining $mn - \alpha mn$ pixels. The filter by averaging the remaining pixels is defined an alpha-trimmed filter:

$$\hat{f}(x, y) = \frac{1}{mn - \alpha mn} \sum_{(s, t \in R_{xy})} g_{rm}(s, t) \quad (2.37)$$

where α belong to 0—1. If $\alpha=0$, the alpha-trimmed filter is the same as the arithmetic mean filter. When α is equal to $\frac{mn-1}{mn}$, this filter is like a median filter. When α is other values between 0 to 1, the alpha-trimmed filter is helpful in cases including many types of noises like such as a combination of Gaussian and salt & pepper noise. Fig. 2.7 shows the result of filtering with a alpha-trimmed mean filter of size 3×3 and $\alpha=1/3$ when a combination of Gaussian and salt & pepper noise is added.

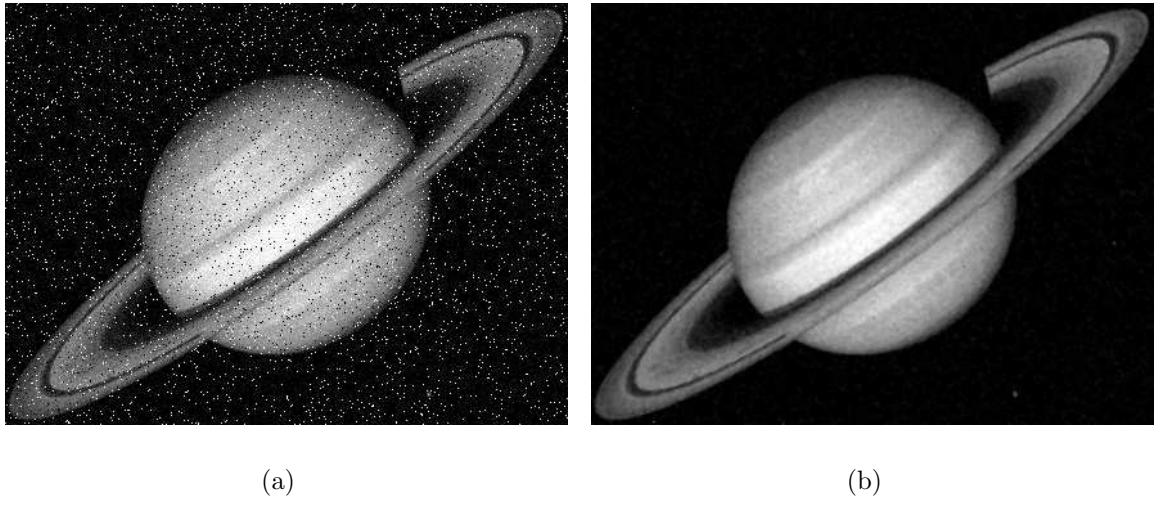


Fig. 2.7: An Alpha-trimmed filter: (a) Image corrupted by a combination of Gaussian and salt & pepper noise. (b) Image filtered with an alpha-trimmed mean filter size 3×3 and $\alpha=1/3$.

Alpha-trimmed filter is preferred when the probability density function (PDF) tail length is between that of Gaussian and Laplacian distributions. Meanwhile, it

remains more details of a restored image than a mean filter does. The design problem of these filters is nothing more than selecting its only parameter, α , optimally for a given noise PDF.

However, selecting the best α may be not possible if the noise PDF is not given or varies with time. In these cases, we should design an adaptive filter which changes the α value corresponding to the observed samples.

Signal Dependent Rank Order Mean Filter

Signal dependent rank order mean (SD-ROM) filter [?, ?, ?] is a nonlinear method to remove impulse noise from highly contaminated images however the details and edges are preserved at the same time. The algorithm depends on a detection-estimation strategy. All pixels are measured by a filtering detector. Only those pixels that are detected as contaminated are substituted to an estimation of a true value.

Even though the size and shape of a subimage window can be arbitrary, we used a 3×3 window center at $x(\mathbf{n})$ typically in most applications. For our purposes, $\mathbf{w}(\mathbf{n})$ is as an eight element observation vector neighboring with the pixels $x(\mathbf{n})$ except $x(\mathbf{n})$ in the window:

$$\begin{aligned}\mathbf{w}(\mathbf{n}) &= [w_1(\mathbf{n}), w_2(\mathbf{n}), \dots, w_8(\mathbf{n})] \\ &= [x(n_1 - 1, n_2 - 1), x(n_1 - 1, n_2), x(n_1 - 1, n_2 + 1), x(n_1, n_2 - 1), \\ &\quad x(n_1, n_2 + 1), x(n_1 + 1, n_2 - 1), x(n_1 + 1, n_2), x(n_1 + 1, n_2 + 1)],\end{aligned}\quad [2.38]$$

which corresponds from the 3×3 window starting from a left to right, top to bottom mapping to the one dimensional vector $\mathbf{w}(\mathbf{n})$.

First, we sort the observation samples by rank from smallest to largest

$$\mathbf{r}(\mathbf{n}) = [r_1(\mathbf{n}), r_2(\mathbf{n}), \dots, r_8(\mathbf{n})] \quad (2.39)$$

where $r_1(\mathbf{n}), r_2(\mathbf{n}), \dots, r_8(\mathbf{n})$ are the elements of $\mathbf{w}(\mathbf{n})$ arranged in increasing sequence, therefore $r_1(\mathbf{n}) \leq r_2(\mathbf{n}) \leq \dots \leq r_8(\mathbf{n})$.

Next, the rank-ordered mean (ROM) is defined as

$$m(\mathbf{n}) = \frac{(r_4(\mathbf{n}) + r_5(\mathbf{n}))}{2} \quad (2.40)$$

Finally, the rank-ordered differences are denoted as:

$$\mathbf{d}(\mathbf{n}) = [d_1(\mathbf{n}), d_2(\mathbf{n}), d_3(\mathbf{n}), d_4(\mathbf{n})] \quad (2.41)$$

where

$$d_i(\mathbf{n}) = \begin{cases} r_i(\mathbf{n}) - x(\mathbf{n}) & x(\mathbf{n}) \leq m(\mathbf{n}) \\ x(\mathbf{n}) - r_i(\mathbf{n}) & x(\mathbf{n}) > m(\mathbf{n}) \end{cases} \quad (2.42)$$

for $i=1, \dots, 4$.

The process of the SD-ROM filter detects and estimates as follows:

Noise detection:

If any of the below inequations are set, then the algorithm detects the current pixel $x(\mathbf{n})$ as a corrupted noise pixel :

$$d_i(\mathbf{n}) > T_i, \quad i = 1, \dots, 4 \quad (2.43)$$

where T_1, T_2, T_3 and T_4 are threshold values, within ascending order.

Estimation of the true value:

If $x(\mathbf{n})$ is detected as a contaminated pixel, it will be substituted by $m(\mathbf{n})$, otherwise it is retained with no change. We shows the result of filtering with a SD-ROM filter of size 3×3 and $T_1=8, T_2=20, T_3=40, T_4=50$ in Fig. 2.8 when a salt & pepper noise is added. Its MSE is only 1.4825 while the MSE of corrupted image is 1250.9. Therefore it is clear this filter can significantly reduce the impulse noise.

2.3.2 Frequency Domain Filtering

Noise and edges in the gray levels of an image provide significantly the high-frequency component in its Fourier transform . Thus denoising is obtained by diminishing a special high-frequency area in the transformation of a given image in the frequency domain.

The model for filtering in the frequency domain is given as

$$G(u, v) = H(u, v)F(u, v) \quad (2.44)$$

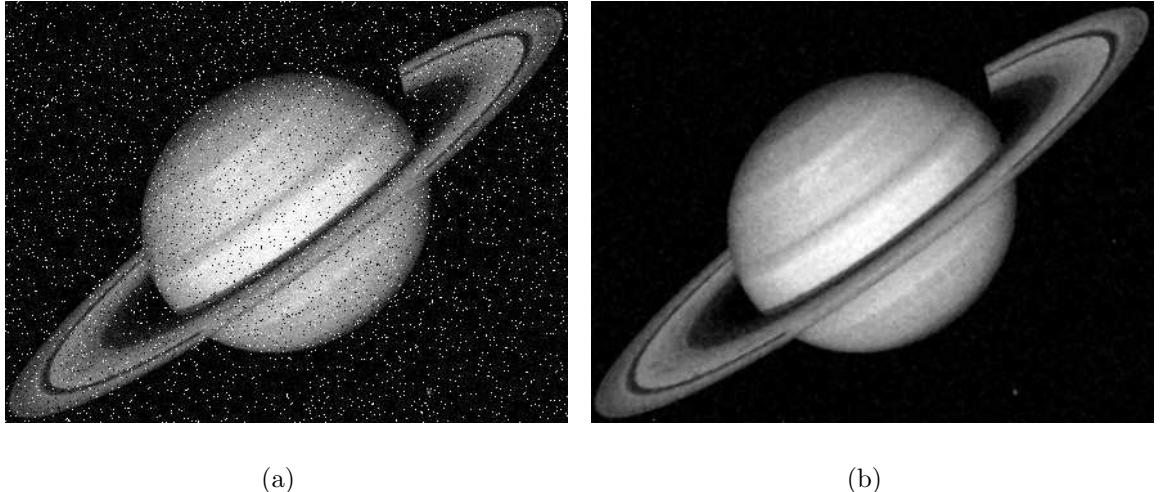


Fig. 2.8: A SD-ROM filter: (a) An image corrupted by salt & pepper noise. (b) The result of filtering with a SD-ROM filter of size 3×3 .

where $F(u, v)$ is the Fourier transform of the image to be denoised. The goal is to choose a filter transformation function $H(u, v)$ that produces $G(u, v)$ by decreasing high-frequency components including the noise in $F(u, v)$.

Lowpass filter can be classified into three types in frequency-domain: ideal filter , Butterworth filter , Gaussian filter . These three filters cover the range from very sharp (ideal) to very smooth (Gaussian) filter function. We also designed a Lowpass filter with frequency transformation method , frequency sampling method and windowing method .

Classification of Lowpass Filters

Lowpass filter can be classified as

Ideal Lowpass Filters

The simplest lowpass filter is a filter that eliminates all high frequency components in the Fourier transform when a distance is bigger than a special distance D_0 from the origin of the centered transform. This filter is named as a two dimensional ideal lowpass filter (ILPF) and has the transfer function as

$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases} \quad (2.45)$$

where D_0 is a special positive quantity and $D(u, v)$ is the distance from point (u, v) to the center of the frequency rectangle. If the image is in size $M \times N$, its transform also is in this size, then the center of the frequency rectangle is at $(u, v) = (M/2, N/2)$ because the transform has been centered. In the case, the distance from a point (u, v) to the origin of Fourier transform is given by

$$D(u, v) = [(u - \frac{M}{2})^2 + (v - \frac{N}{2})^2] \quad (2.46)$$

An ideal lowpass filter transfer function is shown in Fig 2.9(a)

Butterworth Lowpass Filters

The transfer function of a Butterworth lowpass filter (BLPF) of n th with cutoff frequency at a distance D_0 from the origin order is defined as

$$H(u, v) = \frac{1}{1 + (D(u, v)/D_0)^{2n}} \quad (2.47)$$

where $D(u, v)$ is given by Eq.(2.45). An Butterworth lowpass filter transfer function is shown in Fig 2.9(b).

Unlike the ILPF, the BLPF transfer function does not have a sharp discontinuity which builds a clear cutoff between passed and stopped frequencies. We can define a cutoff frequency place at points for which $H(u, v)$ is down to a special fraction of its maximum customary quantity.

Gaussian Lowpass Filters

The form of a two dimensional Gaussian lowpass filter is given by

$$H(u, v) = e^{-D^2(u, v)/2\sigma^2} \quad (2.48)$$

where $D(u, v)$ is the distance from the center of the Fourier transform as in Eq.(2.46) which we assume has been transferred to the center of the frequency rectangle, σ is a measurement of the spread of the Gaussian curve. If we set $\sigma = D_0$, the filter can be expressed as

$$H(u, v) = e^{-D^2(u, v)/2D_0^2} \quad (2.49)$$

where D_0 is the cutoff frequency. A Gaussian lowpass filter transfer function is shown in Fig 2.9(c).

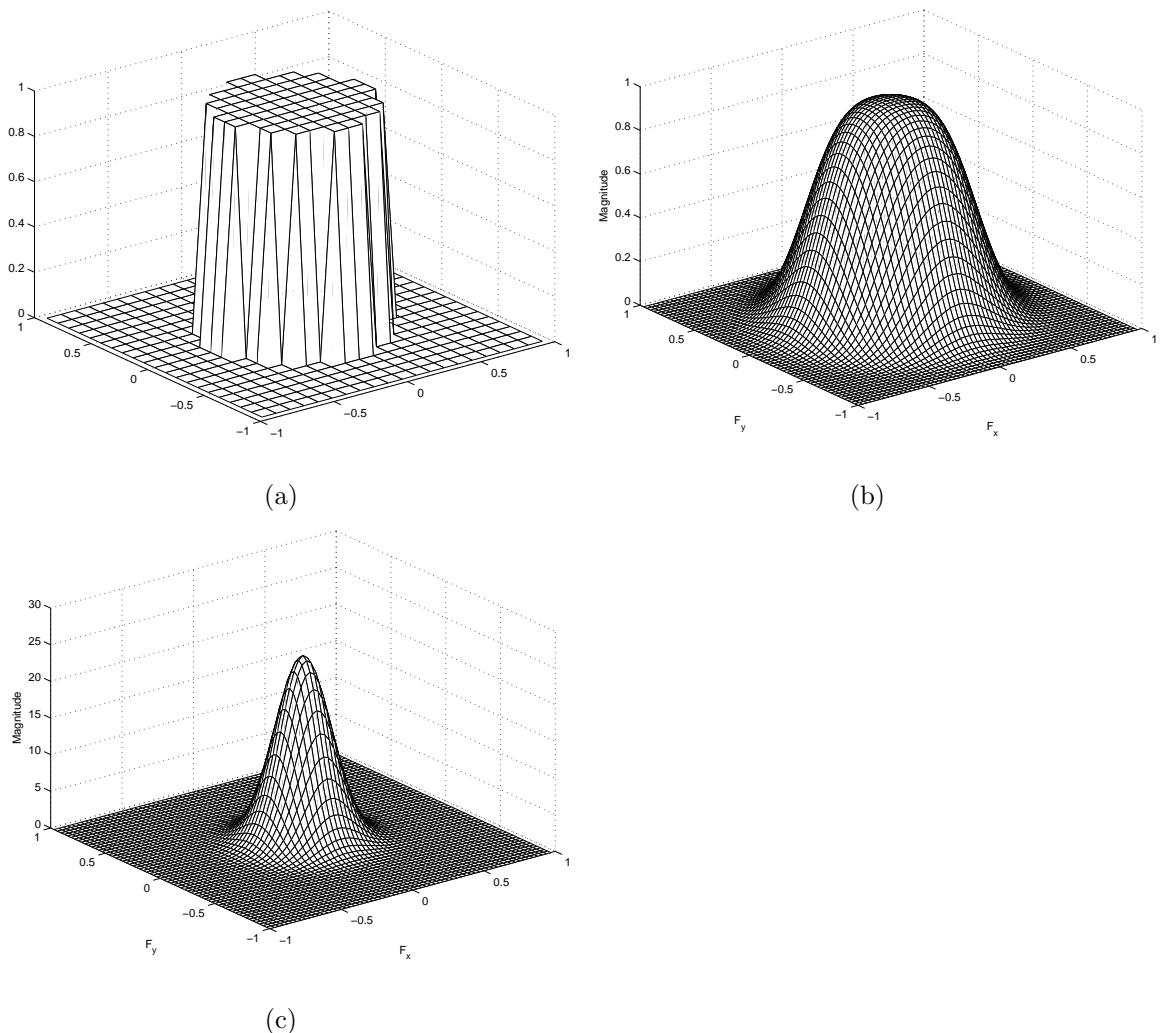


Fig. 2.9: *Lowpass filter transfer functions:* (a) An ideal lowpass filter transfer function. (b) A Butterworth lowpass filter transfer function. (c) A Gaussian lowpass filter transfer function.

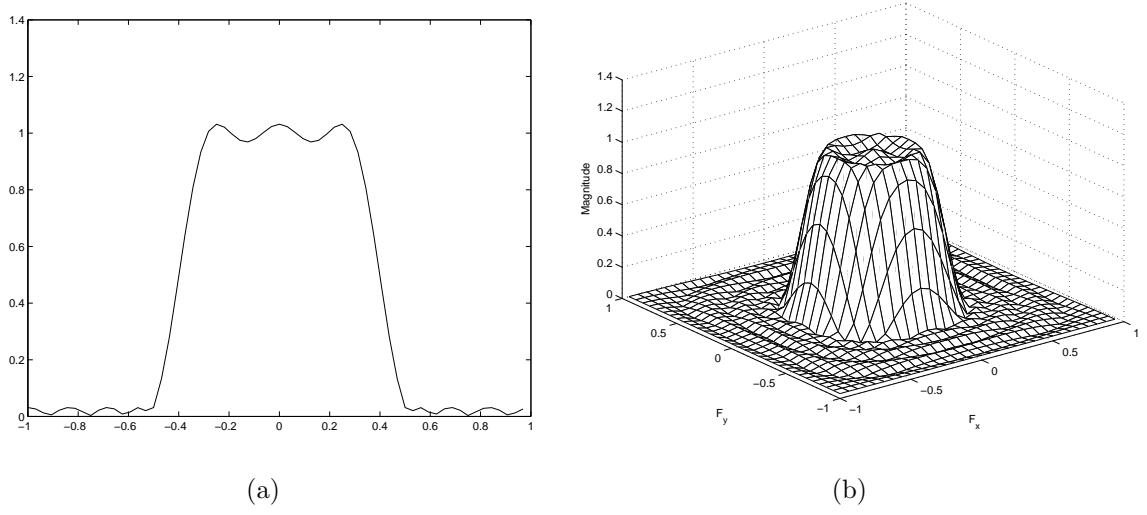


Fig. 2.10: *The 14th order lowpass filter frequency response with truncation frequency of 0.5 designed by frequency transformation method: (a) A one-dimensional frequency response . (b) A two-dimensional frequency response.*

Design Methods of Frequency Domain Filters

Filter design is the key no matter which filter mode is used. For simplifying calculation and reducing consumed time, the design of filter is set in frequency domain. There are three main design methods:

Frequency Transformation Method

Frequency transfer method first designs a one-dimensional finite impulse response (FIR) filter of good performance and then transfers to a two-dimensional FIR filter. Because designing a one-dimensional special function filter is much easier than designing a two-dimensional filter, the frequency transfer method can be useful. It preserves most of the properties of the one-dimensional filter, particularly the transition bandwidth and ripple characteristics. It has effective realizing structure and smaller calculation than convolution or discrete fourier transform (DFT). A 14th order lowpass filter frequency response with truncation frequency of 0.5 designed by frequency transformation method is shown in Fig. 2.10.

Frequency Sampling Method

Frequency sampling method produces a filter based on a expected frequency re-

sponse. If a matrix of points that defines the shape of the frequency response is given, this method produces a filter whose frequency response passes through those points. Assume $h(m, n)$ is the expected frequency response matrix, where $0 \leq m \leq M$ and $0 \leq n \leq N$. Its frequency response is

$$H(e^{jw_1}, e^{jw_2}) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h(m, n) e^{-j(w_1 m + w_2 n)}. \quad (2.50)$$

The sampling in frequency is

$$H(k, l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h(m, n) e^{-j2\pi(\frac{mk}{M} + \frac{nl}{N})} \quad (2.51)$$

here k and l are sampling pixels and $0 \leq k \leq M$ and $0 \leq l \leq N$. The designed impulse response is

$$h(m, n) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} H(k, l) e^{-j2\pi(\frac{mk}{M} + \frac{nl}{N})}. \quad (2.52)$$

For a two dimensional sampling filter of linear phase, the calculation process can be further simplified by the symmetric relation of DFT coefficients. A lowpass filter frequency response designed by frequency sampling method is shown in Fig. 2.11.

Windowing Method

The windowing method multiplies the ideal impulse response by a window function to create a corresponding filter. Even though the windowing method produces a filter whose frequency response approximates a desired frequency response similar to the frequency sampling method, the windowing method tends to produce better results than the frequency sampling method.

Assume the unit impulse response of the designed two dimensional FIR filter is

$$h(n_1, n_2) = \omega(n_1, n_2) \cdot i(n_1, n_2) \quad (2.53)$$

where $i(n_1, n_2)$ is an infinite impulse response sequence, $\omega(n_1, n_2)$ is a two dimensional windowing function. Applying Fourier transform in Eq.(2.53), it becomes

$$H(\omega_1, \omega_2) = \frac{1}{4\pi^2} \int \int I(\Omega_1, \Omega_2) W(\omega_1 - \Omega_1, \omega_2 - \Omega_2) d\Omega_1 d\Omega_2 \quad (2.54)$$

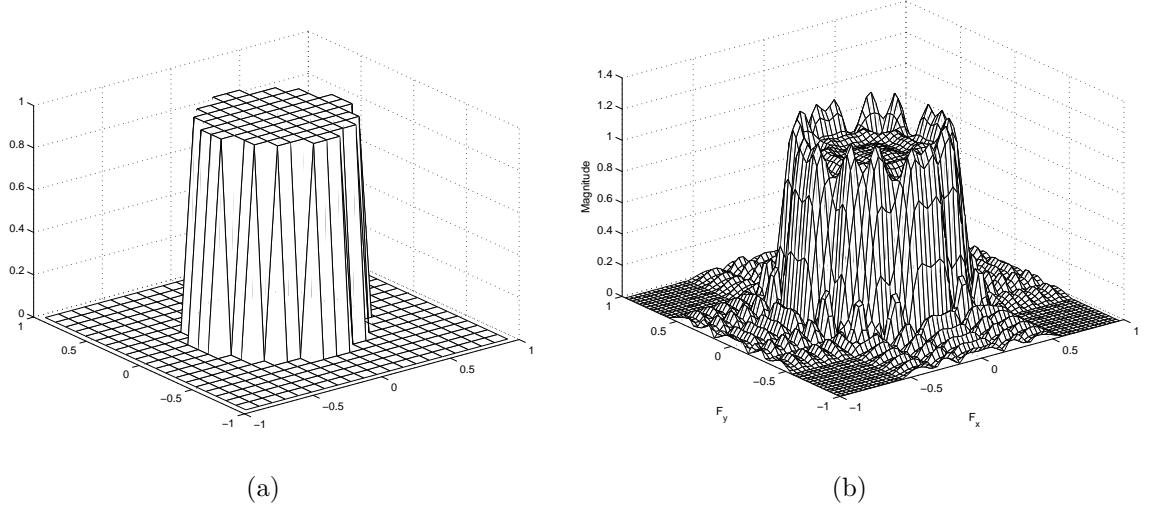


Fig. 2.11: *Lowpass filter frequency response designed by frequency sampling method:*
 (a) *Desired two-dimensional frequency response.* (b) *Actual two-dimensional frequency response.*

Eq.(2.54) indicates that the effect adding window to infinite impulse response is equal to smoothing $I(\omega_1, \omega_2)$ with $W(\omega_1, \omega_2)$ in frequency domain.

The common windowing functions are boxcar window, Bartlett window, Hamming window, Hanning window, Blackman window and Kaiser window. Fig. 2.12 illustrates Bartlett window, Hamming window, Hanning window, Blackman window separately.

2.3.3 Wiener Filter

Wiener filter considers signals and noise as stochastic signals. It designs the optimal filter based on the analysis and statistic for these stochastic signals. Stochastics implies that there is a lack or no information known about the distortion (blurring and noise). We can not write the mathematical expression for noise or input signal because of lack or no information. Therefore, we can not analyze the quality and quantity of the signals.

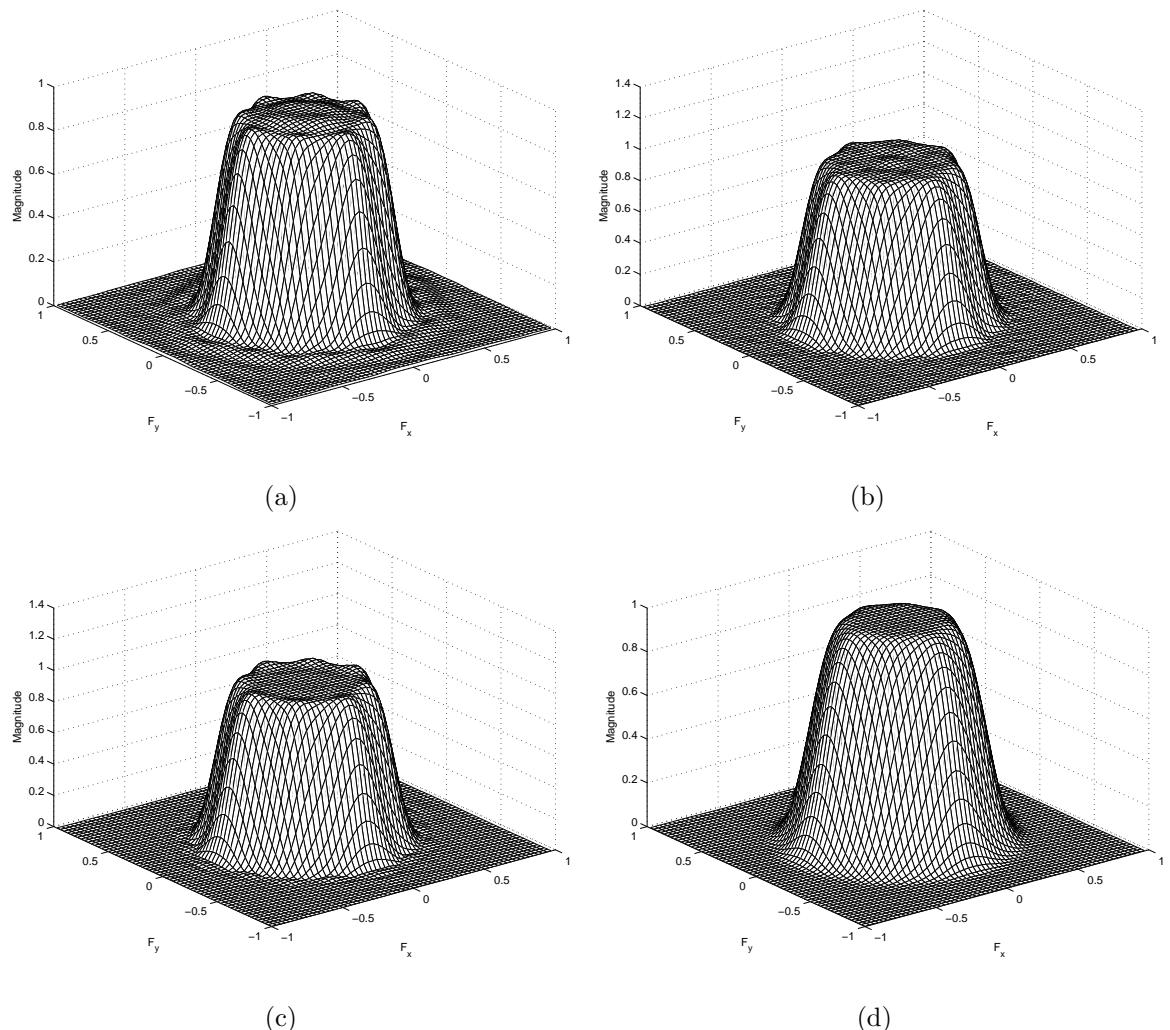


Fig. 2.12: Lowpass filter frequency responses designed by windowing method: (a) Bartlett window; (b) Hamming window; (c) Hanning window; (d) Blackman window.

Stochastic Variables

We can handle Stochastic Variables as follows: Consider a sampling set including infinite function members and assume a membership function (but not sure which one) is blemished noise. By making the general representation applicable for the entire sampling set, using this general representation denotes the corrupted signals. For handling these stochastic variables, the expected operator is introduced:

$$\epsilon(x(t)) = \int_{-\infty}^{\infty} x(t)dt. \quad (2.55)$$

This operator expresses the mean of the stochastic variable x at time t . Assume the autocorrelation function of signal $x(t)$ is given by

$$R_n(\tau) = \int x(t)x(t+\tau)dt. \quad (2.56)$$

Then the power spectrum of $x(t)$

$$P_n(s) = \mathcal{F}\{R_n(\tau)\} \quad (2.57)$$

is known. It is said that the amplitude spectrum of signal $x(t)$ is given but the phase spectrum is unknown. In fact, membership function set is made up of the infinite function mean with the difference in phase.

Principle of Wiener Filter

Wiener filter is a classical linear denoising filter and Wiener filtering is an adaptive filtering. It can adjust the output according to the local square error of the image. If the local square error is larger, the smoothing effect of this filter is stronger. Its final goal is to minimize as much as possible the mean square error (MSE) between the restored image $\hat{f}(x, y)$ and the original image $f(x, y)$. This error measurement is given by

$$e^2 = E[(f(x, y) - \hat{f}(x, y))^2] \quad (2.58)$$

where $E[.]$ is the expectation of the argument. It is assumed that the noise is not related to the image in Eq.(2.58). It is also assumed that the noise has a zero mean and

that the gray levels in the estimated image have a linear relationship with the levels in the degradation image. The minimum in Eq.(2.58) is expressed in the frequency domain as follows:

$$\begin{aligned}\hat{F}(u, v) &= \left[\frac{H^*(u, v)S_f(u, v)}{S_f(u, v)|H(u, v)|^2 + S_\eta(u, v)} \right] G(u, v) \\ &= \left[\frac{H^*(u, v)}{|H(u, v)|^2 + S_\eta(u, v)/S_f(u, v)} \right] G(u, v) \\ &= \left[\frac{1}{H(u, v)} \frac{H^*(u, v)^2}{|H(u, v)|^2 + S_\eta(u, v)/S_f(u, v)} \right] G(u, v)\end{aligned}\quad (2.59)$$

where

$H(u, v)$ is the transform of the degradation function,

$H^*(u, v)$ is complex conjugate of $H(u, v)$,

$|H(u, v)|^2 = H^*(u, v)H(u, v)$,

$S_\eta(u, v) = N(u, v)$ which is the power spectrum of the noise,

$S_f(u, v) = F(u, v)$ which is the power spectrum of the undegraded image,

$G(u, v)$ is the transform of the degraded image,

$\hat{F}(u, v)$ is the frequency domain estimate of the degraded image.

The restored image in the spatial domain is obtained by the inverse Fourier transform of the frequency domain estimate $\hat{F}(u, v)$. The fact is used that the product of a complex quantity with its conjugate is equal to the magnitude of the complex quantity squares. This result was obtained by Wiener filter, whose concept first was presented by N. Wiener [?] in 1942. The filter which is inside the brackets is commonly called the minimum mean square error filter or the least mean square error filter.

2.3.4 Denoising with Wavelet Analysis

Fourier transform has been the widest used method of analysis in the signal processing since it was presented by Fourier in 1822. It decomposes a signal into a summation of a series of continuous sinusoids with different frequencies or transfers signals from a temporal domain to a frequency domain as shown in Fig. 2.13.

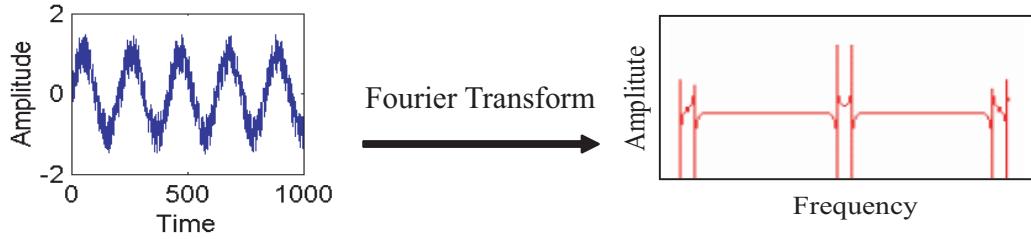


Fig. 2.13: *Perspective plot of Fourier transform (redwave from MATLAB manual).*

However, Fourier analysis has a serious weakness. In transforming to the frequency domain, time information is missed. Most signals contain numerous drift, trends, abrupt changes. These situations are often the most important message of the signal, and Fourier analysis is not suited to detecting them.

In 1946, Gabor presented the famous Gabor transform and then developed further as short time Fourier transform (STFT) shown in Fig 2.14. It added a small window to analyze local characteristics of signal. However, the size and shape of the window function is not changeable according to time and frequency. It is shortcoming to analyze a time-changeable signal because high frequency maintains a short time and needs a small window, and low frequency maintains a long time and needs a large window.

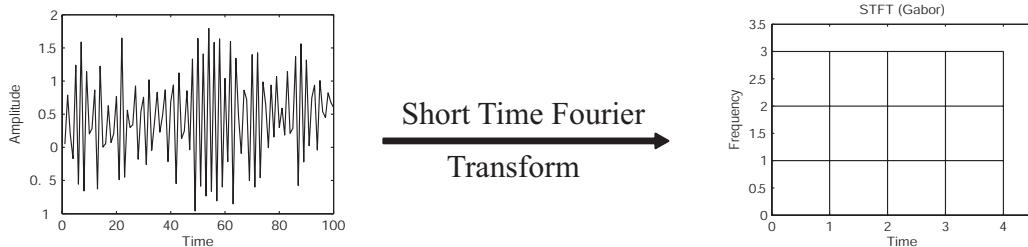


Fig. 2.14: *Perspective plot of STFT.*

Wavelet analysis offsets the shortage of STFT: a windowing technique with variable-sized regions as shown in Fig. 2.15. Wavelet analysis allows us to use long time intervals when we want low-frequency information, and shorter regions when we want high-frequency information. The contrast of the time-based, frequency-based, STFT and wavelet views of a signal is shown in Fig. 2.16

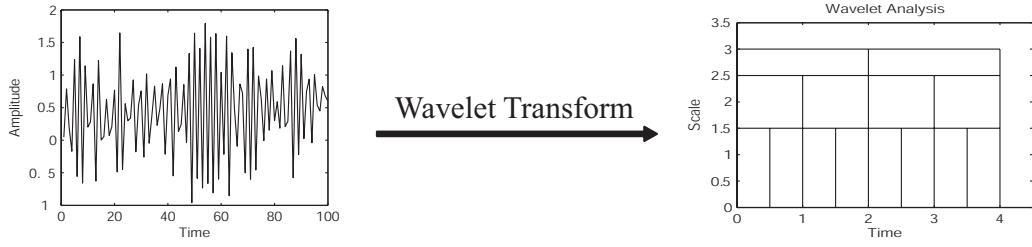


Fig. 2.15: *Perspective plot of Wavelet.*

Definition and Characteristics of Wavelet Transform

A wavelet is a waveform of specially limited duration and has a mean value of zero. Unlike sinusoid which is the basis of Fourier analysis, wavelets tend to be irregular, asymmetric and of limited duration. Four types of wavelet function are shown in Fig. 2.17. Wavelet analysis is the decomposition of a signal into shifted and scaled versions of the mother wavelet to obtain a series of coefficients like sinusoid in Fourier transform.

Continuous Wavelet Transform

The continuous wavelet transform (CWT) is defined as the summation over all time of the signal multiplied by scaled, shifted versions of the wavelet function Ψ . The expression is given by:

$$C(s, p) = \int_{-\infty}^{\infty} f(t)\Psi(s, p, t)dt \quad (2.60)$$

where s is the scale and p is the shifted position of the wavelet function, and t is time. The results of the CWT are many wavelet coefficients C , which are a function of scale and time position. The different scales of wavelet function are illustrated in Fig 2.18.

The five steps for creating a CWT are easy to follow:

Step 1: Choose a wavelet and compare it to a section at the start of the original signal.

Step 2: Calculate a scale (coefficient), C , which expresses how closely correlated the wavelet is with this section of the signal.

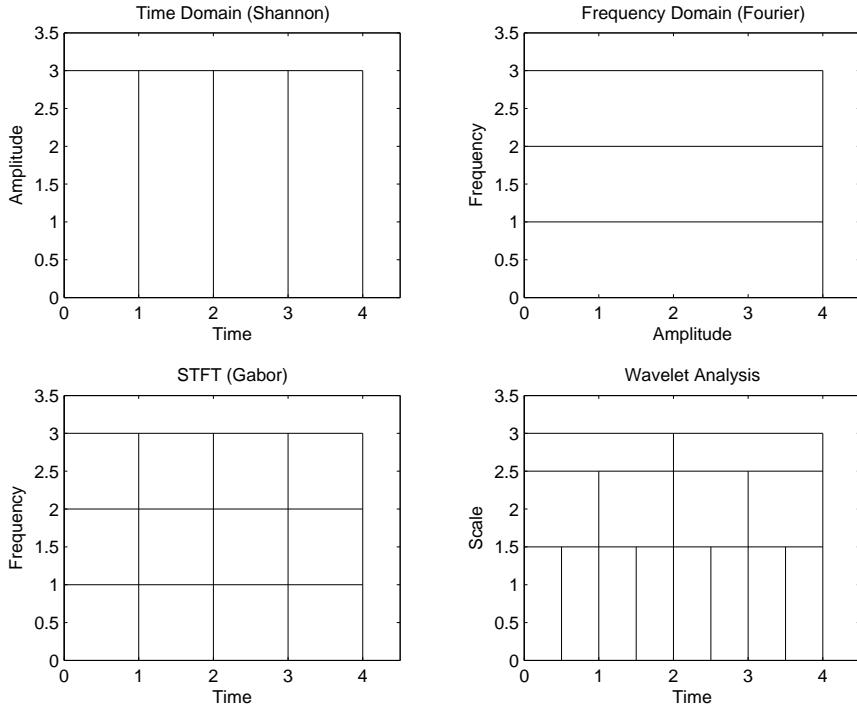


Fig. 2.16: *The contrast of different signal analysis methods.*

Step 3: Shift the wavelet to the next time position and repeat steps 1 and 2 until you've obtained the coefficients of the whole signal.

Step 4: Scale or stretch the wavelet and repeat steps 1 through 3.

Step 5: Repeat steps 1 through 4 for all scales.

The perspective plot of calculation process for wavelet coefficients is shown in Fig 2.19.

Discrete Wavelet Transform (DWT)

The low frequency component is the most important part in many signals. The content of the signal can be identified by examining the low frequency component of the signal. However, the high frequency component expresses the flavor or nuance. In wavelet analysis, the low frequency component of the signals is called the approximations . The details are the low-scale, high-frequency components. The coefficients of the approximations are obtained when the signals pass by the lowpass filter in wavelet transform. Meantime, the coefficients of the details are obtained when the signals pass by the highpass filter in wavelet transform. The perspective plot of a 2

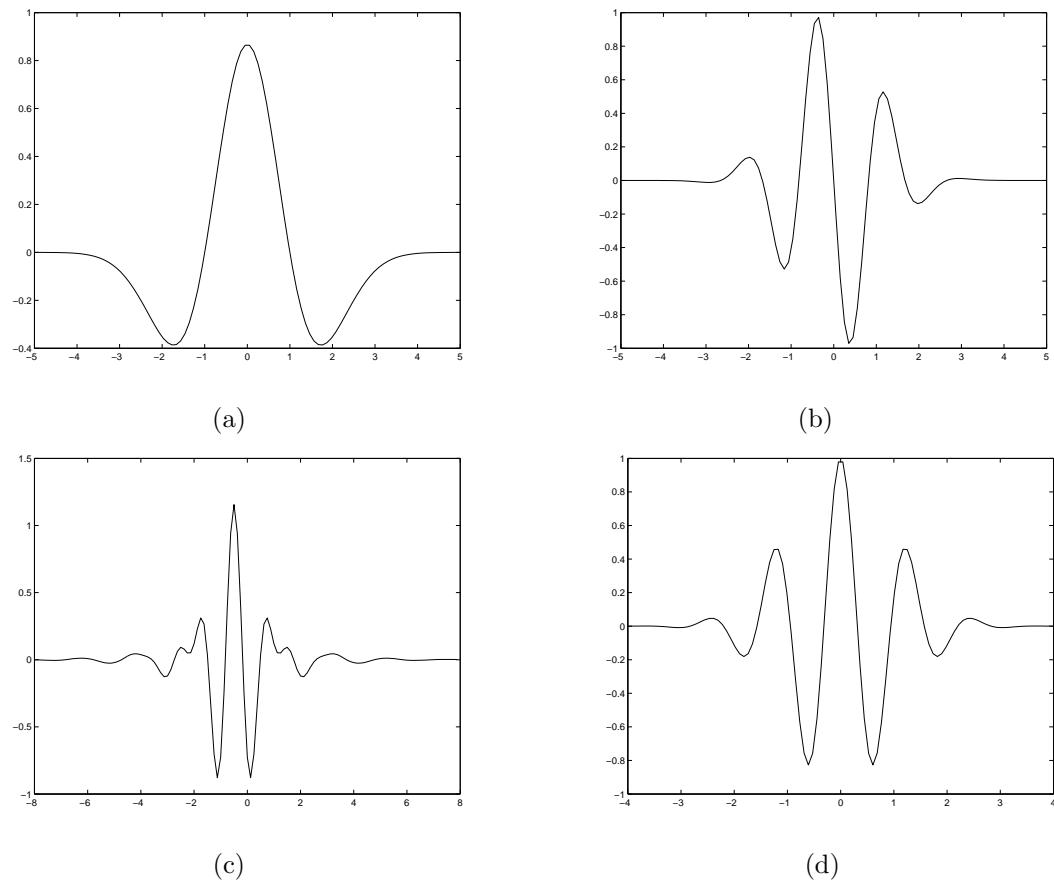


Fig. 2.17: Four types of wavelet functions: (a) Mexican hat wavelet function. (b) Gaussian wavelet function. (c) Meyer wavelet function. (d) Morlet wavelet function.

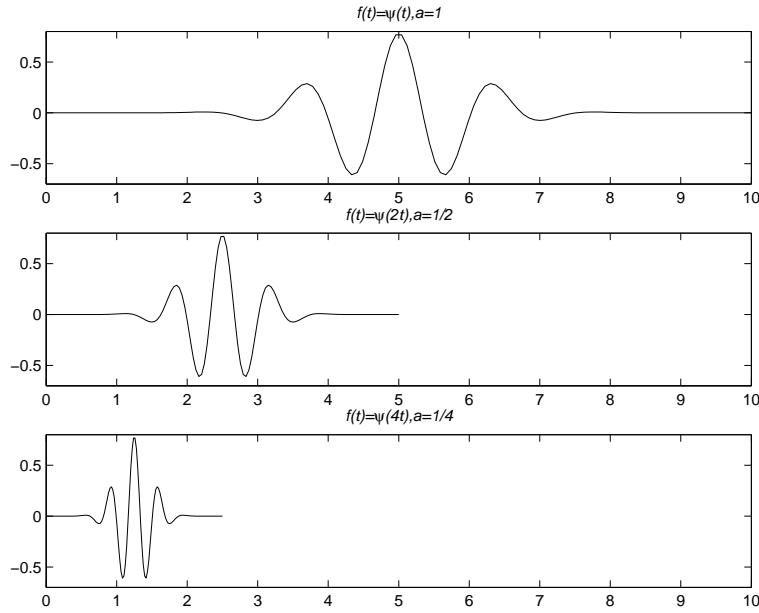


Fig. 2.18: The different scales of wavelet function.

lever image decomposition is shown in Fig. 2.20.

The decomposition process can be repeated and successive approximations can be decomposed in turn so that one signal is decomposed into lower resolution components. This is called the wavelet decomposition tree as shown in Fig. 2.21(a).

Fig. 2.21(b) illustrates a practical case of wavelet decomposition. Of course, the decomposition levels can not be chosen arbitrarily. For example, a signal of length N can be decomposed into $\log_2 N$ levels. The suitable levels can be chosen any integer in the area $(0, \log_2 N)$.

We've known the process analyzing or decomposing signals and images in the discrete wavelet transform. This process is called decomposition or analysis. The inverse process of decomposition or analysis allows those components to be assemble back into the signal we want to obtain without loss of information. This process is called reconstruction, or synthesis. The conduct that effects synthesis is called inverse discrete wavelet transform (IDWT). The left side of Fig. 2.22 shows two level decomposition to the coefficients of the approximations and details, and the right side shows the process of reconstruction in a image wavelet analysis. Fig. 2.23 illustrates an example of decomposition from one to three level with wavelet function *haar*.

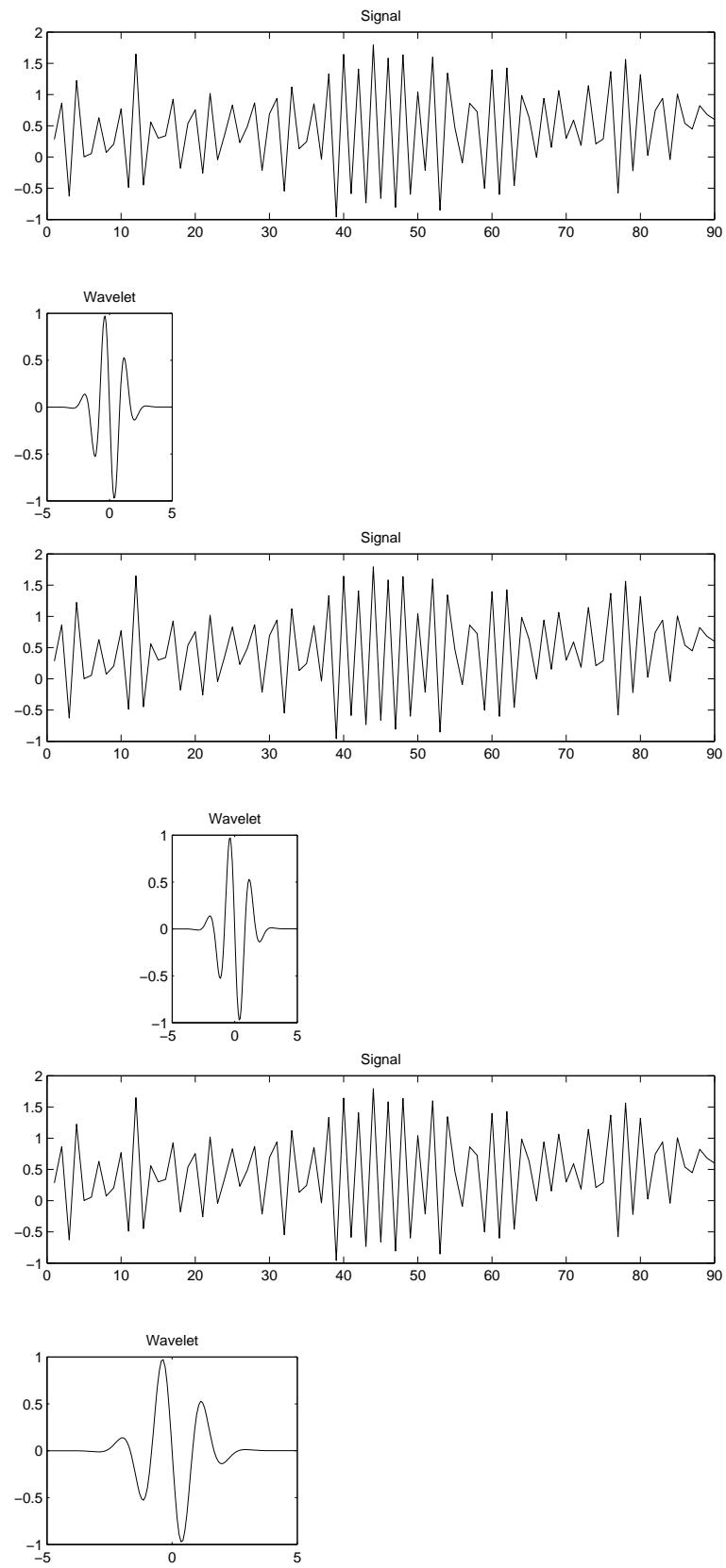


Fig. 2.19: Calculating the coefficient by scaled, shifted versions of the wavelet function.

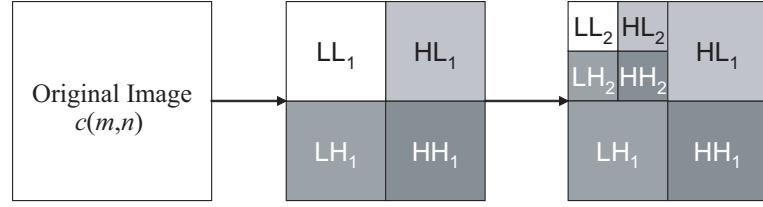


Fig. 2.20: The perspective plot of a 2 level image decomposition (redrawn from Gonzalez and Woods (2001)).

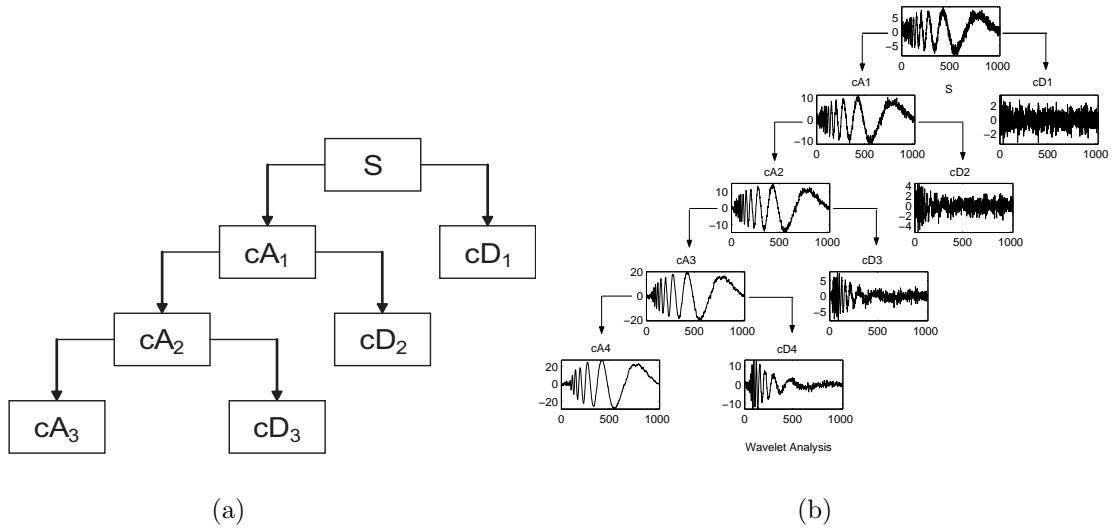


Fig. 2.21: Multiple level decomposition trees: (a) Three level decomposition tree. (b) Four level decomposition tree in a practical case (redrawn (a) from MATLAB manual).

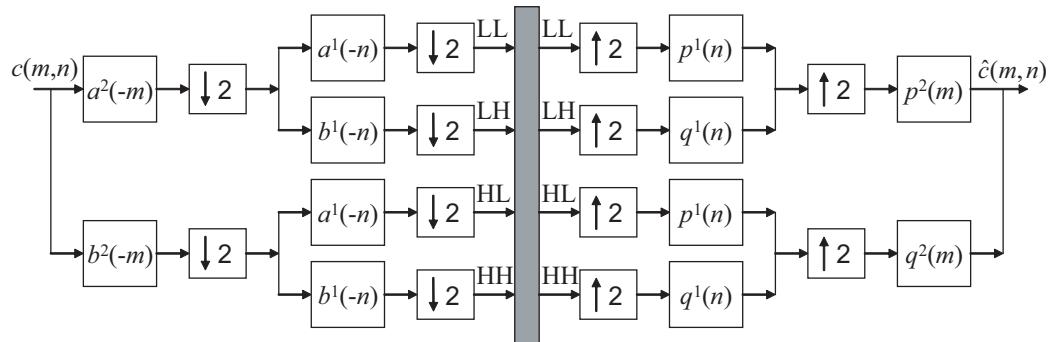


Fig. 2.22: Two level decomposition and reconstruction of an image (redrawn from MATLAB manual).

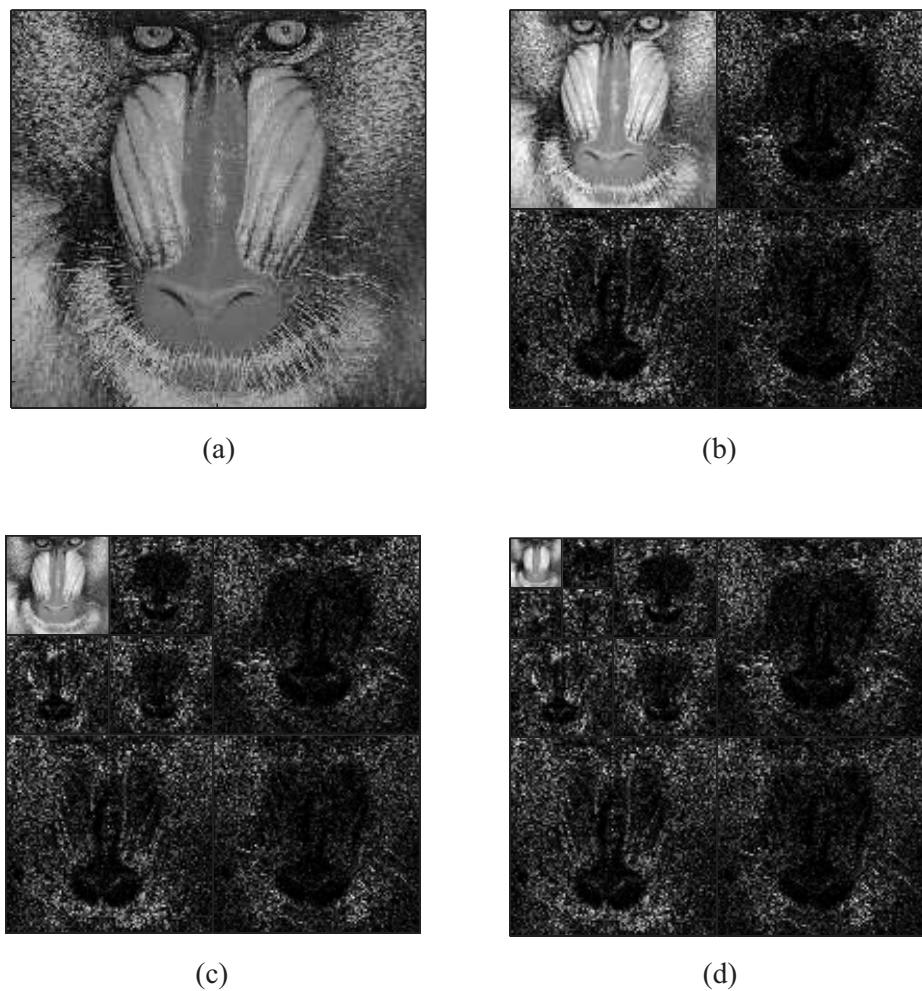


Fig. 2.23: *The original image and its decomposition at different levels: (a) Original image; (b) Decomposition at level 1; (c) Decomposition at level 2; (d) Decomposition at level 3.*

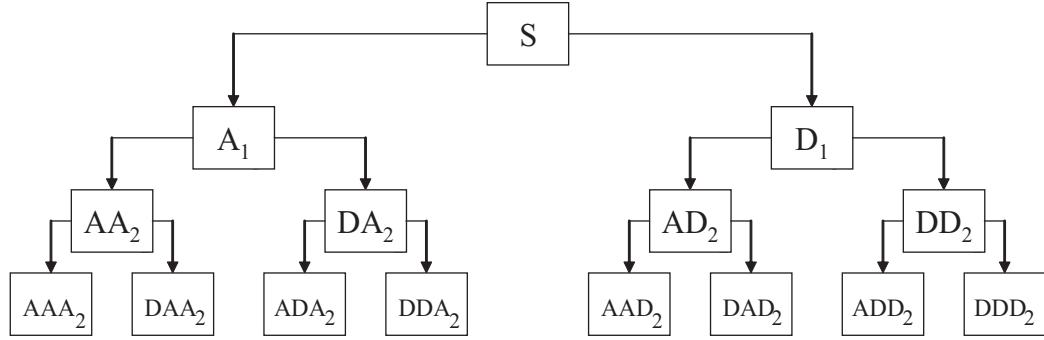


Fig. 2.24: A two level decomposition tree with wavelet packet (redrawn from MATLAB manual).

Wavelet Packet Analysis

In wavelet analysis, a signal is divided into an approximation and a detail. The approximation itself is then divided into a second-level approximation and detail, and the process is repeated. For an n level decomposition, there are $n+1$ possible ways to decompose.

However, unlike wavelet analysis, the details as well as the approximations can be split in wavelet packet analysis . This yields more than $2^{2^{n-1}}$ different ways to decompose. Fig 2.24 is the schematic diagram of wavelet packet decomposition tree. Therefore, wavelet packet method is a generalization of wavelet decomposition that offers a richer range of possibilities for signal analysis than does wavelet method.

Denoising with Wavelet

The model for the noisy signal in an image is as Eq.(2.28). The denoising goal is to suppress the noise part $n(x, y)$ of the image $g(x, y)$ and to recover $f(x, y)$.

Denoising Procedure Principles

Because the noise part $n(x, y)$ can be thought of as a high frequency signal and the image is a low frequency signal, we can follow the below method to handle the noise: First decompose the signal with wavelet transform. In general, the noise signal mostly includes the details of high frequency so that we can use a threshold to cope with the decomposed wavelet coefficients. Then reconstruct with the wavelet coefficients and

finally reach the objective for denoising the signal.

The denoising procedure principles can be divided into three steps:

Step 1: Decompose Choose a wavelet function and a level N . Compute the wavelet decomposition coefficients of the signal g at level N .

Step 2: Threshold detail coefficients For each level from 1 to N , select a threshold and apply hard and/or soft threshold to the detail coefficients.

Step 3: Reconstruct Reconstruct the image using the original approximation coefficients of level N and the modified detail coefficients of levels from 1 to N .

Fig. 2.25 illustrates the wavelet approximations and details in a level three with function $db1$ for visually representation of the process of denoising an image with noise.

Hard and Soft Threshold

Assume t denotes the threshold. The hard threshold signal is

$$T_h = \begin{cases} x & |x| > t \\ 0 & |x| \leq t \end{cases} \quad (2.61)$$

The soft threshold signal is

$$T_s = \begin{cases} \text{sign}(x)(|x| - t) & |x| > t \\ 0 & |x| \leq t \end{cases} \quad (2.62)$$

We show the curves of (a) the original signal, (b) hard threshold and (c) soft threshold in Fig. 2.26.

2.4 Denoising Images with Adaptive Neuro-Fuzzy Inference Systems

Fuzzy reasoning [?, ?, ?] is a procedure which maps a given input space to a special output space by the methods of fuzzy logic. The mapping procedures adhere to the

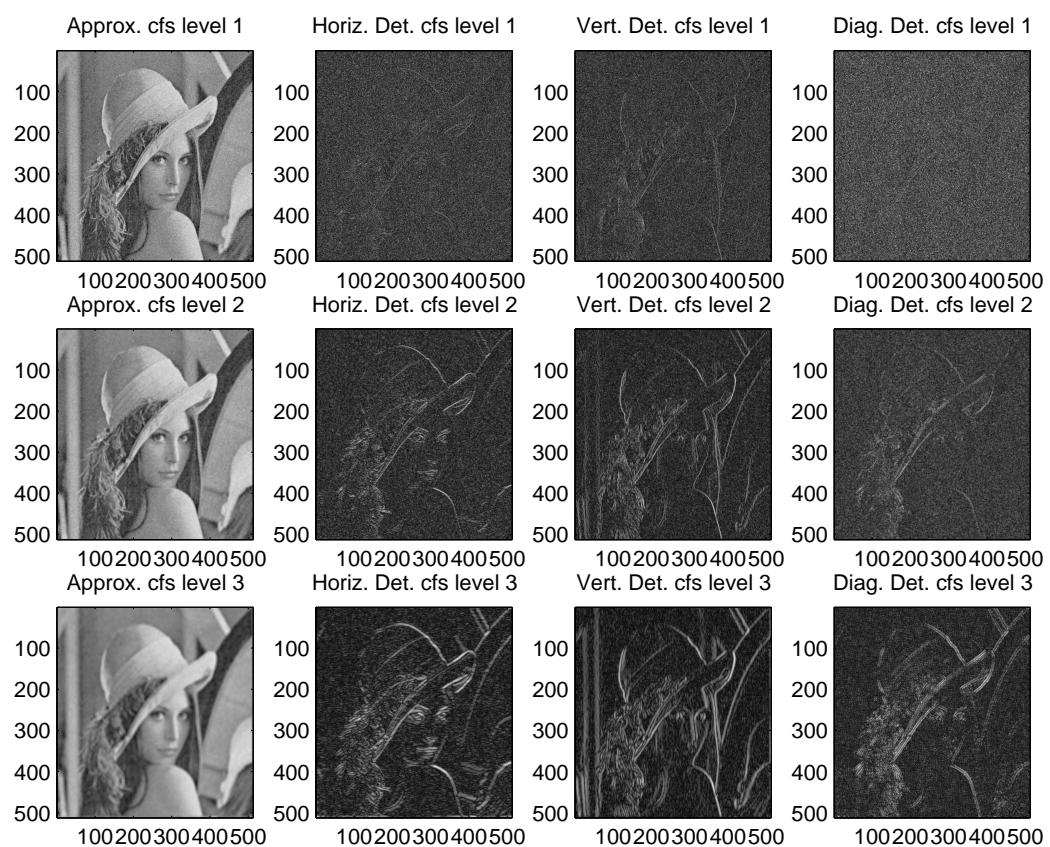


Fig. 2.25: A three level decomposition and reconstruction for an image.

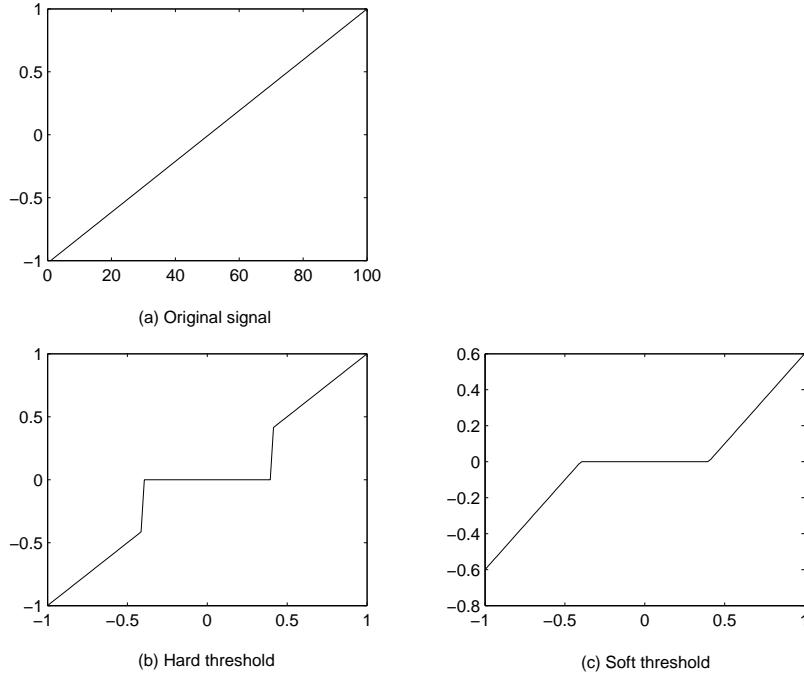


Fig. 2.26: *The curves of (a) original signal; (b) hard threshold; (c) soft threshold.*

basic theories of membership functions (MF), the logical operation, if-then rule et cetera. They have been successfully applied to the areas of automatic control, data classification, decision analysis, pattern recognition, expert systems and computer visual systems.

2.4.1 Introduction

Fuzzy reasoning is a wide concept. It includes fuzzy rule systems, fuzzy expert systems, fuzzy modelling, fuzzy controllers, and simple fuzzy systems differentiated by the different characteristics. It is the backbone of fuzzy inference systems (FIS).

The common FIS can be classified into three models: pure FIS, Takagi-Sugeno FIS [?] and Mamdani FIS [?].

Pure Fuzzy Inference Systems

The inputs and outputs of pure FIS are all fuzzy sets shown in Fig. 2.27. The fuzzy rule database consists of several if-then rules. Fuzzy reasoning is the core of the fuzzy

inference system. It maps input fuzzy sets to output fuzzy sets according to fuzzy rules. Because the inputs and outputs of this inference system are all fuzzy sets, it provides a scaled message of expert language and generates the general model for using this kind of linguistic message under the fuzzy sets.

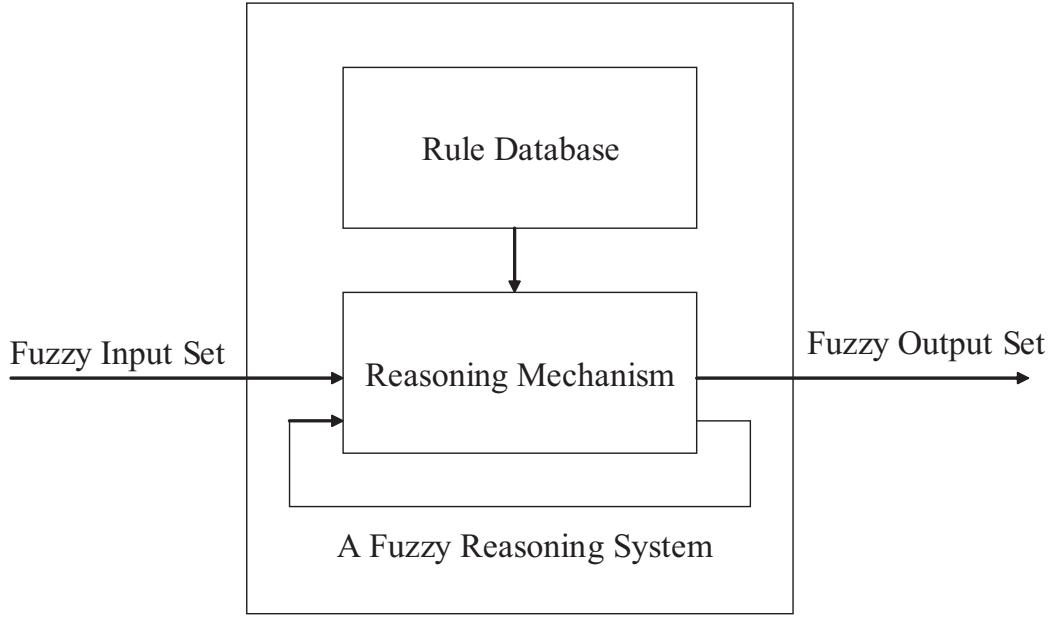


Fig. 2.27: *Block diagram for pure FIS (redrawn from Wu and Ling (2002)).*

Because the inputs and outputs of the pure FIS are all fuzzy sets however the inputs and outputs are accurate values in most engineering systems in reality, this kind of inference system can not directly be applied in practical engineering. For solving this problem, Mamdani et al. presented a kind of FIS with fuzzy generator and defuzzifier (Mamdani model). Takagi and Sugeno presented another kind of FIS in which the fuzzy rule consequence is a crisp value called Sugeno FIS.

Mamdani Fuzzy Inference Systems

Because Mamdani FIS adds fuzzy generator and defuzzifier to the input and output of pure FIS respectively, the input and output of FIS all are the crisp values. It can, therefore, be directly used in practical engineering. This FIS is called FIS with fuzzy generator and defuzzifier. Because of the wide applications, it is also simply called

FIS. The construct of this FIS is shown in Fig. 2.28

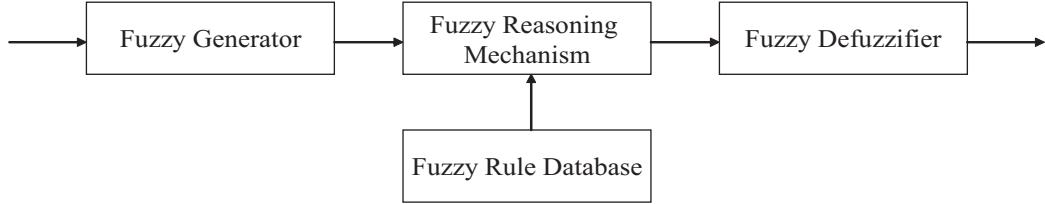


Fig. 2.28: *Block diagram for Mamdani FIS (redrawn from Wu and Ling (2002)).*

Mamdani FIS is the commonest algorithm. It was the first attempt to use the theory of fuzzy set to control systems. It was presented for controlling a steam and boiler by Mamdani in 1975. It adopted a set of linear linguistic control rules obtained from experienced human operators. It derived from Zadeh [?] thought about the application of complex systems and decision processes with fuzzy set.

In the application of engineering, it is expected that the reasoning output is a certain controlling scale value or other numeral value. However, the outputs reasoning from every rule in Mamdani FIS is the MF of variables or discrete fuzzy sets. The expected outputs are obtained after the conclusions of rules are aggregated and then defuzzification for each fuzzy set of output variables finishes.

The conclusion of every reasoning in Mamdani FIS is a distributed fuzzy set. However, it is more useful if the conclusion is a certain point in the region rather than a distributed set. This point is regarded as defuzzifying the fuzzy set. Because the method develops the effect, it greatly simplifies the calculation of Mamdani FIS. It obtains the centroid by using the average of summation with part of data instead of two dimension integral.

Mamdani FIS includes the following parts:

Fuzzy Rule Database

Fuzzy rule database consists of summation of several if-then rules. It is the backbone of FIS. The functions of other parts in the system are to explain and use these rules for solving the special problems. In general, fuzzy rules can be obtained by two approaches: from consulting experts and from learning algorithm based on the measured data.

Fuzzy Reasoning Mechanism

The function of fuzzy reasoning mechanism is to transfer if-then rule in fuzzy rule database to a mapping, which maps fuzzy sets in input space to fuzzy sets in output space. It mainly includes conjunction calculation, the expression of if-then rules, intuitive reasoning judgement, and some relative operations.

Conjunction calculation mainly includes three operations among AND, OR, and ALSO. The fuzzy implication of them mainly includes two algorithm of R_c and R_p

- Minimal rule for fuzzy implication

$$\mu_{A \rightarrow B}(x, y) = \min\{\mu_A(x), \mu_B(y)\}$$

- Product rule for fuzzy implication

$$\mu_{A \rightarrow B}(x, y) = \mu_A(x)\mu_B(y)$$

where A is the input of fuzzy reasoning mechanism, and B is the output of fuzzy reasoning mechanism.

Fuzzy Generator

The function of fuzzy generator is to map a special point into a fuzzy set in input space.

Defuzzifier

The goal of defuzzifier is to map a fuzzy set in output space to a special point to meet the practical application.

Takagi-Sugeno FIS

Takagi-Sugeno FIS is also called Sugeno FIS . It is a kind of special FIS. Its fuzzy rules are different from the common fuzzy rules. Usually, the antecedent conditions of fuzzy rules and the consequent conclusions of fuzzy rules are fuzzy linguistic values. They are shown as follow:

if x_1 is A_1 , x_2 is A_2 , \dots , then y is B .

where $A_i (i = 1, 2, \dots, n)$ are input fuzzy linguistic values, B is output fuzzy linguistic value.

However, in Sugeno FIS, the fuzzy rules are shown as follow:

$$\text{if } x_1 \text{ is } A_1, x_2 \text{ is } A_2, \dots, \text{then } y = c_0 + \sum_{i=1}^n c_i x_i.$$

where $A_i (i = 1, 2, \dots, n)$ are input fuzzy linguistic values, c_i is the parameters of certain values. Sugeno FIS is shown in Fig. 2.29

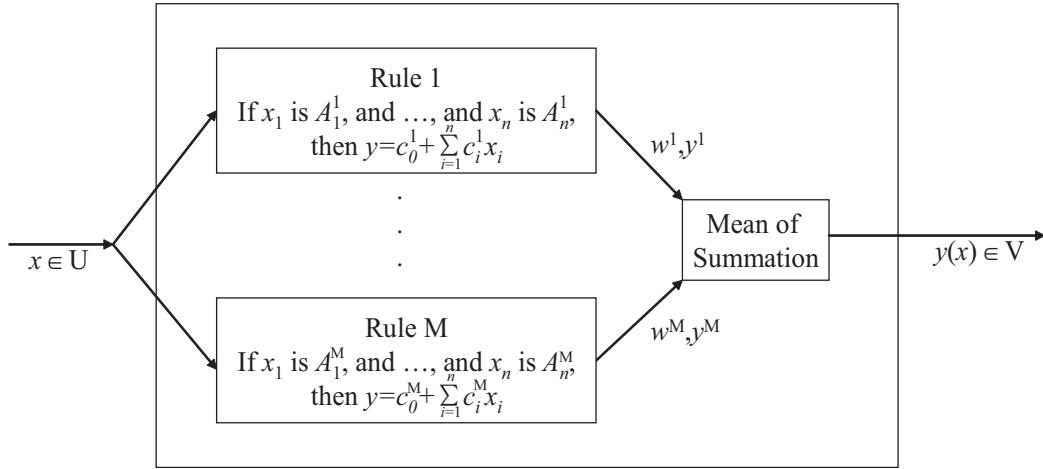


Fig. 2.29: *Block diagram for Sugeno FIS (redrawn from Wu and Ling (2002)).*

Compared with above equations, we can find that outputs of Sugeno FIS are still crisp values even though there is no defuzzifier. The advantage of this kind of FIS is that outputs can be described with the linear combination of inputs. It can decide the parameters of the system c_i ($i = 0, 1, 2, \dots, n$) by the method of estimating parameter. Meantime, It can approximately analyze and design FIS by using the analysis method of linear control system. The disadvantage of this kind of FIS is that the output part of the rules have no forms like fuzzy linguistic values. Therefore it can not fully use the control knowledge of experts. The applications of different rules of fuzzy sets are limited in this kind of FIS. It is similar to Mamdani FIS in many aspects. They are the same in the first two steps (fuzzification for input variables and aggregation of fuzzy sets). The main differences between them is that: for Sugeno FIS, the MFs of output variables in the conclusions are only linear functions of input variables or constant function; however, for Mamdani FIS, the output crisp values

are obtained by defuzzifying the output fuzzy sets.

For Sugeno FIS, if the MFs of output variables are the linear function of inputs, it is called a one order system. However, if the MFs are constant functions, it is called a zero order system.

In zero order Sugeno FIS, a typical fuzzy rule have a following form:

$$\text{if } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z=k$$

where A and B are fuzzy sets in the antecedent while k is an exact constant (non-fuzzification concept). If each rule's output conclusions is constant, this kind of Sugeno FIS is the same as Mamdani FIS. The only difference is that each MF of rule's outputs in zero order Sugeno FIS is a singleton set and the implication algorithm in the procedure of fuzzy reasoning and the aggregation algorithm of output are fixed. The implication algorithm in Sugeno FIS adopts simple product operation (here the conclusions of product and minimum operation are the same because the MF values are equal to 1 in singleton sets). The aggregation algorithm adopts simple summation of these singleton sets (the calculation results of the operations of *max*, *sum*, and *prob* are the same).

With more common used in application, fuzzy rules in first order Sugeno FIS generally have the following format:

$$\text{if } x \text{ is } A \text{ and } y \text{ is } B \text{ then } z=px + qy + r$$

where A and B are fuzzy sets in the antecedent while p , q and r are exact constant. It is regarded as the extension of zero order Sugeno FIS in which every rule defines the place of a dynamically moving singleton set. It seems that a zero output singleton set moves the place of a singleton set in output space as linear mode according to the inputs. Higher order Sugeno FIS is feasible in theory. However, it makes the model very complex and has no obvious advantage with respect to the solving problem. Therefore high order Sugeno FISs are few in application.

When Sugeno FIS applies to nonlinear systems, it can be an internal interpolation manager for multiple linear controllers because each rule has a linear relationship with the system input variables (integrated manager for multiple linear system). These linear controllers correspond to different work conditions (points). Sugeno FIS is an

excellent choice to smoothly finish interpolation of the increasing of linear matrix in input space. Essentially Sugeno FIS fits the core control mechanism of this high quality manager. It can also be used in nonlinear system modelling by linear interpolation mode.

Architecture of FIS and its Design Steps

A typical Mamdani FIS includes the following parts:

- (1) Input and output linguistic variables, including linguistic values and MF;
- (2) Fuzzy rules;
- (3) The methods of fuzzy generation and defuzzification;
- (4) The algorithm of fuzzy reasoning.

According to the architecture of FIS, the steps to construct a Mamdani FIS are as follows:

- (1) Decide input and output linguistic variables and linguistic values;
- (2) Decide MFs for linguistic values, including types and parameters of MF;
- (3) Decide the fuzzy rules;
- (4) Decide the methods of fuzzy operation, including method of fuzzy reasoning, method of fuzzifying, and method of defuzzifying.

2.4.2 Adaptive Network-based Fuzzy Inference Systems

The key characteristic of neural networks is its function of adaptive learning. Using this characteristic adaptive learning to analyze and model generates the technique of adaptive neural network. This is very effective tool for building the model of a fuzzy system.

The most important point of adaptive network-based fuzzy inference system (ANFIS) [?, ?, ?, ?, ?] is that the method of system modelling is based on data. MFs and fuzzy rules in ANFIS are obtained by learning a great deal of known data rather than by arbitrarily selecting a given experience or based on intuition. This is very important when working with systems whose characteristics are unknown or complex.

The shapes of MFs depend on parameters. Changing parameters changes their shapes and some important properties.

Overview of ANFIS and its Algorithm

Design of ANFIS does not depend on the models of objects. It depends considerably on experience and the knowledge of experts and operators. The construction of ANFIS is very suitable for the expression of quality or fuzzy experience and knowledge. These experiences and knowledge are represented by if-then fuzzy rules. If the experience is short, the design can not be expected to produce good control effect.

For the above problem, adaptive is a good solution. However, the adaptive method brings great difficulty and trouble to the design and construction of the system. It concerns many special profound theories. Also the applicability of different adaptive theories and methods is narrow. Thus the design and realization of ANFIS are pretty difficult.

On the other hand, the development of fuzzy sets and neural networks (NN) hasten the progress of intelligent control. They are two different areas. The differences between them in basic theory are great. However, they all belong to artificial intelligence. Theory and practice provide that they can be combined together. Adaptive network-based fuzzy inference system is also called adaptive neuro-fuzzy inference system (ANFIS).

Because fuzzy reasoning has no function of adaptive learning, its application is greatly limited. Neural network can not express fuzzy language and lacks transparency like a black box, so it can not express reasoning function like brains do. However, ANFIS can organically combine both together. Not only does it develop the advantages of both but it also compensates for the shortages of each one. FIS has

a obvious disadvantage that it lacks effectively learning mechanism. The outstanding property of ANFIS is that it compensates the disadvantage of FIS with the learning mechanism of NN.

Mamdani FIS and Sugeno FIS have their advantages and disadvantages. Because its form of rules fits the habits of human's thought and language expression, Mamdani FIS can easily express a human's knowledge. However, its disadvantages are that it involves complex calculation and also goes against mathematical analysis. On the other hand, simple calculation is the strongpoint of Sugeno FIS which benefits to mathematical analysis. It can easily be combined with Proportional, Integral, Derivative (PID) control, optimization and adaptive methods. We can obtain controllers with optimal and adaptive ability or tools of fuzzy modelling by Sugeno FIS. According to its properties, it is used to construct ANFIS with NN, which has adaptive learning ability. The combination of fuzzy logic and NN is an important research area of intelligent computing. ANFIS which combines fuzzy logic and NN has the advantage that fuzzy set is easy to express human's knowledge and NN has the ability of message storage with distribution and learning. It provides an effective tool for modelling and control in complex systems.

In theory, the architectures of ANFIS for standard Mamdani model and Sugeno model have perfect design methods.

Correspondent to first order Sugeno model, Jang et al. [?] presents ANFIS similar to first order Sugeno FIS. The architecture of ANFIS with two inputs is shown in Fig. 2.30. Each node in every layer has same function.

For simplicity, we assume FIS has two inputs x and y and one output z . For a first order Sugeno FIS, [?, ?, ?], a common rule set with two fuzzy if-then rules is as follow:

Rule 1: If x is A_1 and y is B_1 , then $f_1 = p_1x + q_1y + r_1$.

Rule 2: If x is A_2 and y is B_2 , then $f_2 = p_2x + q_2y + r_2$.

First Layer: Node i in this layer is denoted as square nodes (Parameters in this

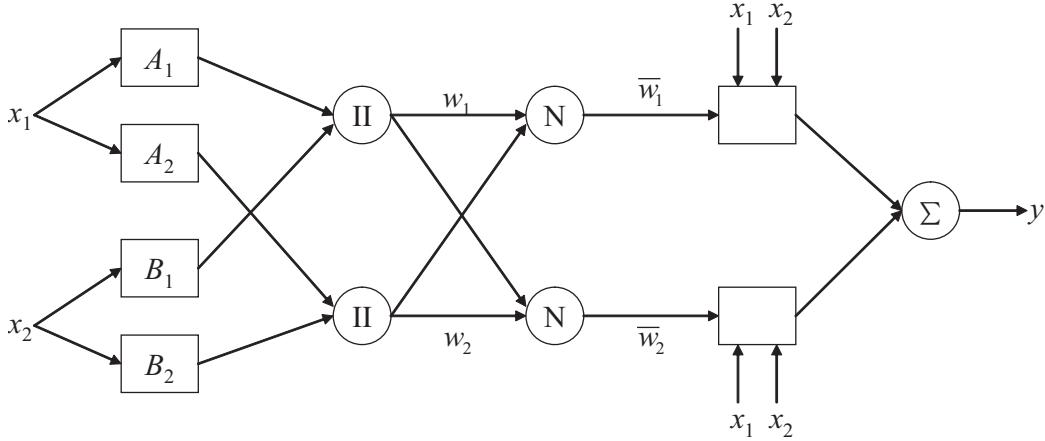


Fig. 2.30: The architecture of ANFIS for two-input first order Sugeno fuzzy model with two rules (redrawn from Jang, Sun and Mizutani (1997)).

layer are changeable). Here $O_{l,i}$ is denoted the output of the i th node in layer l .

$$O_{1,i} = \mu_{A_i}(x_1), \quad i = 1, 2 \quad (2.63)$$

$$O_{1,i} = \mu_{B(i-2)}(x_2), \quad i = 3, 4 \quad (2.64)$$

where x_1 (or x_2) is the input of i th node. A_i (or B_{i-2}) is a linguistic variable (such as “big” or “small”) associated with this node. Or it is said that $O_{1,i}$ is the membership function of a fuzzy set $A (= A_1, A_2, B_1 \text{ or } B_2)$. It specified the degree to which the given input x_1 (or x_2) satisfies the quantifier A . Here the membership function for A can be any appropriate parameterized membership function, such as the function of product by two sigmoid functions:

$$\mu_A(x) = \frac{1}{(1 + e^{-a_1(x-c_1)})(1 + e^{-a_2(x-c_2)})} \quad (2.65)$$

where parameters a_1 , a_2 , c_1 and c_2 decide the shape of two sigmoid functions. Parameters in this layer are referred to as premise parameters.

Second layer: the nodes in this layer are labelled by \prod in Fig. 2.30. The outputs are the products of inputs

$$O_{2,i} = w_i = \mu_{A_i}(x_1)\mu_{B_i}(x_2), \quad i = 1, 2 \quad (2.66)$$

Each node output represents the firing strength of a rule. Usually any other T-norm operators which perform fuzzy AND can be used as the node function in this layer.

Third layer: The nodes in this layer are labelled with N . i th node calculates the ratio of i th rule's firing strength to the summation of all rules' firing strengths.

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \quad (2.67)$$

The outputs of this layer are named as normalized firing strengths.

Fourth layer: Each node in this layer is called an adaptive node. The outputs are

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x_1 + q_i x_2 + r_i), \quad i = 1, 2 \quad (2.68)$$

where p_i , q_i , r_i are parameters of the nodes. Parameters in this layer are called consequent parameters.

Fifth layer: Every node in this layer is a fixed node labelled \sum . Its total output of the summation of all input is

$$O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (2.69)$$

ANFIS Design

ANFIS uses the mature parameter learning algorithm in neural networks — the back-propagation algorithm or the BP least square method (LSM). It adjust the shape parameters of MF in FIS by learning from a set of given input and output data.

Learning and Reasoning Based on the Model of ANFIS

The algorithm of ANFIS is very simple. It provides a learning method which can obtain corresponding messages (fuzzy rules) from data for fuzzy modelling. This learning method is very similar to the learning algorithm of NN. It can effectively calculate the best parameter for MF. It designs a Sugeno FIS which can best simulate the expected or actual relations between inputs and outputs. ANFIS is a modelling method based on the given data. It is the best measuring criterion whether or not the results of FIS model can simulate the data well.

Adjustment of Structure and Parameters for FIS

The structure in Fig. 2.30 is similar to the structure of NN. It first maps inputs with the MFs of inputs and parameters. Then it maps the data of input space to output space with the MFs of output variables and parameters.

The parameters determining the shapes of MFs can adjust and change by learning procedure. The adjustment of these parameters is accomplished by a gradient vector. It is to appraise to a set of certain parameters how FIS meets the data of inputs and outputs. Once this gradient is constructed, the system can adjust these parameters to reduce the error between it and the expected system by optimal algorithm (This error is usually defined as the square of difference between the outputs and targets).

The functions of ANFIS can estimate the parameters of MFs by BP algorithm or the combination of the LSM estimation with BP algorithm.

Validity of the Training Data and the Resulting Model

The modelling procedure of ANFIS is familiar to the method of system recognition. First, ANFIS assumes a parameterized model structure. This model connects to input variables, the MF of input variables, fuzzy rules, output variables and the MF of output variables. Then it obtains a group of the pair of input and output data and composes training data to the algorithm of ANFIS according to a certain format. Now it can train the previous parameterized FIS model by ANFIS functions and adjusts the parameters of MFs according to special rules of error. Finally it makes this model continuously approximate (simulate) to the given training data.

This modelling method can usually obtain good results if the data reflects well the model characteristics. However, in application this is not always the case. Sometimes the training data can not represent all the characteristics of the system if the training data include noisy signals. Measuring the validity of the training procedure and the resulting model is a very important procedure.

Usually not all data are expected to train the system in ANFIS modelling. Because the calculation of ANFIS increases much more than the ratio of increasing training data, it absolutely increases the calculation greatly by using vast training data and consumes time. Also more importantly learning of NN does not always converge toward the optimal direction. Training results become poor with the increase of training data in some cases. In worse cases, the system changes the structure by increasing its complexity to reflect the properties of a few of data with large noisy signals. This result must be avoided when the ANFIS method is used. Thus the data

must be separated in three groups: The first group is the training data set which trains the model; the second group is the checking data set which checks the model in the training procedure; and the third group is the testing data set which tests the results of the model.

Measuring the Result Model with Checking and Testing Data Sets

The choice of training data can result in the inclusion of some unreliable factors. Besides the necessity of doing some preprocessing work on the obtained data, checking during the training procedure and the final result model also are important. In general, the checking procedure of the result model uses those input and output data, which are not used to train the system to compare with the trained model to see whether it can match and predict those data. This procedure usually uses the called testing data set to finish. In ANFIS function, we also use another called checking data set to control the training procedure and find the result model overfitting. If the training data is limited compared with the model parameters, the model can simulate the training data well after training some epochs. However, if the model is trained continuously, it will have some characteristics of overfitting. This model deviates from the expected results. For example, if we train NN to simulate a second order curve by the given data, the training results obtain a sixth order curve. Even though the errors are fewer for the training data, the system can not work well for some other data.

The checking data input ANFIS function with the training data simultaneously. When the training data is sparse, the system will converge to multiple directions. The function will choose the suitable results according to the rule resulting in the fewest errors of the checking data. The checking data do not directly participate to train the parameters of system as training data. Their values only help for judgement and choice.

This case is a typical problem in adaptive technique: It is important to select not only the expected model which represents the need of the system but also the effective checking data differing from the training data (to avoiding useless checking). For those systems which have plentiful reliable data and represent the properties of

model, it is easy to choose enough training data. However, if the chosen training data can not express the characteristics of the model or include a lot of noise, it is necessary to use checking data because it can avoid generation of the structure of overfitting for simulating the characteristics of noise.

The checking data are different from the testing data. The testing data do the calculation and comparison correspondingly after the final model obtains the results. However, the usage and calculation of the checking data are simultaneous with the training data. If the system experiences the case of not matching (for example the training data is too small), the errors of the training data become fewer and fewer and the errors of the checking data become larger and larger when the training continues to some extent. If the above case appears, we know the system parameters are not matching the training data or overfitting.

Chapter 3

Proposed Method for Two Dimensional Image Denoising with ANFIS Algorithm

In this chapter, we derive the 2 dimensional counterparts of 1 dimensional signals, and then we use them in image processing applications.

3.1 Transformation from 2-D Image Matrix to 1-D Sequence Vector

As we know before, a 2 dimensional signal is described as a matrix shown in Fig. 3.1. If we connect the rows (or columns), then we have a long vector, that is, a 1 dimensional sequence shown in Fig. 3.2. We assume a 2 dimensional image matrix has M rows and N columns. Then after we change the 2 dimensional image matrix to a 1 dimensional vector, we have

$$s(i) = S(j, k) \quad (3.1)$$

where $j = 1, 2, \dots, M$, $k = 1, 2, \dots, N$, and $i = 1, 2, \dots, M \times N$; S is the original image matrix; s is the 1 dimensional transformed sequence vector.

$$i = j \times N + k \quad (3.2)$$

Then we can use a 1 dimensional conventional ANFIS algorithm.

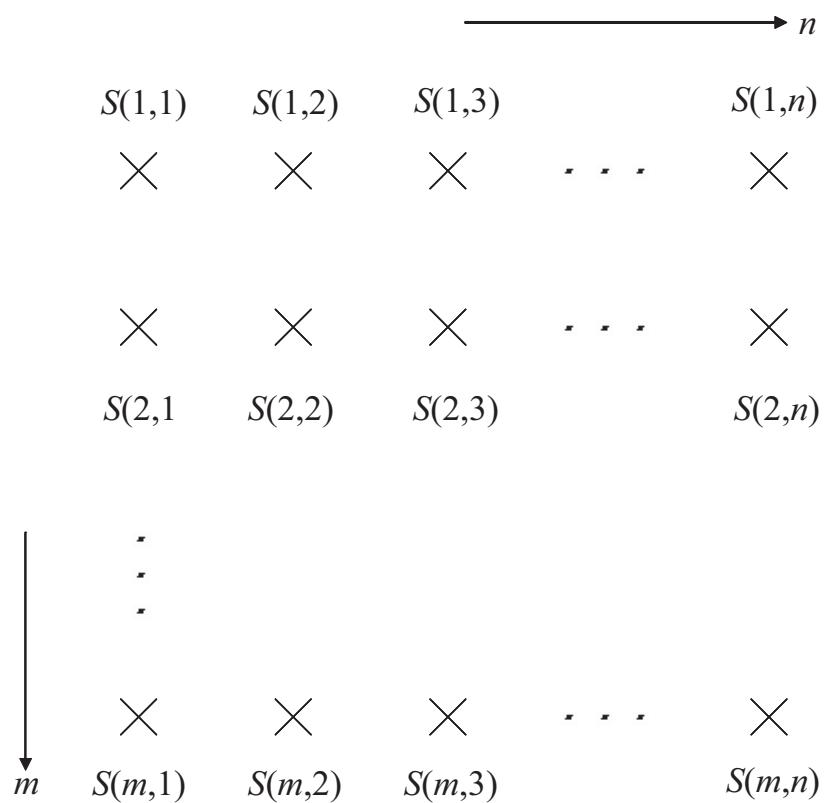


Fig. 3.1: A 2 dimensional image matrix with M rows and N columns (redrawn from Bose (2003)).

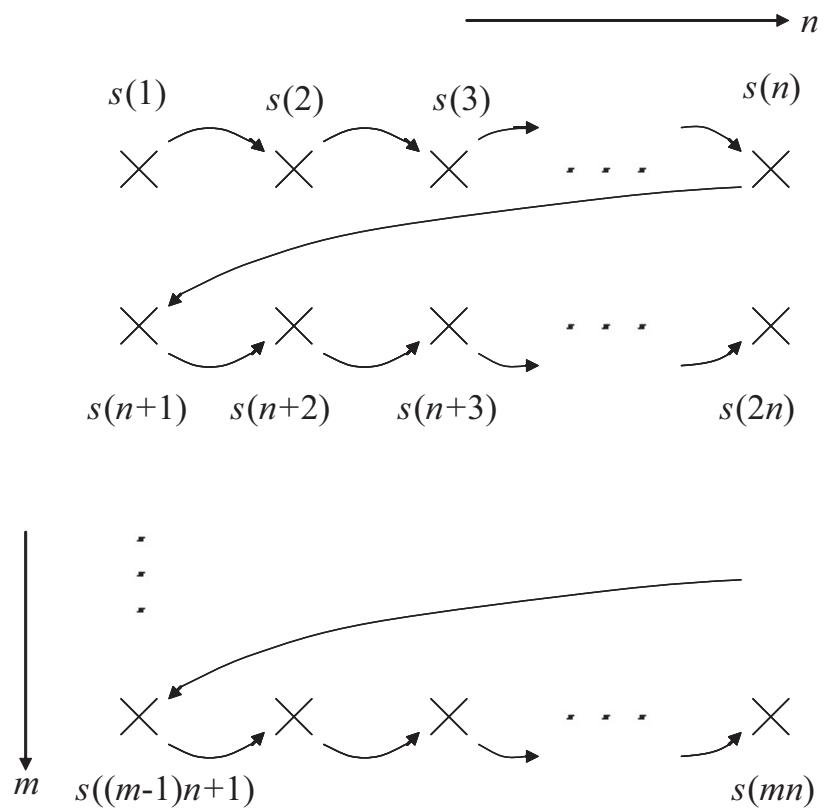


Fig. 3.2: A 1 dimensional sequence vector after a 2 dimensional image matrix is connected with the row (or columns) (redrawn from Bose (2003)).

3.2 Generating and Adding Noises into the Processed Images

Now we generate and add different types of noise and different intensity of noise.

For Gaussian noise, we choose different mean, variance, standard deviation from a normal distribution to generate different intensity of noise. We divide them into three categories according to the signal noise ratio (SNR): Low noise, medium noise, and high noise.

For salt & pepper noise, it is a kind of impulse noise. The range of pixel value for a grey image is from 0 to 255. The maximum, 255, is denoted as white color and the minimum, 0, black color. The pixel value for salt noise is 255 because salt is white color and similarly the pixel value for pepper noise is 0 because pepper is black color. The intensity of salt & pepper noise is measured by the probability of occurrence in each pixel. If the probability of salt & pepper noise is high, its intensity is strong. Otherwise, the intensity is weak when its probability is low. After we generate the noises, we add them to images for simulating the practical case.

We also save these noises for comparison with the conventional filtering techniques. If we always randomly generate noises in different programs of ANFIS or the conventional filtering techniques, these noises will have some slight changes in their means, variances and standard deviations. Therefore, for increasing comparability, it is better to save these noises rather than randomly generate them every time in different programs of ANFIS or the conventional filtering techniques.

3.3 Nonlinear Adaptive Noise Cancellation with ANFIS

The typical application of adaptive filter is to work as adaptive noise cancellation. It is the transformation of the best filtering and is very excellent in a lot of applications. Adaptive noise cancellation is used when the signals is very weak or can not be

measured in noise fields. It will filter the auxiliary signals or the reference signals from a or several sensor(s) and substrates them from the original signals including signals and noise. As a result, the original noise is attenuated or eliminated owing to cancellation.

Adaptive noise cancellation was first presented by Widrow and Glover in 1975 [?]. The goal is to filtrate an interference component between a measurable noise source and the corresponding unmeasurable interference by a linear model. Adaptive noise cancellation has successfully been used in the following real world applications:

- Interference cancelling of electrical source in electrocardiograms (ECGs);
- Measurement of fetal heartbeat by filtering out mother's heartbeat and interference;
- Speech ingestion of special person from interference of other voice;
- Antenna sidelobe interference cancelling;
- Echo elimination on long-distance telephone transmission lines.

The concept of linear adaptive noise cancelling can be extended to the nonlinear area by using nonlinear adaptive systems. ANFIS can be used to recognize an unknown nonlinear passage dynamics which transmits a noise source into an interference component in a detected signal. Sometimes the proposed nonlinear adaptive cancelling technique is more suitable to filter some noise component than noise cancelling techniques based on frequency-selective filtering.

The schematic diagram of an ideal situation of adaptive noise cancellation is shown in Fig. 3.3.

Here an information signal $s(i)$ is unmeasurable and a noise source signal $v_1(i)$ is measurable. The noise source goes through an unknown nonlinear function to generate a distorted noise $v_0(i)$. It is then added to an information signal $s(i)$ to compose a output signal $x(i)$ which is measurable. Our goal is to recover the information signal $s(i)$ from the compound output signal $x(i)$, which forms the information signal $s(i)$

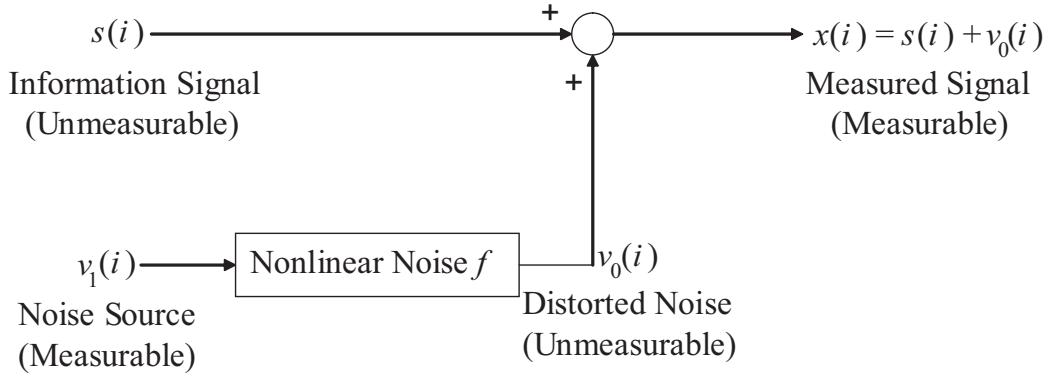


Fig. 3.3: *Schematic diagram of adaptive noise cancellation without ANFIS (redrawn from Jang, Sun and Mizutani (1997)).*

plus $v_0(i)$, a distorted and delayed version of $v_1(i)$. The detected output signal is presented as

$$x(i) = s(i) + v_0(i) = s(i) + f(v_1(i), v_1(i-1), \dots) \quad (3.3)$$

where the function $f(\cdot)$ denotes the nonlinear function that the noise source signal v_1 goes through. If we know the function $f(\cdot)$ exactly, we can easily retrieve the original information signal by $x(i)$ subtracting $v_1(i)$ directly. However, $f(\cdot)$ usually is unknown in advance and may change with time. The spectrum of $v_0(i)$ may overlap that of $s(i)$ at some extent. The result is poor when the common frequency-domain filter is adopted.

The noise signal $v_1(i)$ does not relate with the information signal $s(i)$ for estimating the distorted noise signal $v_0(i)$. However, it can not be measured directly because it is a part of the overall measurable signal $x(i)$. The detected signal $x(i)$ can be measured as the expected output of ANFIS training only if the information signal $s(i)$ is zero mean and not correlated with the noise signal $v_1(k)$ shown in Fig. 3.4

Adaptive noise cancellation has two inputs: the detected output signal, also called the original input $x(i)$, and the noise source signal, also called the reference input $v_1(i)$. The reference input $v_1(i)$ is correlated to the distorted noise signal $v_0(i)$ and not correlated to the information signal $s(i)$. ANFIS accepts the error signal e to control and adjust the weights W which make the output of ANFIS, denoted as

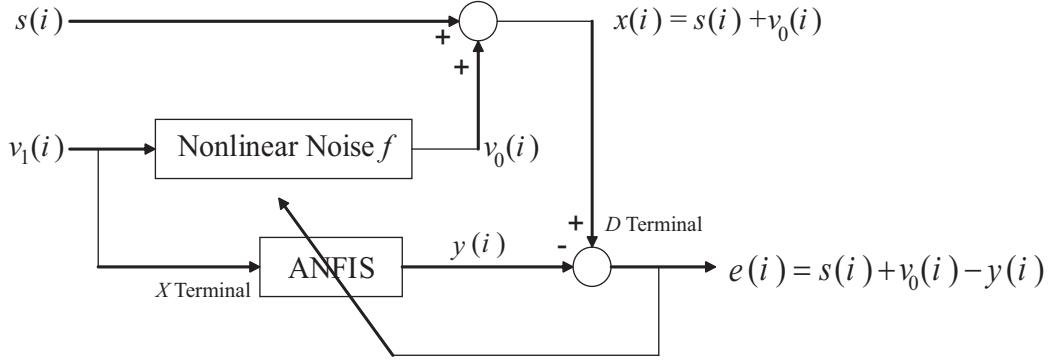


Fig. 3.4: *The architecture of ANFIS for noise cancellation (redrawn from Jang, Sun and Mizutani (1997)).*

$y(i)$, approximating to the distorted noise signal $v_0(i)$ in the original signal $x(i)$. So $e(i)=x(i)-y(i)$ will approach to the information signal $s(i)$.

Now we prove the conclusion. Assume that $s(i)$, $v_0(i)$ and $v_1(i)$ are smooth random signals with a zero mean. $y(i)$ also has this property. Because $s(i)$ is not correlated to $v_0(i)$ and $v_1(i)$,

$$\begin{aligned} e(i) &= x(i) - y(i) \\ &= s(i) + v_0(i) - y(i) \end{aligned} \tag{3.4}$$

We square the above equation and obtain

$$\begin{aligned} e(i)^2 &= (x(i) - y(i))^2 \\ &= (s(i) + v_0(i) - y(i))^2 \\ &= (s(i) + v_0(i) - \hat{f}(v_1(i), v_1(i-1), v_1(i-2), \dots))^2 \end{aligned} \tag{3.5}$$

where \hat{f} is the function implemented by ANFIS.

Since $s(i)$ is not related to $v_1(i)$ and its previous values, ANFIS do not know how to remove the noise attributing to $s(i)$. This means that the information signal $s(i)$ includes an unrelated “noise” component in the data fitting processing. ANFIS can not solve this problem for it at all except obtaining its steady-state value. What ANFIS can do is to minimize the error component of $(v_0(i) - y(i))$ — that is $(v_0(i) -$

$\hat{f}(v_1(i), v_1(i-1), v_1(i-2), \dots)^2$. Eq.(3.5) can be expanded to

$$e(i)^2 = (s(i))^2 + (v_0(i) - y(i))^2 + 2s(i)v_0(i) - 2s(i)y(i) \quad (3.6)$$

Taking means at both sides of Eq.(3.6) yields

$$E[e^2] = E[s^2] + E[(v_0 - y)^2] + 2E[sv_0] - 2E[sy] \quad (3.7)$$

Because $s(i)$ is not correlated to $v_0(i)$, $v_1(i)$ and $y(i)$, $E[sv_0]$ and $E[sy]$ are equal to zero.

$$E[e^2] = E[s^2] + E[(v_0 - y)^2] \quad (3.8)$$

where $E[s^2]$ is no change when ANFIS adjust its MFs to minimize $E[e^2]$ because it do not relate with weight W . So minimizing $E[e^2]$ is equal to minimizing second item, $E[(v_0 - y)^2]$, in Eq.(3.8) such that the ANFIS function $\hat{f}(\cdot)$ approaches the passage dynamics $f(\cdot)$ as much as possible.

$$E[e^2]_{min} \iff E[(v_0 - y)^2]_{min} \quad (3.9)$$

The information signal $s(i)$ we want to retrieve also acts as additive “noise” in ANFIS training. For simplification of the discussion, we assume that

1. The information signal $s(i)$ is a zero signal for all i ;
2. We use the least-square method to handle premise parameters and updated consequent parameters of ANFIS.

Item 1 means that we can obtain ideal training data which have an effect only on the measurement noise. Item 2 means only linear parameters are used in ANFIS. ANFIS with changeable parameters will still generate a fitting error $e(i)$ which is equal to the difference between an expected output and the ANFIS output even using ideal training data and this error $e(i)$ is generated by measurement noise and/or modelling errors. If the error $e(i)$ is zero expectation, the consequent parameters obtained by the least-squares method in ANFIS are unbiased, which is the characteristic of the linear least square estimator (LSE).

The previous assumption can be loosened. It is stated that $s(i)$ is a zero signal in item (1). However, this is unrealistic in the real world. Because $s(i)$ is an additive component, we can still recognize unbias consequent parameters by LSE only if $s(i)$ is zero expectation.

It is required that ANFIS only updates its consequent parameters in item (2). We use the proposed hybrid learning rule — that is the backpropagation learning rule with LSE to update the premise (nonlinear) and consequent (linear) parameters in the simulations. Then we use the backpropagation learning rule compared with the hybrid learning rule. Even though ANFIS is a nonlinear model and the following Gauss-Markov theorem is no longer held, the system has the capacity of reducing modelling errors further.

Theorem 3.3.1. *Gauss-Markov theorem: LSE is unbiased and minimumvariance.*

Theorem 3.3.2. *Gauss-Markov theorem: LSE is consistent.*

If a linear filter is used to replace the ANFIS block in Fig. 3.4, the original adaptive linear noise cancellation is similar to the settings proposed by Widrow and Glover [?]. We can handle a wide range of nonlinear passage dynamics by replacing the linear filter with a nonlinear ANFIS filter.

Before we do some simulations, we need to emphasize the conditions under which adaptive noise cancellation is valid:

- The information signal $s(i)$ should be unrelated to the noise signal $v_1(i)$ which is available.
- The information signal $s(i)$ must be zero mean.
- The order of the passage dynamics can be obtained from experiment and observation so that the number of inputs to the ANFIS filter is determined.

3.4 Restoration of Images Corrupted by the Noise of the Nonlinear Passage Dynamics with ANFIS

In this thesis, we propose to discuss separately the restoration of an image corrupted by noise in nonlinear passage dynamics of orders 2 and order 3.

3.4.1 Restoration of Images Corrupted by the Noise of the Nonlinear Passage Dynamics of Order 2 with ANFIS

We will use ANFIS to restore an image corrupted by noise in a nonlinear passage dynamics of orders 2. In the experiment, the unknown nonlinear passage dynamics are assumed to be defined as

$$v_0(i) = f(v_1(i), v_1(i-1)) = \frac{4 \sin(v_1(i)) v_1(i-1)}{1 + (v_1(i-1))^2} \quad (3.10)$$

where $v_1(i)$ is a noise source and $v_1(i-1)$ is the one order delay of the noise source, $v_0(i)$ is defined as the resultant of the nonlinear passage dynamics $f(\cdot)$ owing to $v_1(i)$ and $v_1(i-1)$, i is from 1 to the number of the pixels in the image.

3.4.2 Restoration of Images Corrupted by the Noise of the Nonlinear Passage Dynamics of Order 3 with ANFIS

In the proposal experiment, the unknown nonlinear passage dynamics in order 3 will also be assumed to be defined as

$$\begin{aligned} v_0(i) &= f(v_1(i), v_1(i-1)) \\ &= \frac{8 \sin(v_1(i)) v_1(i-1) v_1(i-2)}{1 + (v_1(i-1))^2 + (v_1(i-2))^2} \end{aligned} \quad (3.11)$$

where $v_1(i)$ is a noise source and $v_1(i-1)$ and $v_1(i-2)$ is the one unit delay and two unit delay of the noise source, $v_0(i)$ is defined as the resultant of the nonlinear

passage dynamics $f(\cdot)$ owing to $v_1(i)$, $v_1(i-1)$ and $v_1(i-2)$, i is from 1 to the number of the pixels in the image.

That the nonlinear passive dynamics of order 2 and 3 are discussed is because these two cases are typical in the corrupted images. For higher order components, their affection dramatically reduce with the increasing of delay time in the whole noise. Therefore, we do not discuss the nonlinear passive dynamics of order higher than 3.

3.4.3 Other Modifications for 2-D Image Denoising with ANFIS Algorithm

To satisfy one of the conditions for ANFIS that the information signal $s(i)$ must be zero mean, we need modify pixel values of images. Because the pixel values are between 0 - 255, the mean of the pixel values is 122.5. We need subtract each pixel with 122.5 to obtain zero mean for using ANFIS algorithm.

For the operation between random Gaussian noises and image pixels, we need also change each pixel value from signed 8-bit integer to double-precision. After removing the noise part from each pixel value detected by ANFIS, the pixel values need be transformed back to signed 8-bit integer for display.

3.4.4 Other Factors Discussed in the Proposed Method

We also propose to discuss the Gaussian noise with different intensity and salt & pepper noise with different probability to the corrupted images. We want to find how the intensity or probability of the different types of noise affect the denoising results.

We propose to inspect 8 types of membership function which are bell MF, triangle MF, Gaussian MF, two sided MF, pi-shaped MF, product of two sigmoidal MFs, difference of two sigmoidal MFs and trapezoidal MF.

We also will investigate other parameters such as the training epochs, the number of MFs for each input, the optimization method, the type of output MFs and the training data and checking data.

3.5 Transformation from 1-D Sequence Vector Back to 2-D Image Matrix

After we finish the restoration of images corrupted by the noise of the nonlinear passage dynamics with ANFIS, we need recover the image from 1 dimensional sequence vector back to 2 dimensional image matrix.

If we divide N in both side of Eq. (3.2), we can get j^* , the number of the row, and k^* , the number of the column from i^* , the number of 1 dimensional sequence. The number of the row j^* gets from the integer of $\frac{i^*}{N}$ and k^* from the numerator or residue of $\frac{i^*}{N}$.

$$j^* + \frac{k^*}{N} = \frac{i^*}{N} \quad (3.12)$$

where $j^* = 1, 2, \dots, M$, $k^* = 1, 2, \dots, N$, and $i^* = 1, 2, \dots, M \times N$.

after that, we get the recovered image matrix from the 1 dimensional transformed sequence vector denoised by ANFIS.

$$\hat{f}(j^*, k^*) = \hat{s}(i^*) \quad (3.13)$$

\hat{f} is the recovered image matrix; \hat{s} is the 1 dimensional transformed sequence vector denoised by ANFIS.

Chapter 4

Results of Image Denoising with ANFIS to Different Types of Nonlinear Passage Dynamics and of Noises

In this chapter, we display the results of image denoising with ANFIS to different types of nonlinear passage dynamics and of noises. The types of nonlinear passage dynamics are order 2 and order 3 because the effect of higher order components dramatically reduce with the increasing of delay time in the whole noise. We also show the different types of noises: Gaussian noise and salt & pepper noise in the different intensity.

4.1 Application of ANFIS to the Nonlinear Passage Dynamics of Order 2 to Restore an Image Corrupted by Noise

Now we use ANFIS to restore an image corrupted by noise into the nonlinear passage dynamics of order 2. In the experiment, the unknown nonlinear passage dynamics is assumed to be defined as

$$v_0(i) = f(v_1(i), v_1(i-1)) = \frac{4 \sin(v_1(i)) v_1(i-1)}{1 + (v_1(i-1))^2} \quad (4.1)$$

where $v_1(i)$ is a noise source and $v_1(i-1)$ is the one order delay of the noise source, $v_0(i)$ is defined as the resultant of the nonlinear passage dynamics $f(\cdot)$ owing to $v_1(i)$

and $v_1(i - 1)$, i is from 1 to the number of the pixels in the image.



Fig. 4.1: *The original image: (a) The original color image; (b) The original gray image.*

Fig 4.1(a) shows the original RGB (red, green and blue) color image ‘*Matirapoint*’. Because the original color image is too big and uses out of all the available computer memory, we downsize the original image to 0.25 times. The original RGB color image is transferred to the gray image in order to reduce the time necessary for computing. We use the *bicubic* interpolation method to resize the image. The resulting image after processed is shown in Fig 4.1(b).

The number of MFs each input is generally assigned is two, output MF type is *linear*, the default type of MF is *bell*, and training epoch number is 20. In general, the initial step size, step size decrease rate, and step size increase rate are 0.1, 0.9 and 1.1 respectively. A combination of least-squares and backpropagation gradient descent methods are used for training MF parameters to model a given set of input/output data. The output membership function type is *linear*.

4.1.1 Application of ANFIS to Restore an Image Corrupted by Gaussian Noise

We compare the effects of ANFIS under the intensity of low noise, medium noise and high noise for which the variances are 175.3222, 1577.9 and 4383.1 respectively.

The signal noise ratios (SNR) are equal to 12.7131, 3.1707 and -1.2663 to reflect the intensity of low, medium and high Gaussian noise respectively.

The measurable noise is Gaussian noise (a normal distribution) with mean zero, variance one and standard deviation one as shown in Fig. 4.2(a). For comparison purposes, all the different MFs use the same source noise $v_1(i)$.

We collect all the pixels as training data which have the following data pairs:

$$[v_1(i), v_1(i-1); x(i)], \quad (4.2)$$

The distorted noise $v_0(i)$ caused by the source noise $v_1(i)$ and produced by the nonlinear dynamics in Eq. 4.2 is shown in Fig. 4.2(b). We only show the high noise for the representation of the different intensity noise. The estimated distorted signal $y_0(i)$ is shown in Fig 4.2(c). The error between the estimated distorted noise $y(i)$ by ANFIS and the distorted noise $v_0(i)$ is shown in Fig. 4.2(d). In Fig. 4.2(d), the color shows almost pure black. It means the error between the estimated distorted noise $y(i)$ by ANFIS and the distorted noise $v_0(i)$ is very small since the pixel value of black in 8-bit grey image is denoted as 0.

We now investigate the behavior of these signals in the frequency domain before we proceed to the next step. Fig. 4.3(a) to 4.3(d) display the spectral density distributions of $s(i)$, $v_1(i)$, $v_0(i)$ and $x(i)$, respectively, from the first 256 points. Obviously, the spectra of the information signal $s(i)$ and the distorted noise $v_0(i)$ overlap each other in a large frequency area. This makes it impossible to apply common frequency domain filtering methods to remove $v_0(i)$ from $x(i)$. Fig 4.4 is the ANFIS surface $\hat{f}(.)$ after 20 epochs of batch learning.

To visually display the measurable signal $x(i)$ at the receiving end after adding one dimensional distort noise $v_0(i)$, we change the two dimensional pixel matrix into one dimensional vector $s(i)$. The estimated errors with ANFIS corrupted by low, medium and high Gaussian noise respectively are shown in Fig. 4.5. We find that with the increasing of noise, the error between the estimated distorted noise $y(i)$ and the distorted noise $v_0(i)$ becomes larger.

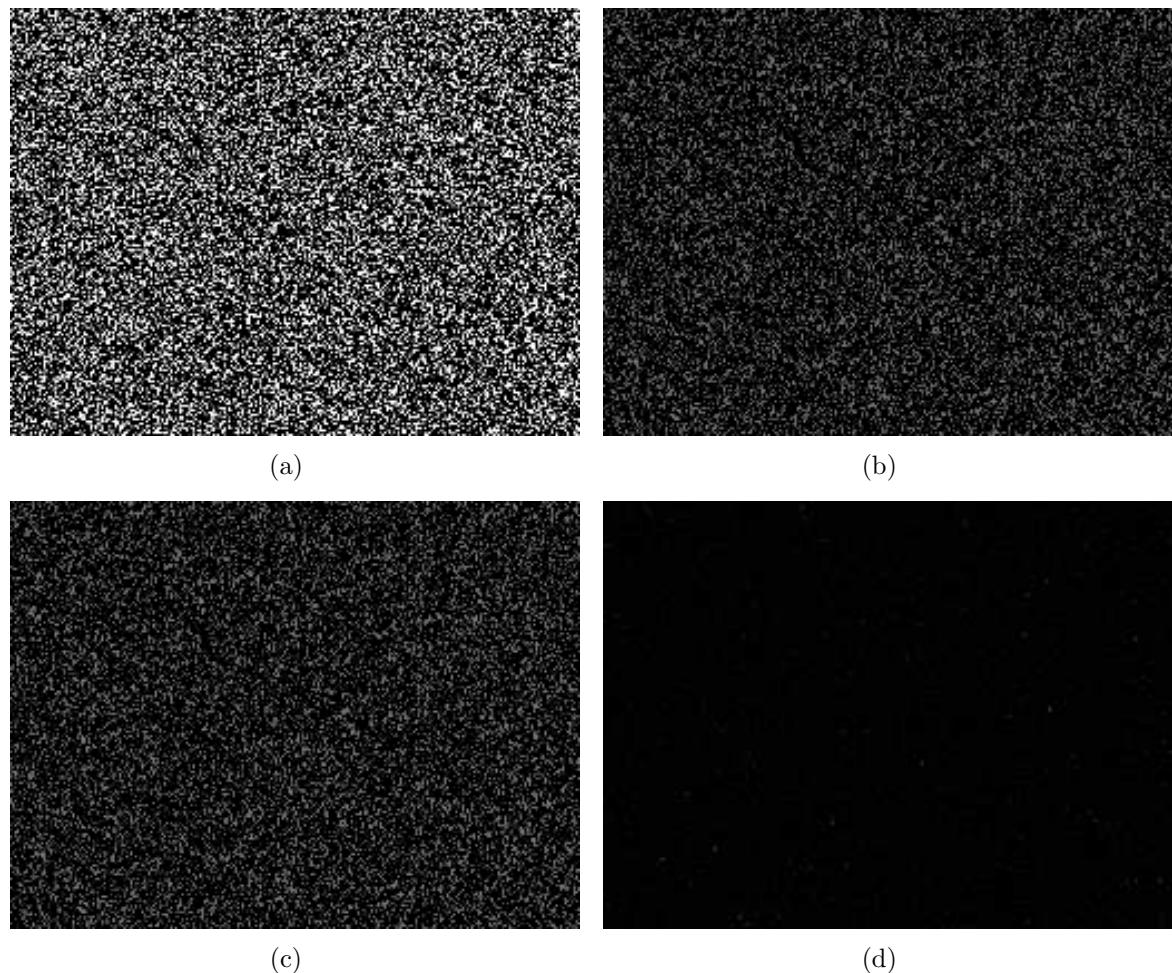
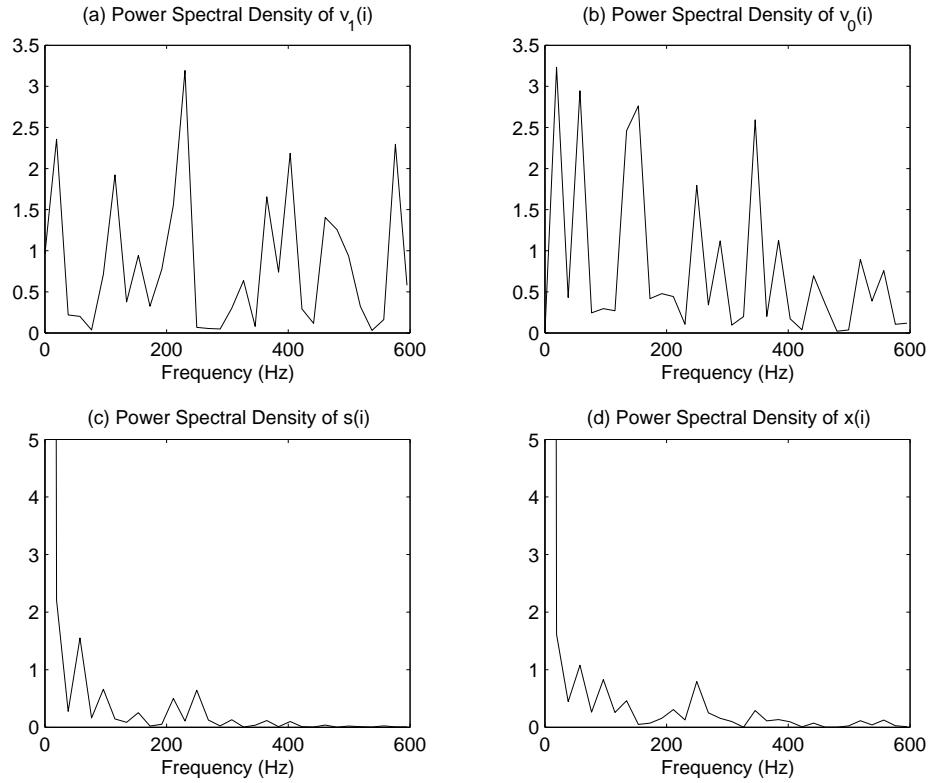
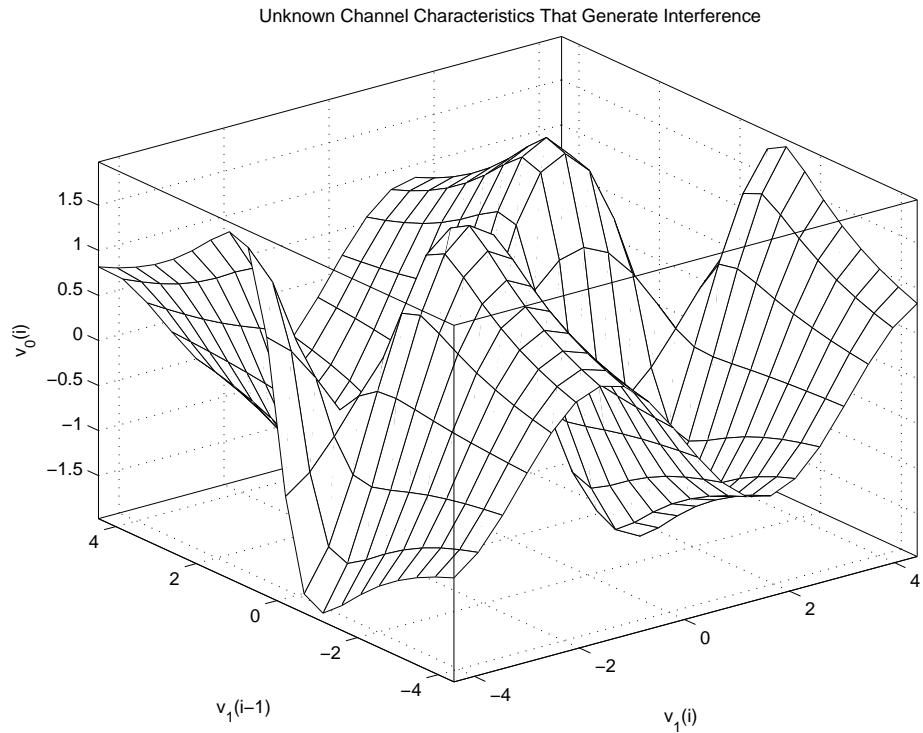


Fig. 4.2: *The signals being used in ANFIS: (a) The measurable source noise $v_1(i)$; (b) The distorted noise $v_0(i)$; (c) The estimated distorted noise $y(i)$ by ANFIS; (d) The error between the estimated distorted noise $y(i)$ by ANFIS and the distorted noise $v_0(i)$.*

Fig. 4.3: *Spectral density distributions.*Fig. 4.4: *The characteristics of ANFIS function \hat{f} .*

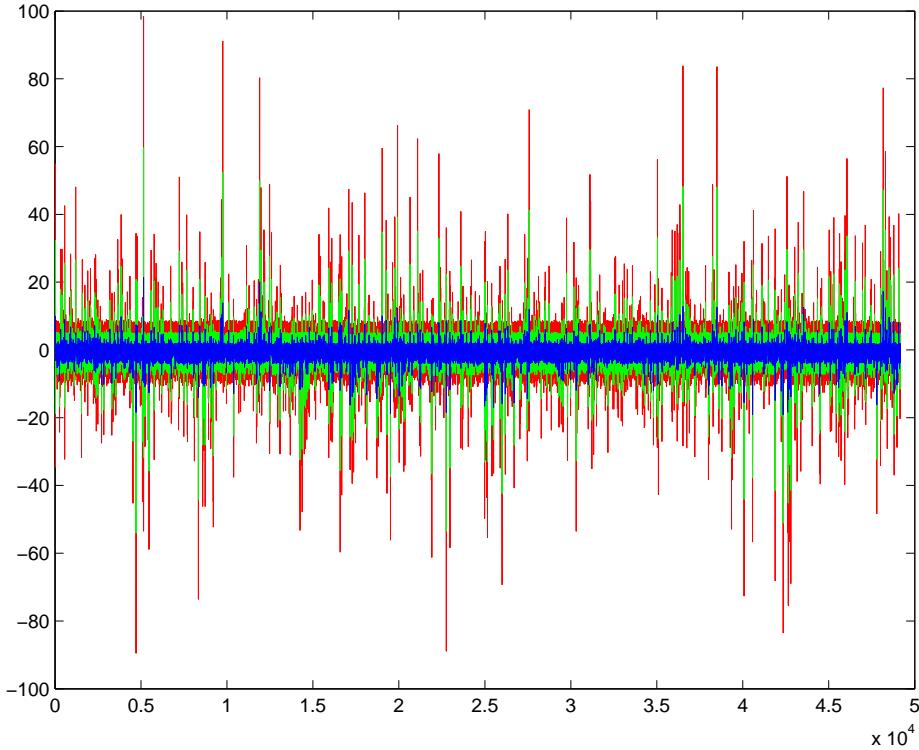


Fig. 4.5: *The estimated errors with ANFIS corrupted by low, medium and high Gaussian noise.*

4.1.2 Application of ANFIS with Different MFs

In this section, we discuss the effect of different MFs of input for filtering the image contaminated with low, medium and high noise. The names of these MFs are bell MF, Gaussian MF, two-sided Gaussian MF, triangle MF, trapezoidal MF, product of two sigmoid MFs, difference between two sigmoidal MFs, pi-shaped MF.

Restoration with the Bell Membership Function

First we choose the bell membership function as MFs of this ANFIS. There are two inputs, $v_1(i-1)$ and $v_1(i)$, for the nonlinear passage dynamics of order 2 and two MFs for each input. The training step size is assumed to be 0.1.

The comparison of RMSEs (root mean square error) and the training step sizes with low, medium and high noise is illustrated in Fig. 4.6.

We show the changes of bell MFs before and after training with low, medium and high Gaussian noise in Fig 4.7, 4.8 and 4.9. Because the differences of the MFs in the

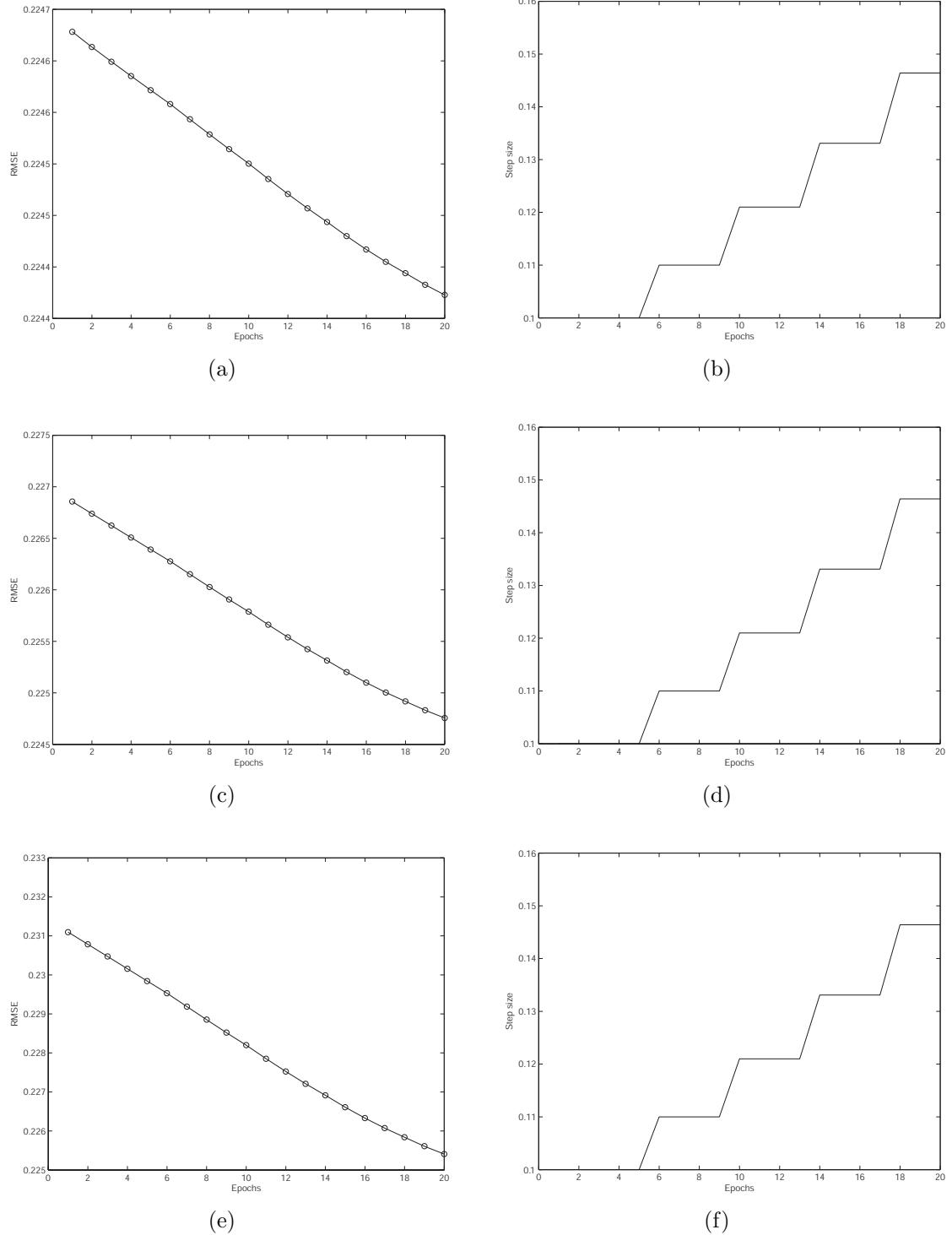


Fig. 4.6: RMSE curves and training step size of bell MFs: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.

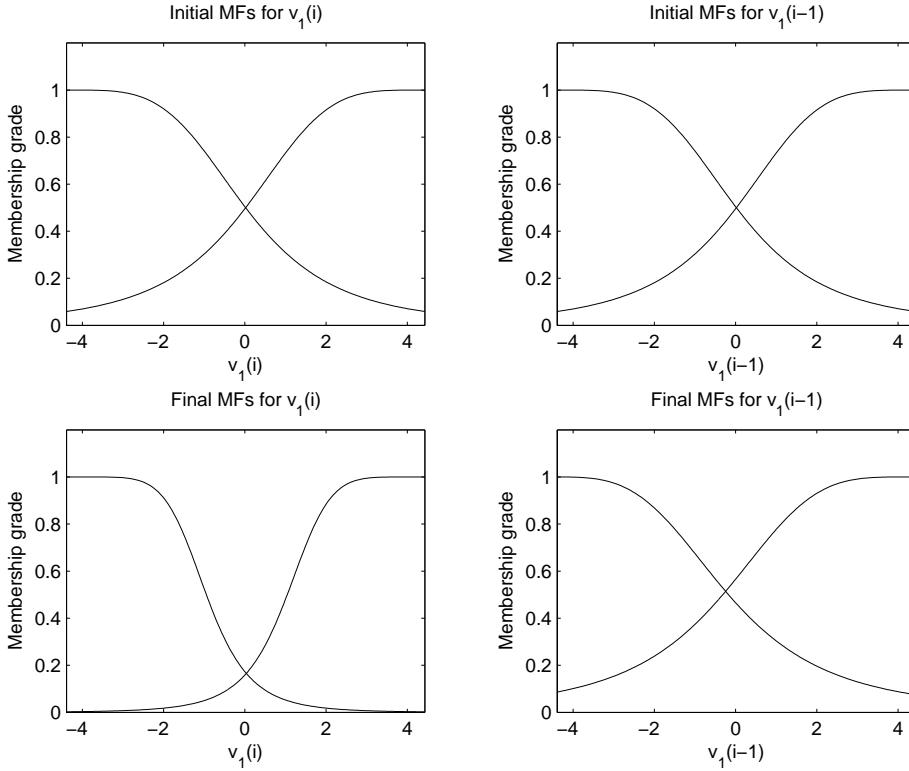


Fig. 4.7: The changes in bell MFs before and after training with low Gaussian noise.

restoration processing are small for low, medium and high noise, we only show the changes of MFs for high noise before and after training in the following discussion with different types of MFs.

The images corrupted by the intensity of low, medium and high Gaussian noise and the results of removing noise with ANFIS are shown in Fig 4.10(a) through 4.10(f) respectively. The restoration effect is good compared with the original image in Fig. 4.1(b) no matter how corrupted the image is by heavy noise.

Restoration with the Triangle Membership Function

We choose another MF, the triangle membership function to remove the noise and restore the image. We first show the MFs to high noise before and after training. As previously stated, the differences of the MFs are small in restoring the image corrupted by low, medium and high noise, therefore we just show the changes of triangle MFs with high noise before and after training in Fig 4.11.

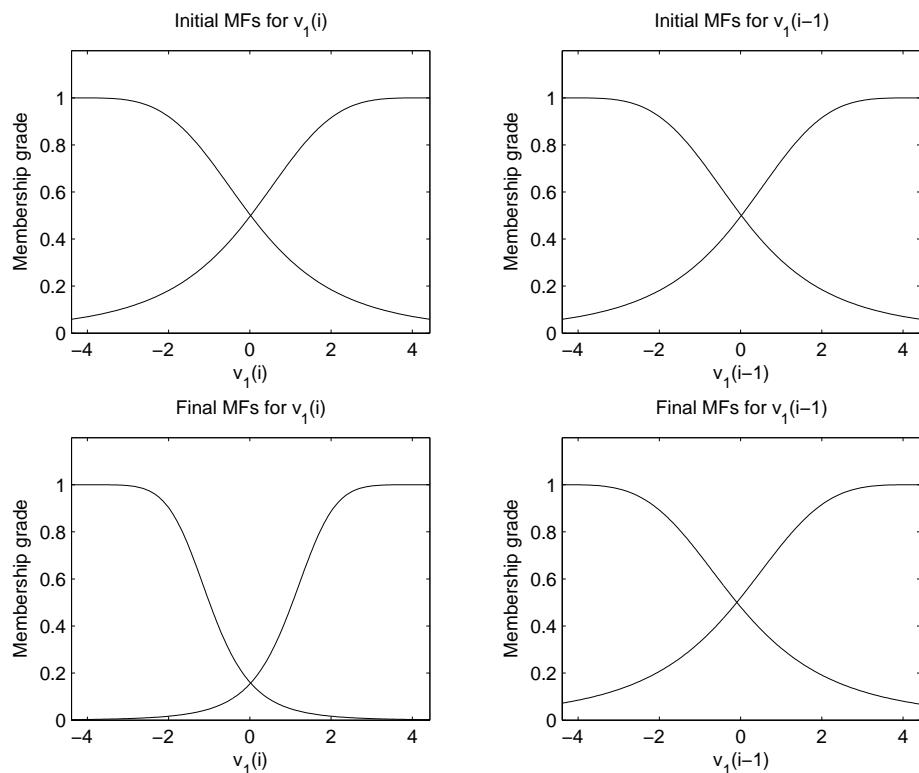


Fig. 4.8: *The changes of bell MFs before and after training with medium Gaussian noise.*

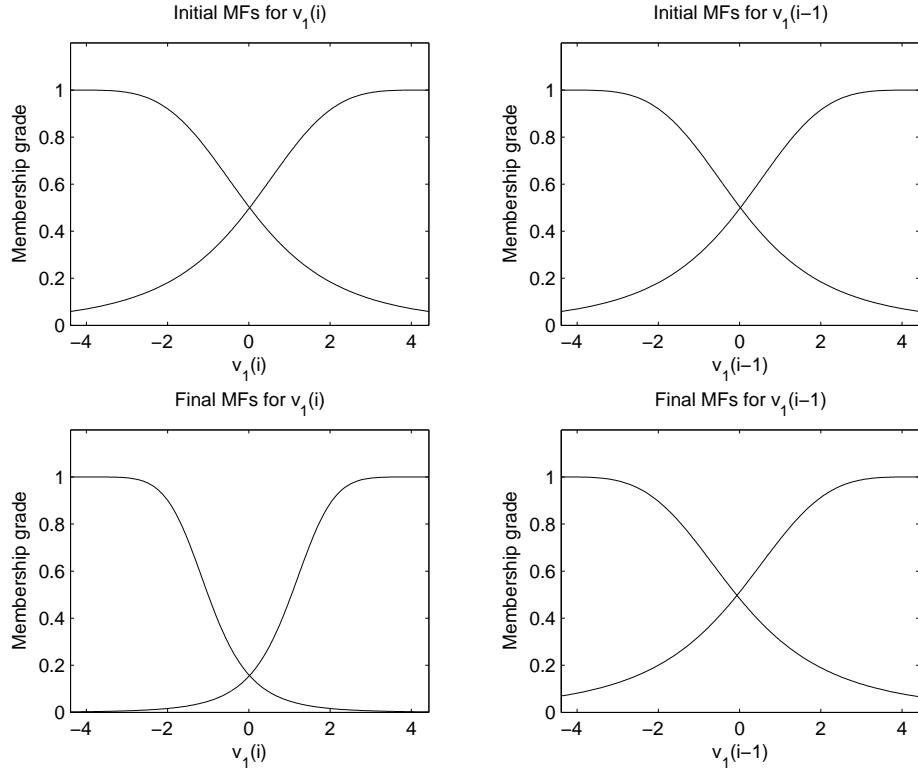


Fig. 4.9: The changes of bell MFs before and after training with high Gaussian noise.

The SNRs are still 12.7131, 3.1707 and -1.2663 for the intensity of low, medium and high Gaussian noises respectively. We show the comparison of RMSEs and the training step sizes with low, medium and high noise levels in Fig. 4.12.

Finally, we display the images corrupted by the intensity of low, medium and high noises and the results of removing noises with ANFIS shown in Fig 4.13(a) to 4.13(f), respectively.

Restoration with the Gaussian Membership Function

Now we use the Gaussian membership function to remove the noise and compare the results of the image restoration. The MFs to high noise before and after training for high noise are shown in Fig 4.14.

We use the same SNRs for the intensity of low, medium and high noises. We display the comparison of RMSEs and the training step sizes with low, medium and high noise levels in Fig. 4.15.

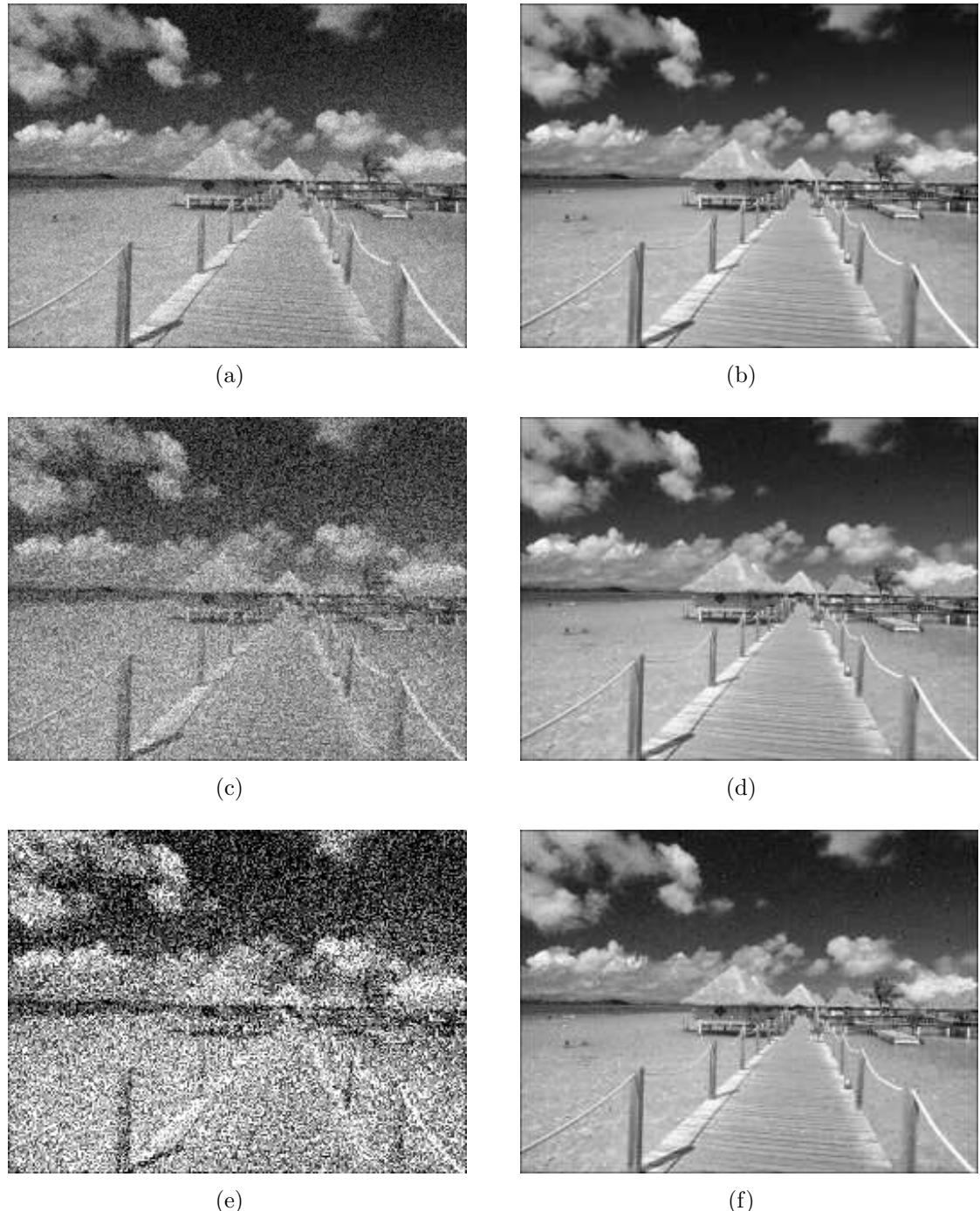


Fig. 4.10: The image corrupted by different intensity noises and the results of removing noises with ANFIS of bell MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

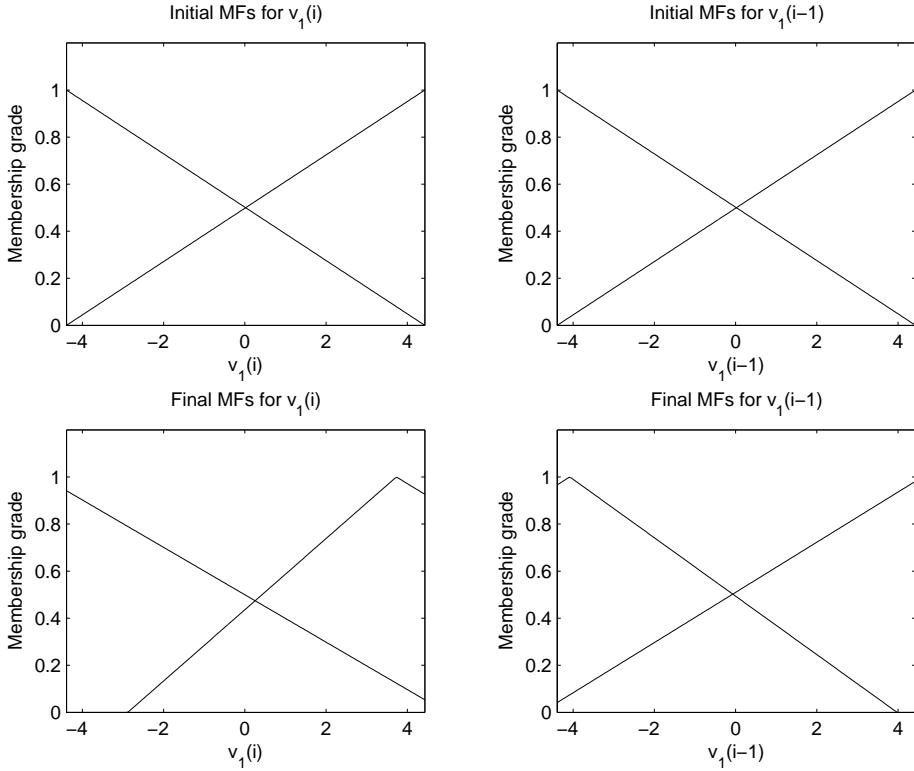


Fig. 4.11: *The changes in triangle MFs before and after training.*

Finally we display the images corrupted by the different intensity noises and the restored results of the images corrupted by low, medium and high noise levels with ANFIS in Fig 4.16.

Restoration with the Two-sided Gaussian Membership Function

The two-sided Gaussian membership function is used to remove the noise and the results are compared with other MFs for cancellation of the noise. Fig 4.17 shows the two-sided Gaussian MFs to high noise level before and after training.

The same SNRs are used for the intensity of low, medium and high noises. We show the comparison of RMSEs and the training step sizes with low, medium and high noise levels in Fig. 4.18.

Finally we display the images corrupted by the different intensity noise and the restored results of the images corrupted by low, medium and high noise levels with ANFIS in Fig 4.19.

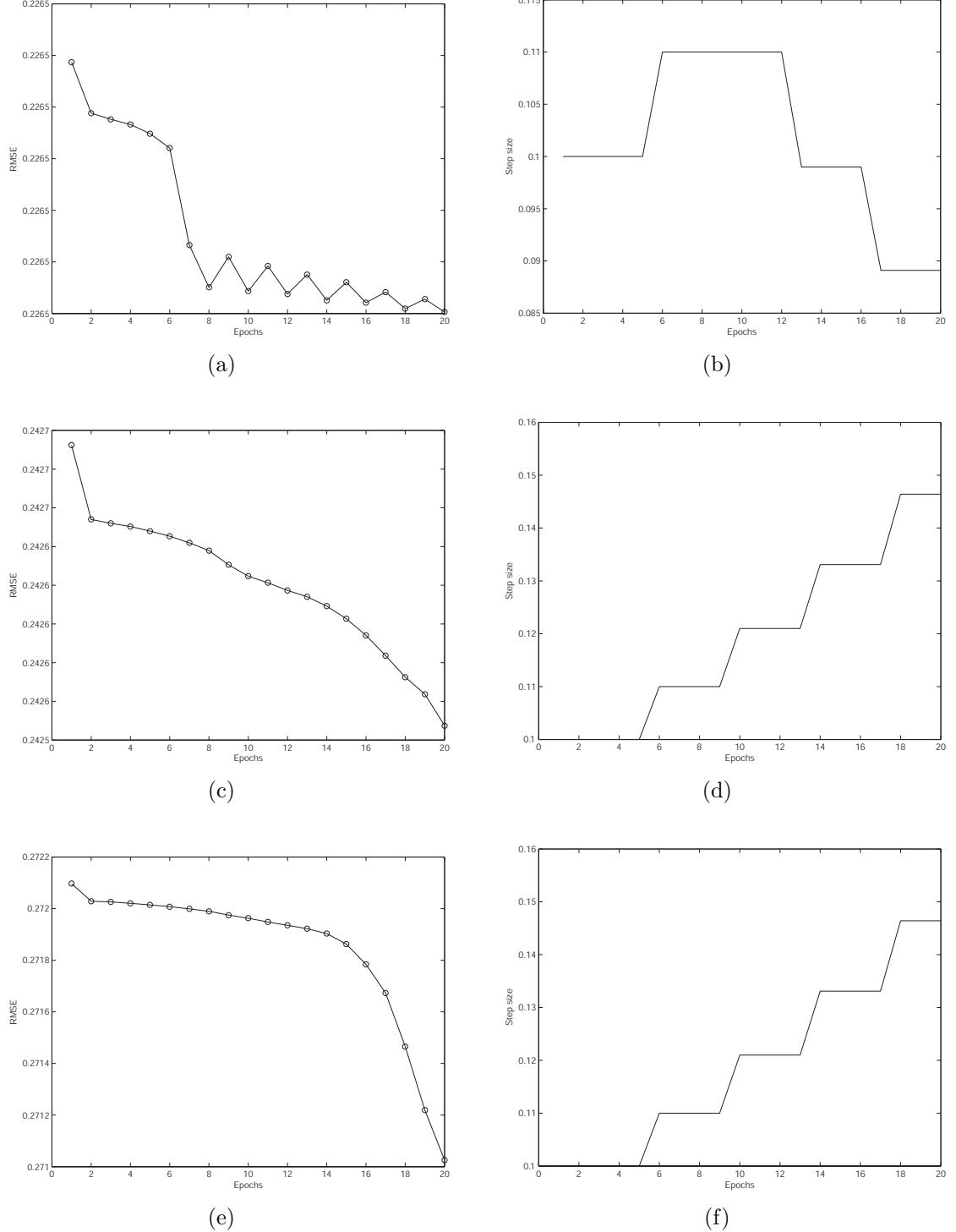


Fig. 4.12: RMSE curves and training step size of triangle MFs: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.

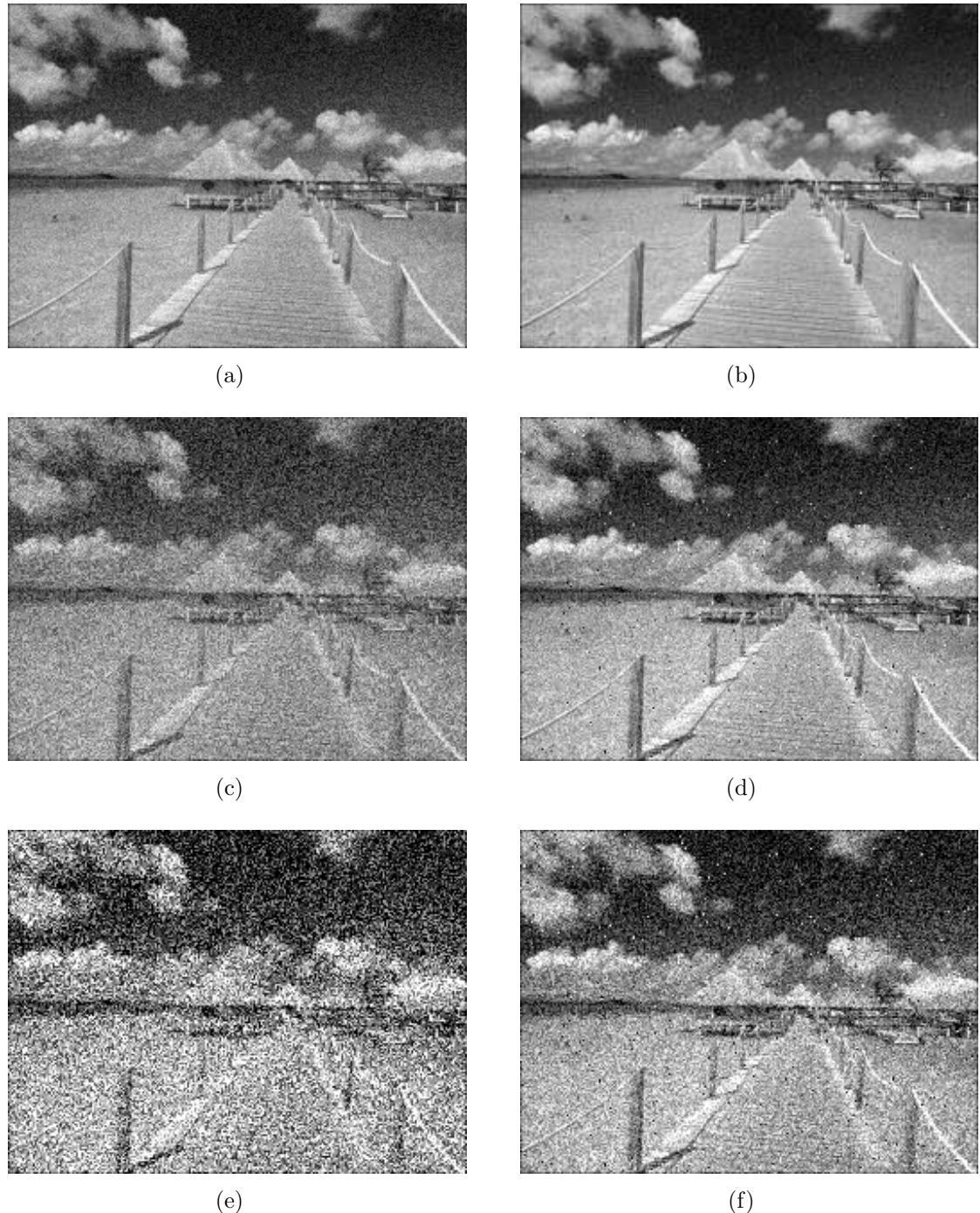


Fig. 4.13: The image corrupted by different intensity noises and the results of removing noises with ANFIS of triangle MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

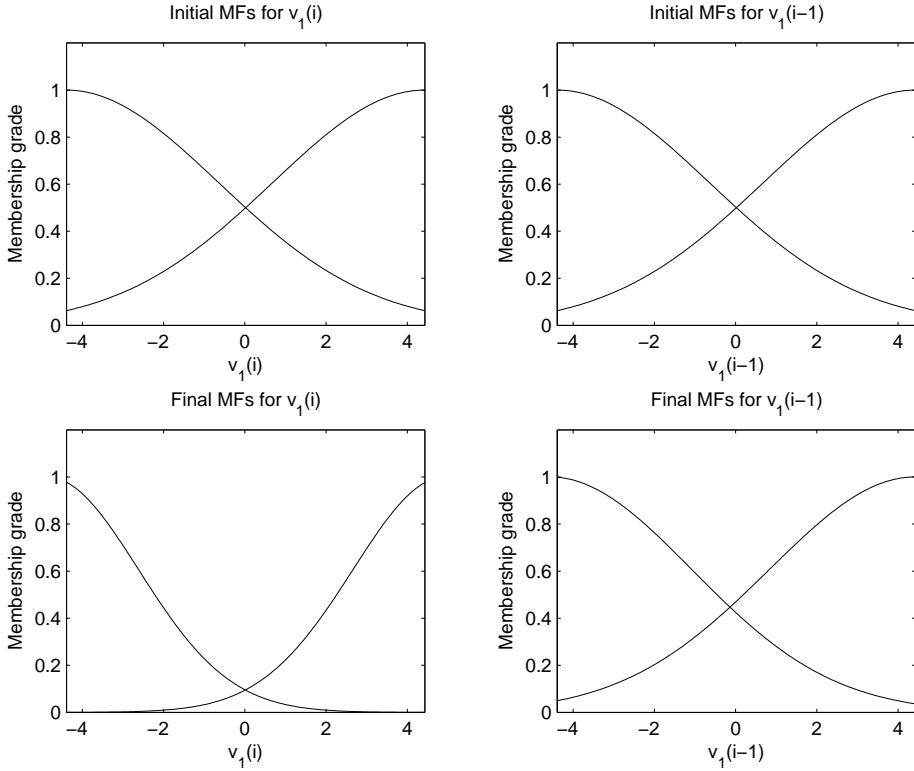


Fig. 4.14: *The changes in Gaussian MFs before and after training.*

Restoration with the Product of Two Sigmoid Membership Functions

The product of two sigmoid membership functions is introduced to restore the image corrupted by different intensity noise and the results are compared with other MFs for removing the noise. Fig 4.20 shows the product of two sigmoid MFs to high noise before and after training. The same SNRs are used for the intensities of low, medium and high noise levels.

We use the same SNRs for the intensities of low, medium and high noises and show the comparison of RMSEs and the training step sizes with low, medium and high noise levels in Fig. 4.21.

The images corrupted by the intensities of low, medium and high Gaussian noises and the results of removing noise with ANFIS are shown in Fig 4.22(a) through 4.22(f) respectively.

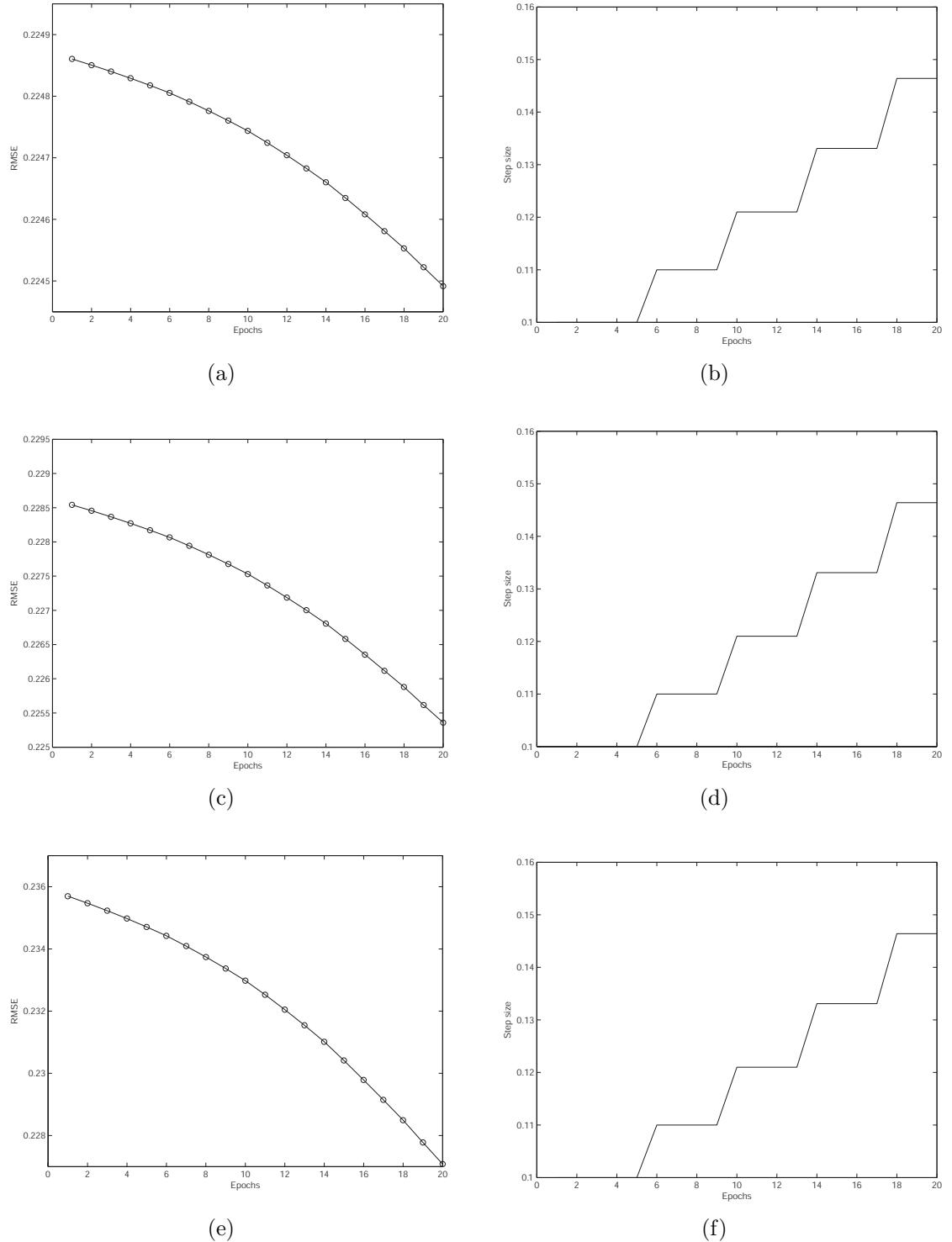


Fig. 4.15: RMSE curves and training step size of Gaussian MFs: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.

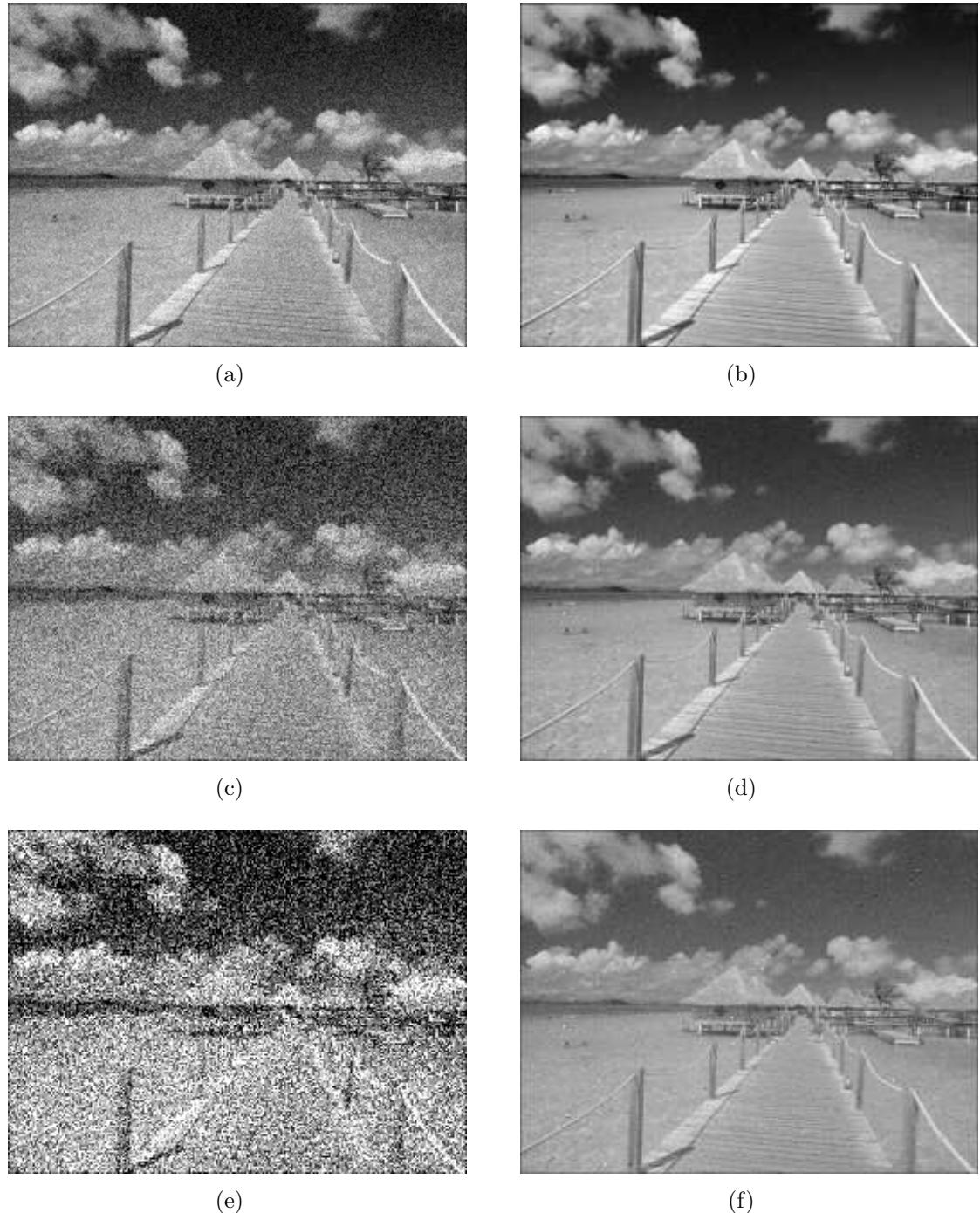


Fig. 4.16: The image corrupted by different intensity noises and the results of removing noises with ANFIS of Gaussian MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

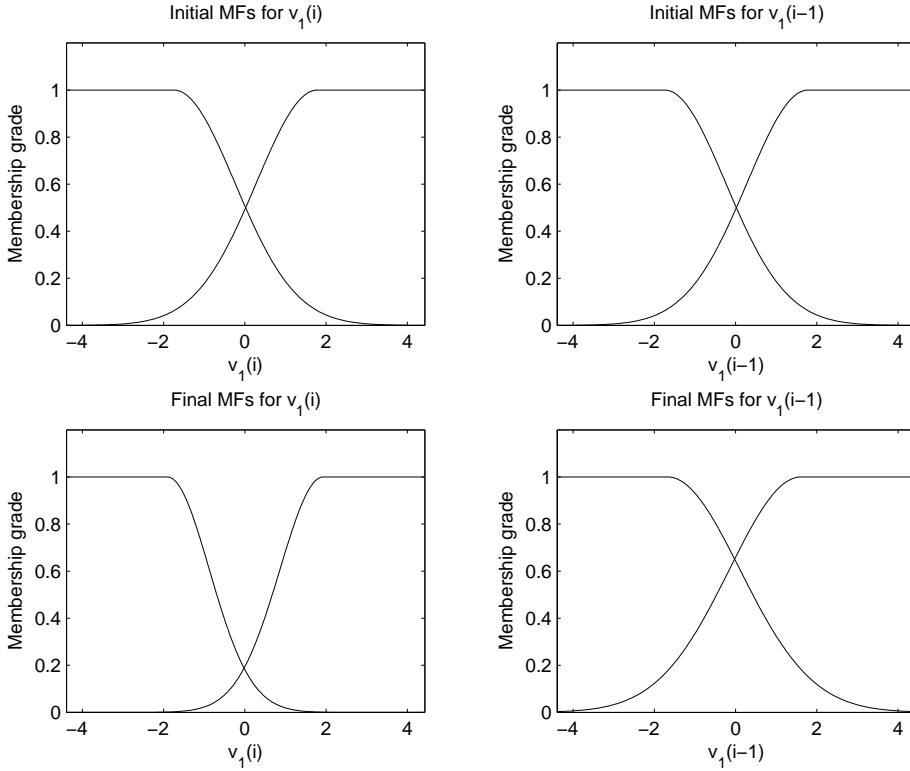


Fig. 4.17: *The changes in the two-sided Gaussian MFs before and after training.*

Restoration with the Trapezoidal Membership Function

The trapezoidal membership function is tested to restore the image corrupted by different intensity noises and the results are compared with other MFs for removing the noise. Fig 4.23 shows trapezoidal MFs to high noise before and after training. The SNRs are the same for the intensities of low, medium and high noise levels.

Then we display the comparison of RMSEs and the training step sizes with low, medium and high noise levels in Fig. 4.24.

Finally we display the images corrupted by the different intensity noises and the restored results of the images corrupted by low, medium and high noise levels with ANFIS in Fig 4.25.

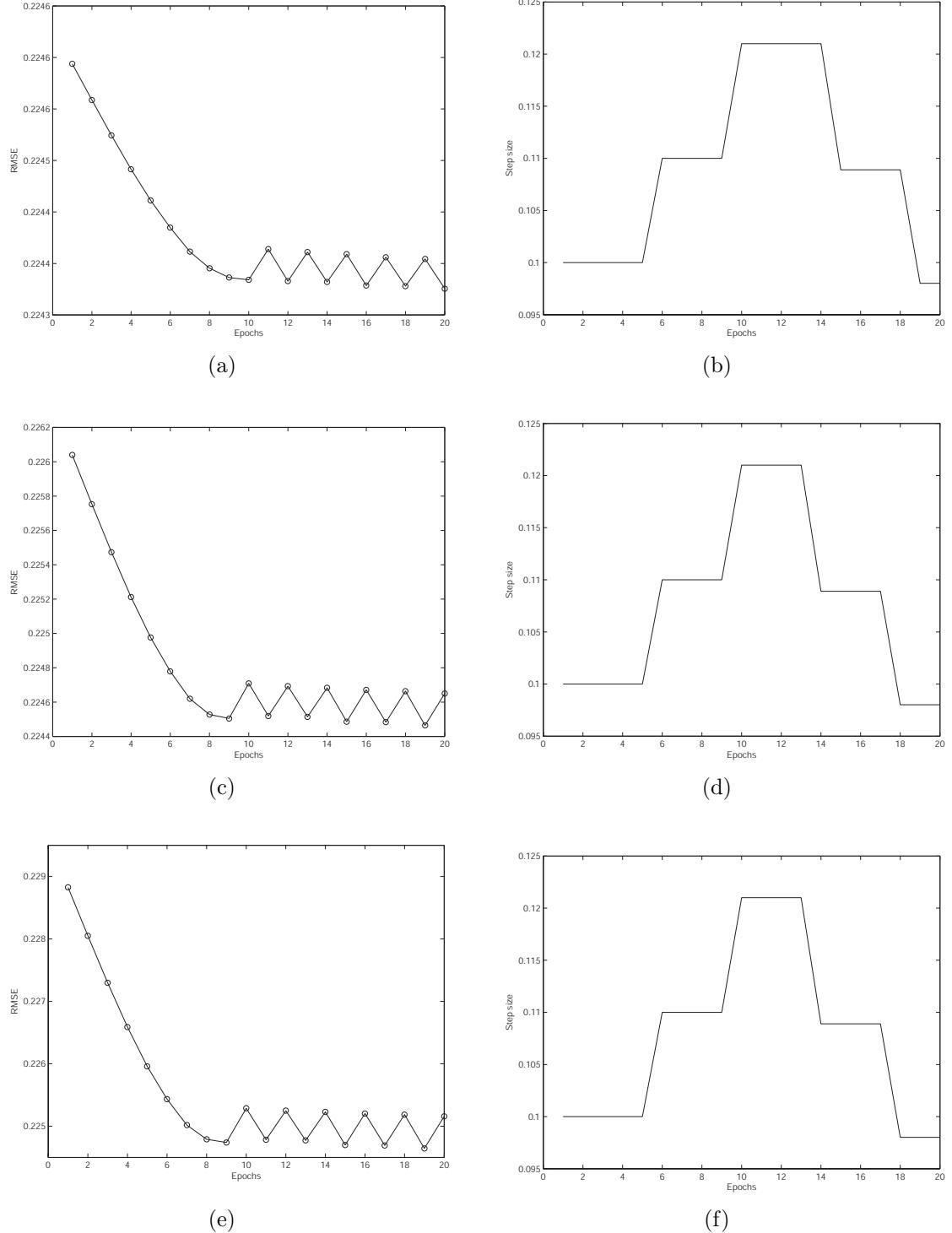


Fig. 4.18: RMSE curves and training step size of two-sided Gaussian MFs: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.

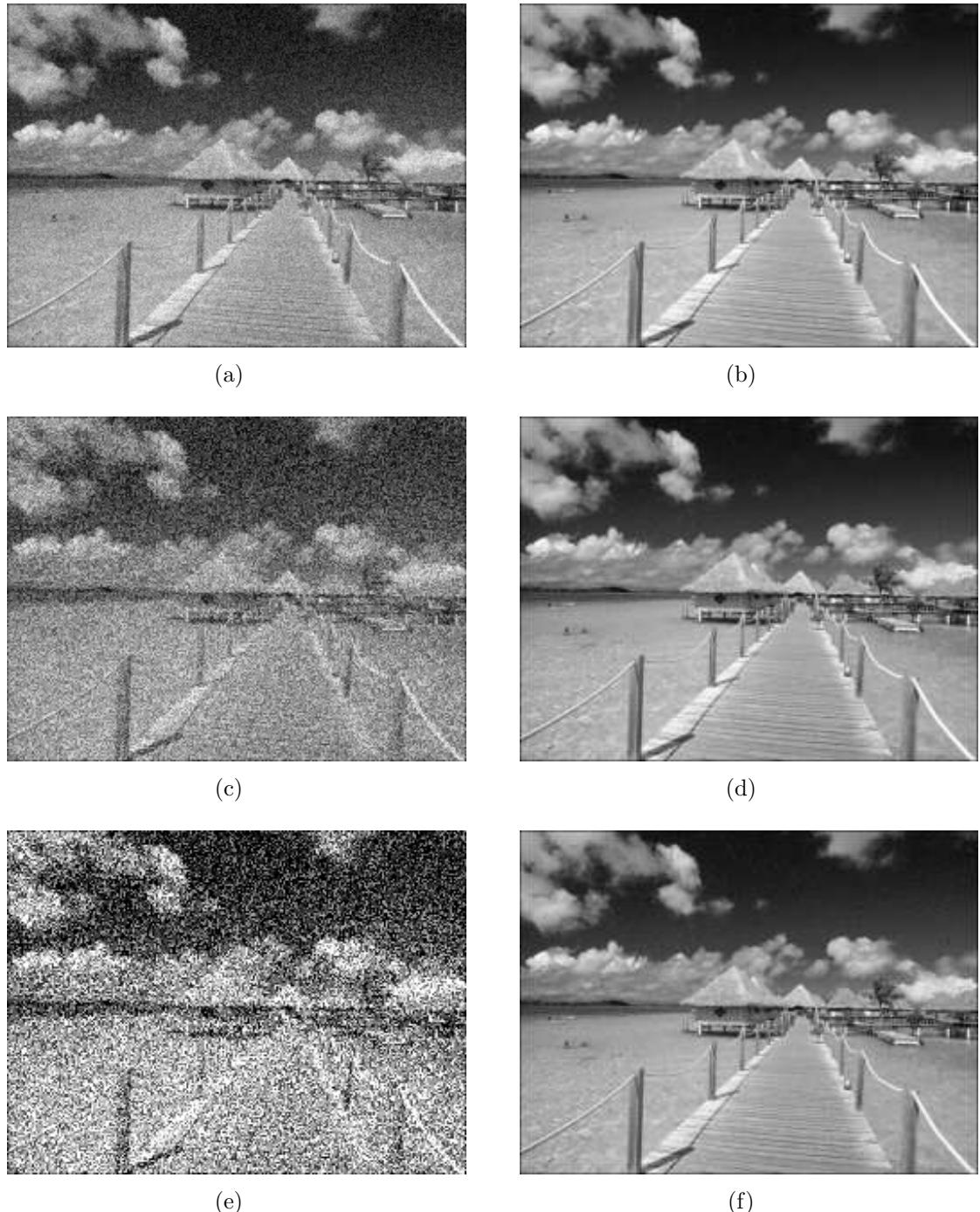


Fig. 4.19: *The image corrupted by different intensity noises and the results of removing noises with ANFIS of the two sided Gaussian MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.*

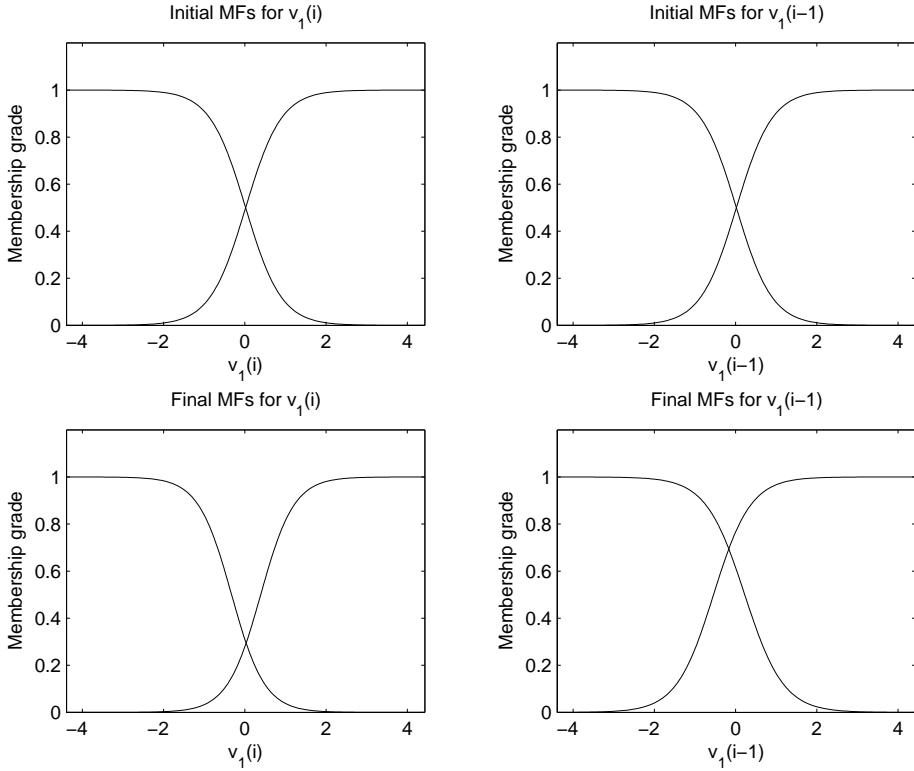


Fig. 4.20: *The changes in product of two sigmoid MFs before and after training.*

Restoration with the Difference between Two Sigmoidal Membership Functions

We use the difference between two sigmoidal membership function to restore the image corrupted by different intensity noises and compare the results with other MFs for removing the noises. Fig 4.26 shows the difference between two sigmoidal MFs to high noise before and after training. The SNRs are the same for the intensities of low, medium and high noise levels.

We display the comparison of RMSEs and the training step sizes with low, medium and high noise levels in Fig. 4.27. We finally show the images corrupted by the different intensity noises and the restored results of the images corrupted by low, medium and high noise levels with ANFIS in Fig 4.28.

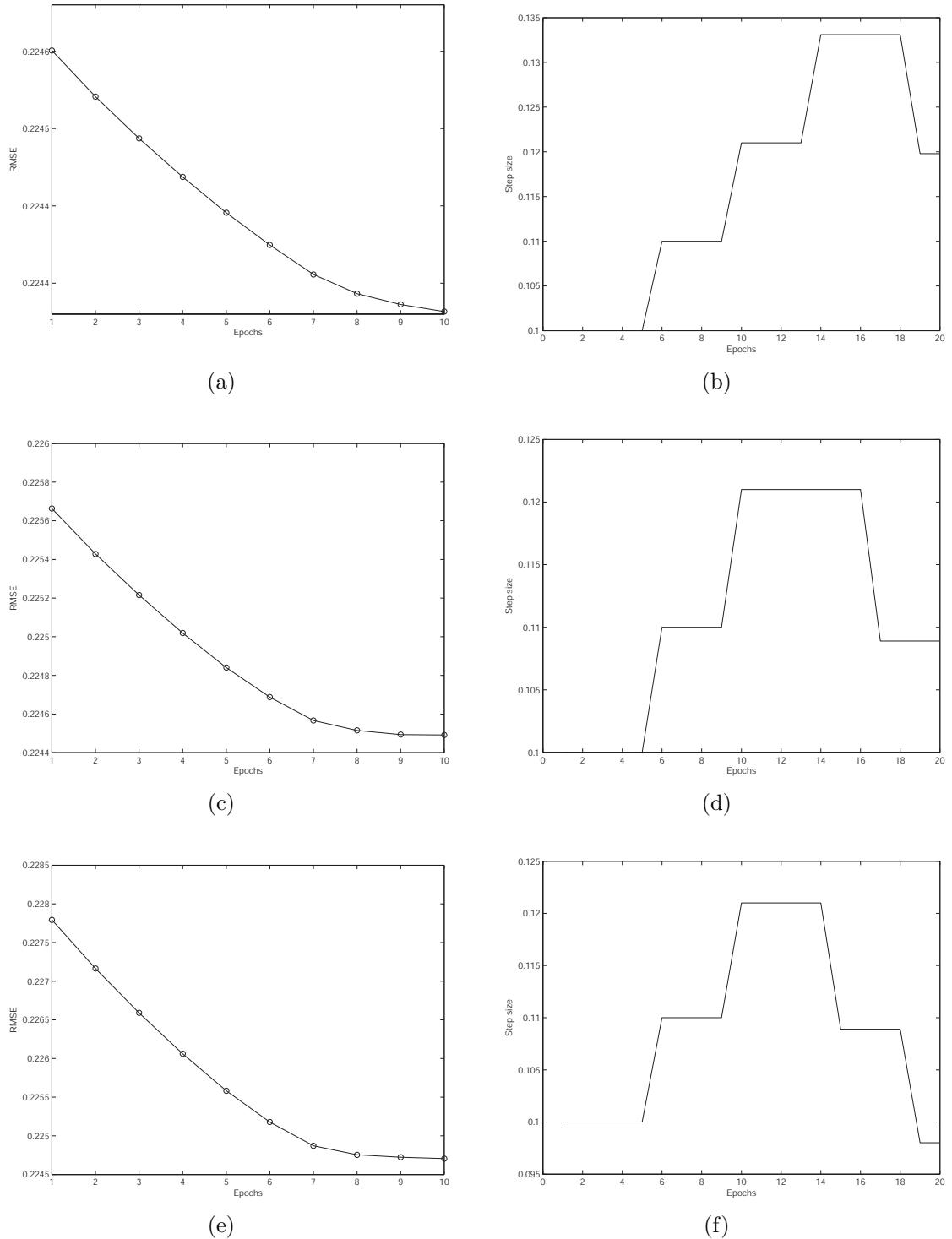


Fig. 4.21: RMSE curves and training step size of product of two sigmoid MFs: (a), (c) and (e) RMSE curves to low, medium and high noises; (b), (d) and (f) The training step size for low, medium and high noises.

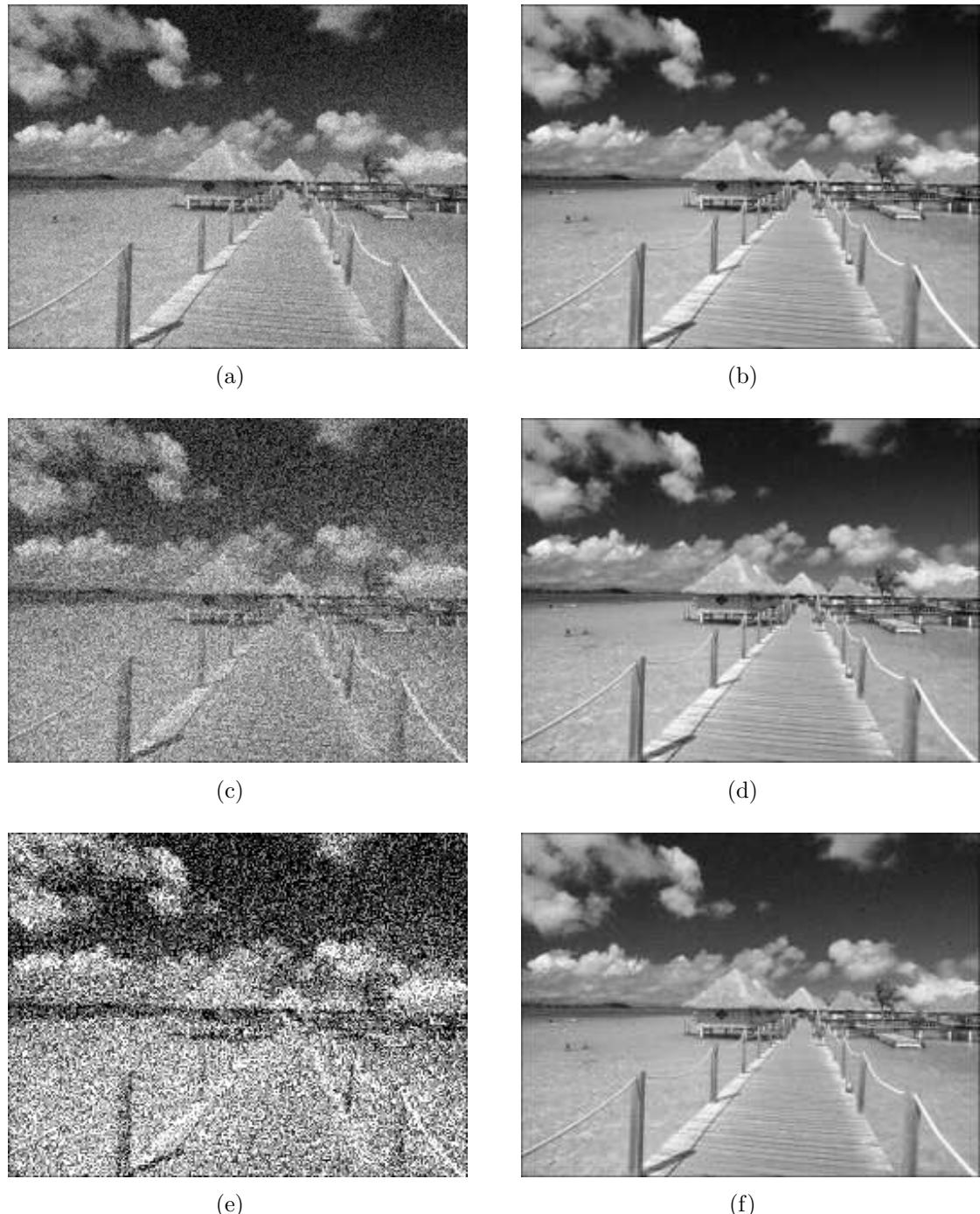


Fig. 4.22: The image corrupted by different intensity noises and the results of removing noises with ANFIS of the product of two sigmoid membership functions: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

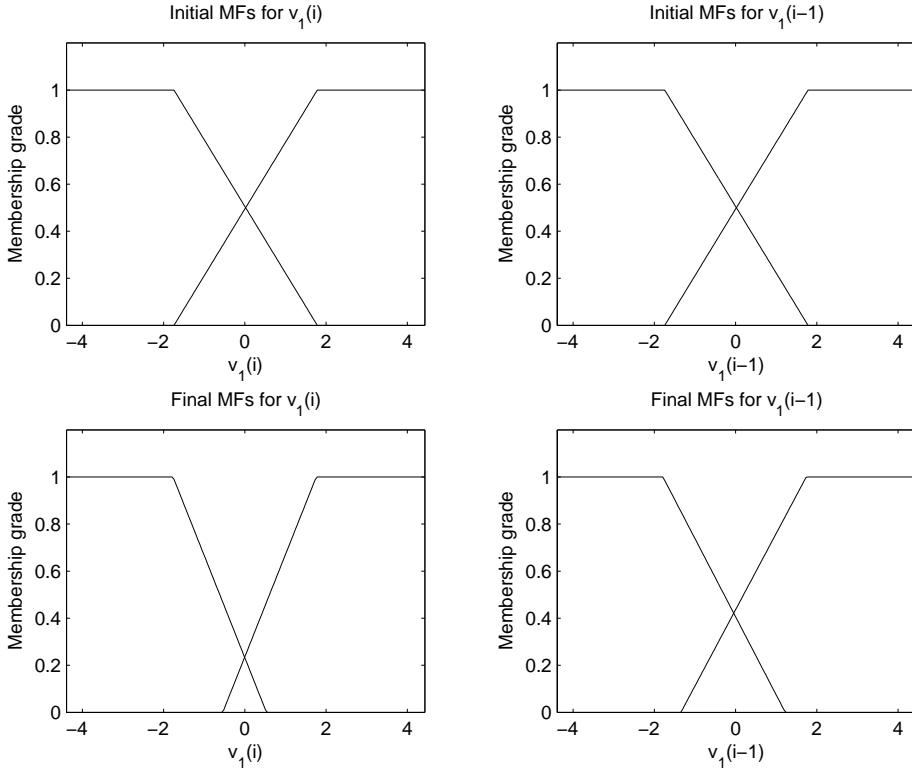


Fig. 4.23: *The changes in the trapezoidal MFs before and after training.*

Restoration with the Pi-shaped Membership Functions

We try the pi-shaped membership function to remove the different intensities of noises and compare the results with other MFs for cancelling the noises. Fig 4.29 shows the pi-shaped MFs to high noise before and after training. The SNRs are the same for the intensity of low, medium and high noise levels.

We display the comparison of RMSEs and the training step sizes with low, medium and high noise levels in Fig. 4.30. We finally show the images corrupted by the different intensity noises and the restored results of the images corrupted by low, medium and high noise levels with ANFIS in Fig 4.31.

Summary of All Membership Functions

All MSE results processed with different MFs are listed in descending order in Table 4.1 for comparison purposes. We now find the image restoration is the best with the pi-shaped MF. The difference between two sigmoidal MF, the product of two

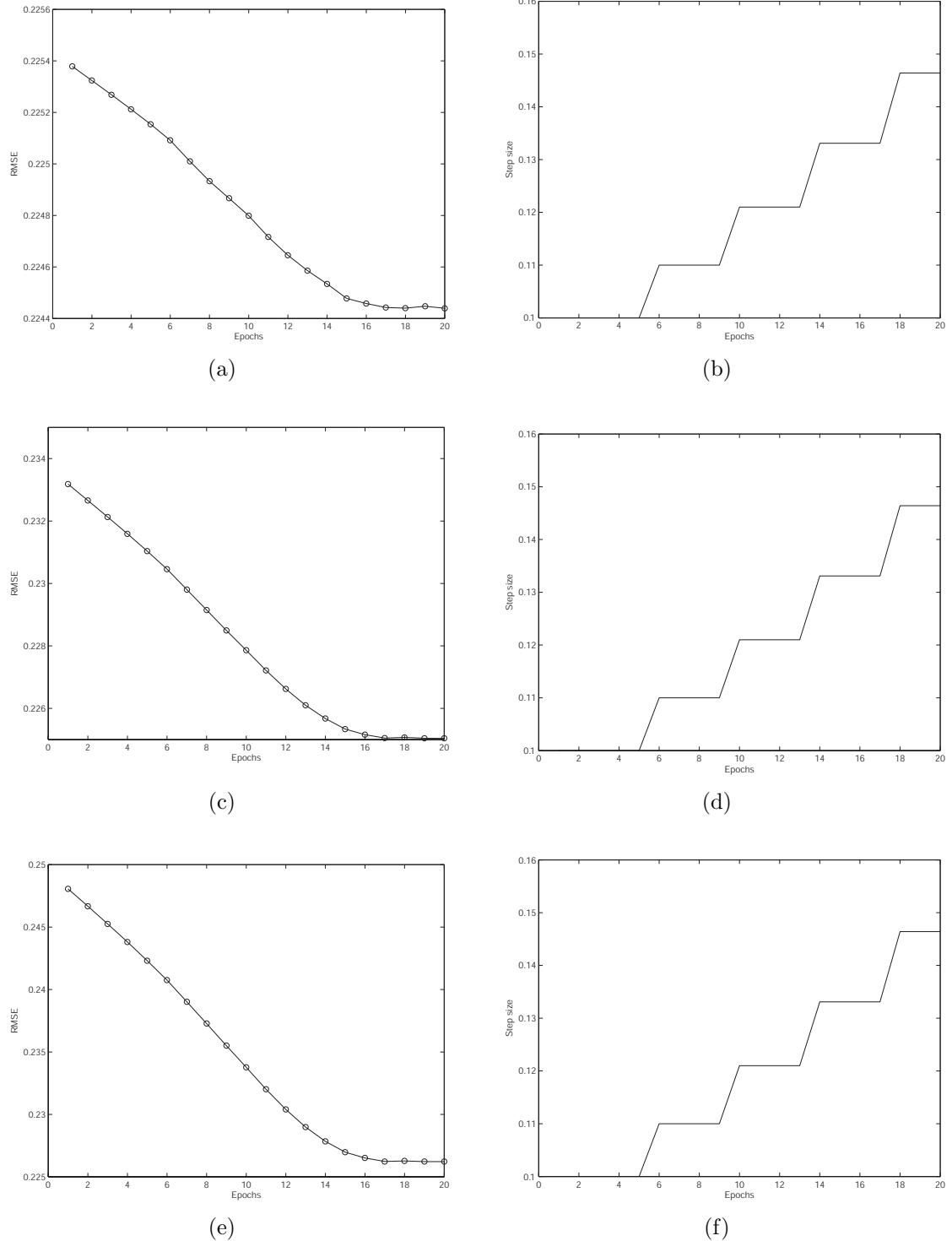


Fig. 4.24: RMSE curves and training step size of trapezoidal MFs: (a), (c) and (e) RMSE curves to low, medium and high noises; (b), (d) and (f) The training step size for low, medium and high noises.

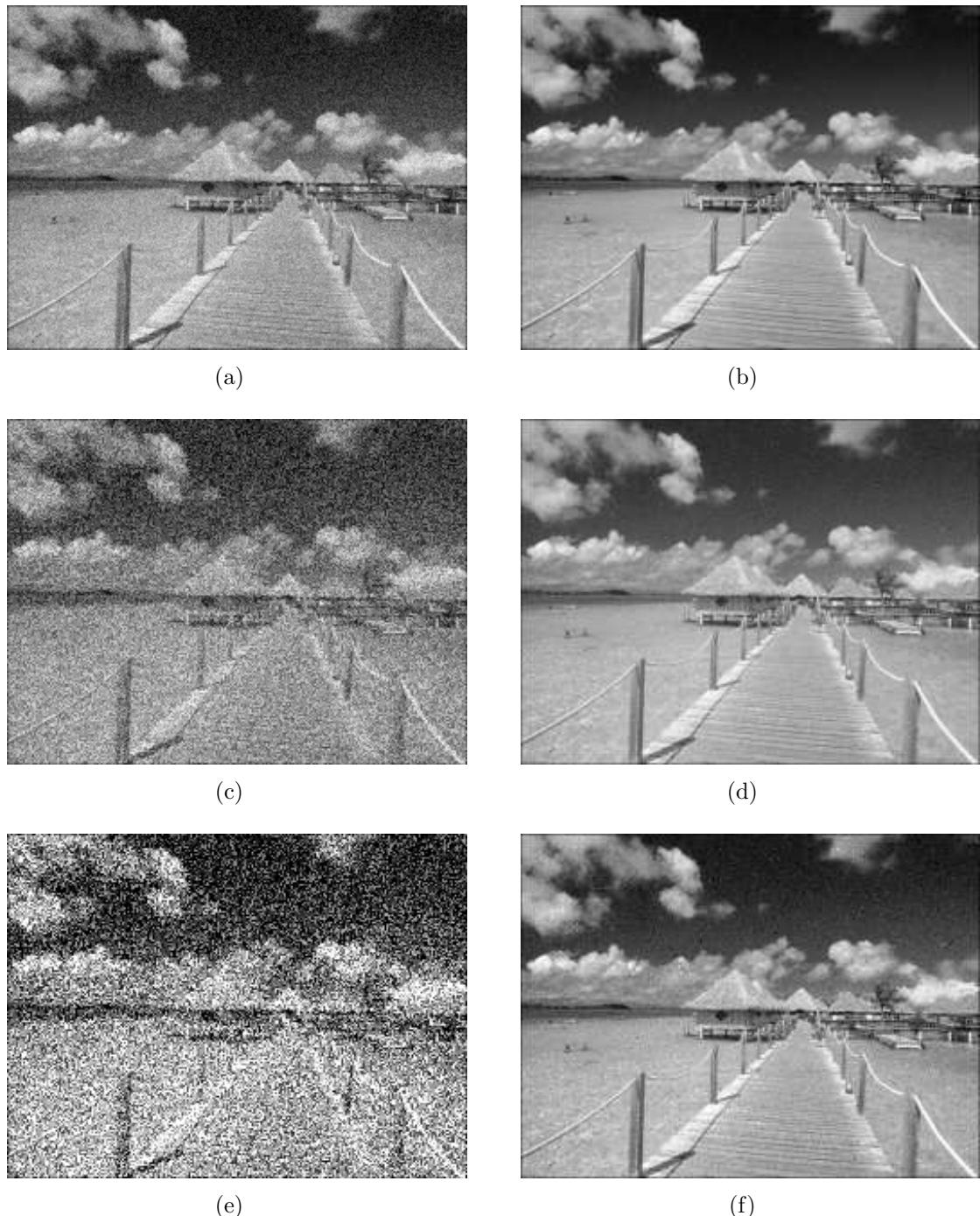


Fig. 4.25: *The image corrupted by different intensity noises and the results of removing noises with ANFIS of the trapezoidal MFs:* (a) *The image with low noise;* (b) *The restoration from low noise;* (c) *The image with medium noise;* (d) *The restoration from medium noise;* (e) *The image with high noise;* (f) *The restoration from high noise.*

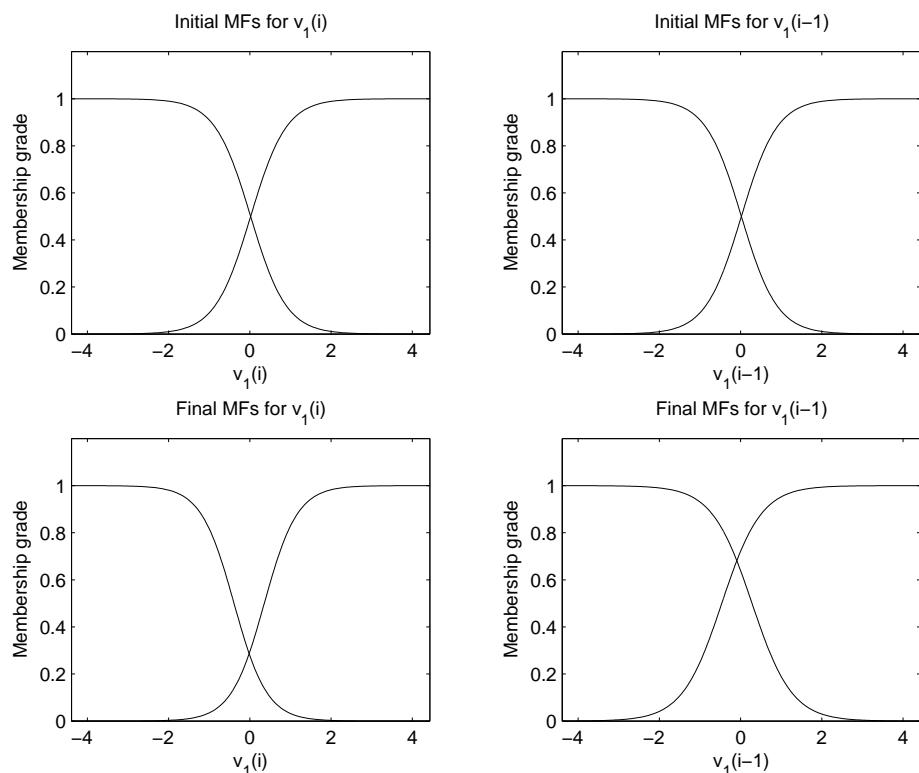


Fig. 4.26: *The changes in the difference between two sigmoidal MFs before and after training.*

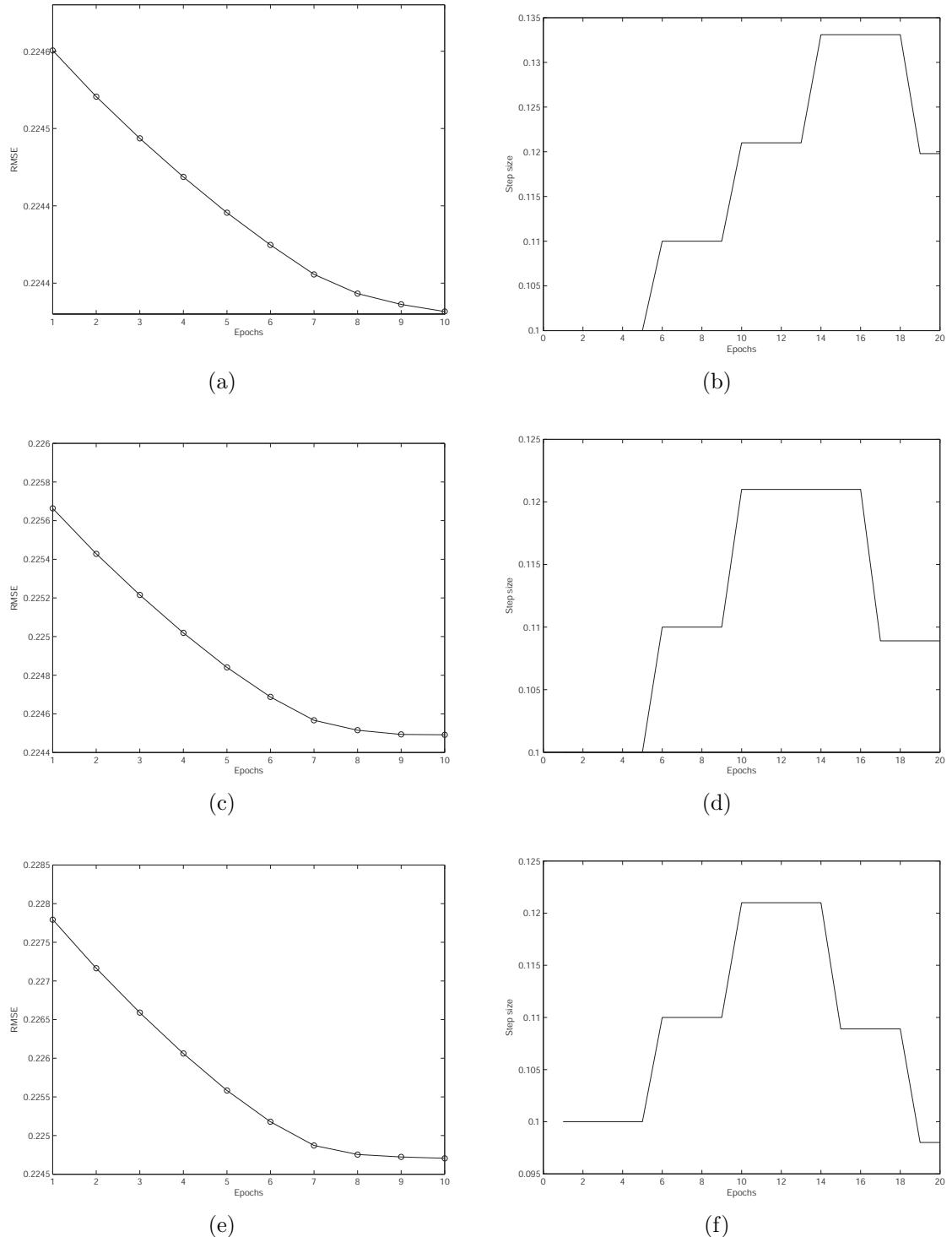


Fig. 4.27: RMSE curves and training step size of difference between two sigmoidal MFs: (a), (c) and (e) RMSE curves to low, medium and high noises; (b), (d) and (f) The training step size for low, medium and high noises.

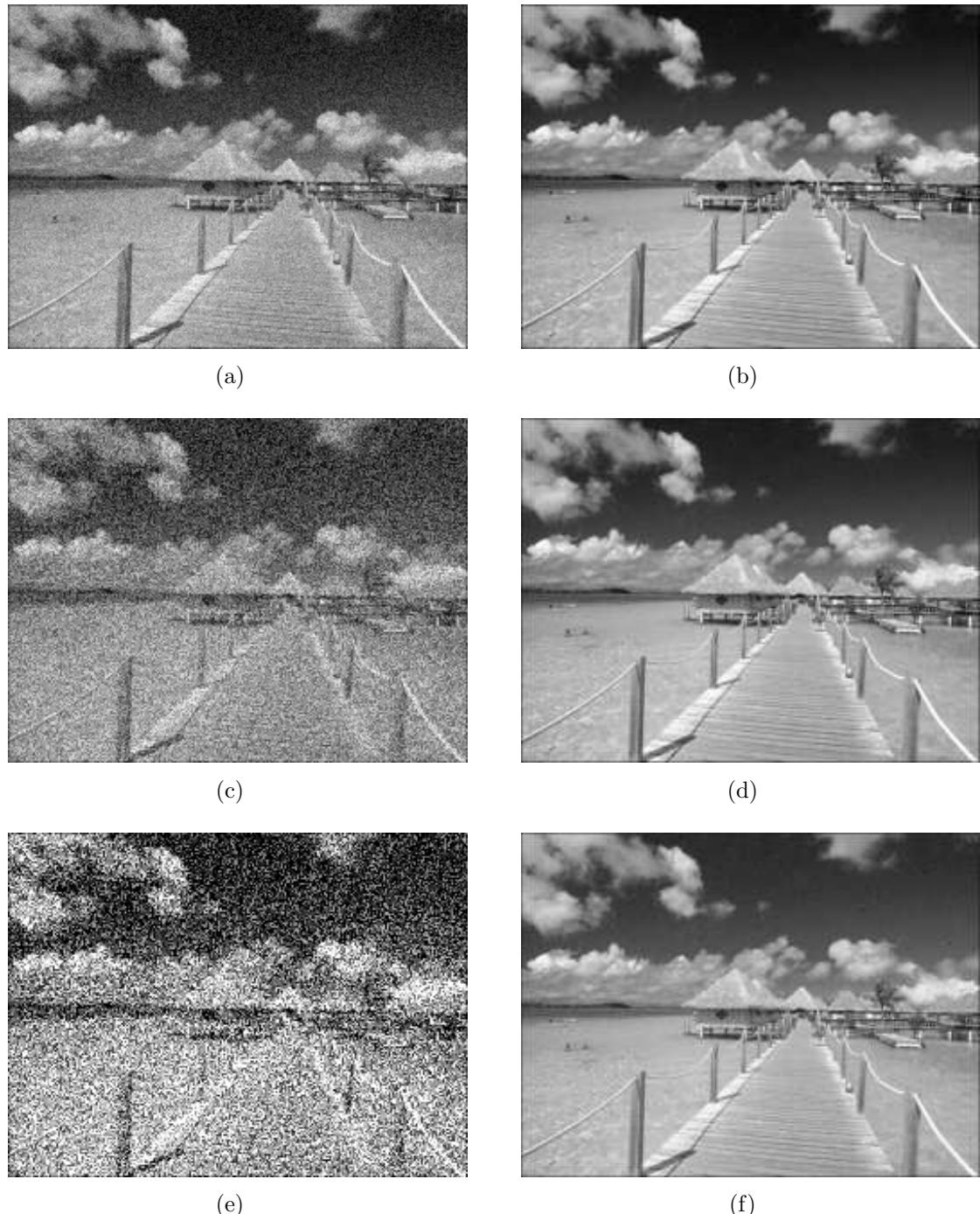


Fig. 4.28: The image corrupted by different intensity noises and the results of removing noises with ANFIS of the difference between two sigmoidal MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

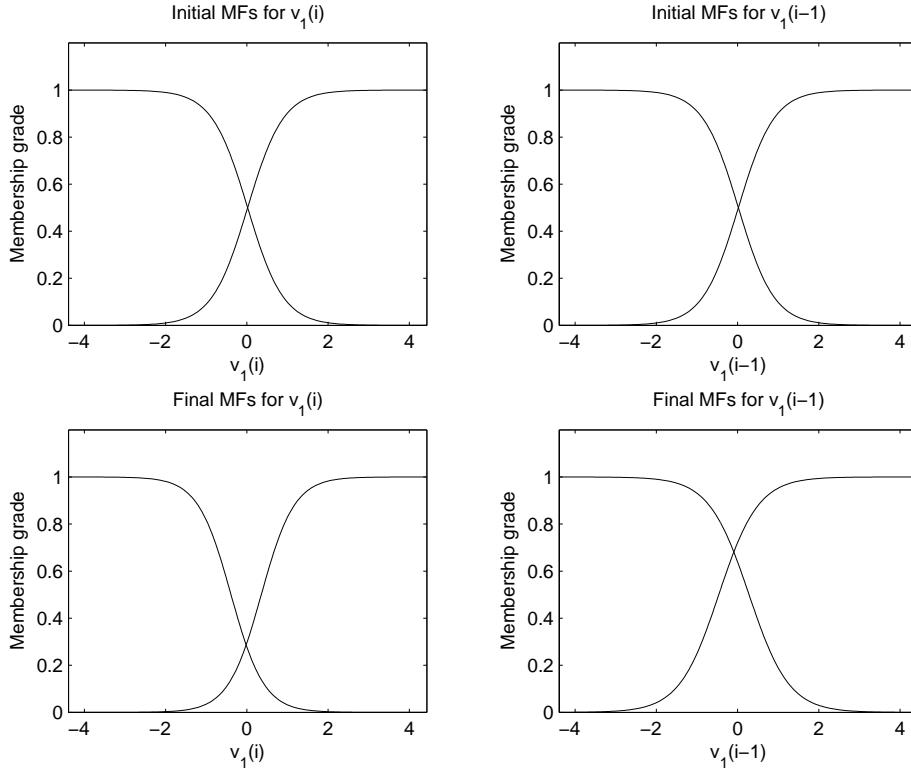


Fig. 4.29: The changes in the pi-shaped MFs before and after training.

sigmoid MF, and the two-sided gaussian MF are slightly worse than the pi-shaped MF. The Gaussian MF and the trapezoidal MF are poorer than the above mentioned MFs in removing noise from the corrupted images. The poorest MF is the triangle MF. These results can also be observed from the restoration of images which are contaminated with the same noise.

4.1.3 Discussion of Parameters of ANFIS

There are many parameters we can select to obtain better results in ANFIS. For the most common case, these parameters are the number and type of membership function for each input, the output membership function type (either ‘*linear*’ or ‘*constant*’), the training epoch number, the training error goal, the initial step size, the step size decrease rate, the step size increase rate. Also we can ensure that the checking (validation) data are used to detect overfitting of the training data set. The checking data has the same format as the training data. Overfitting can be detected when

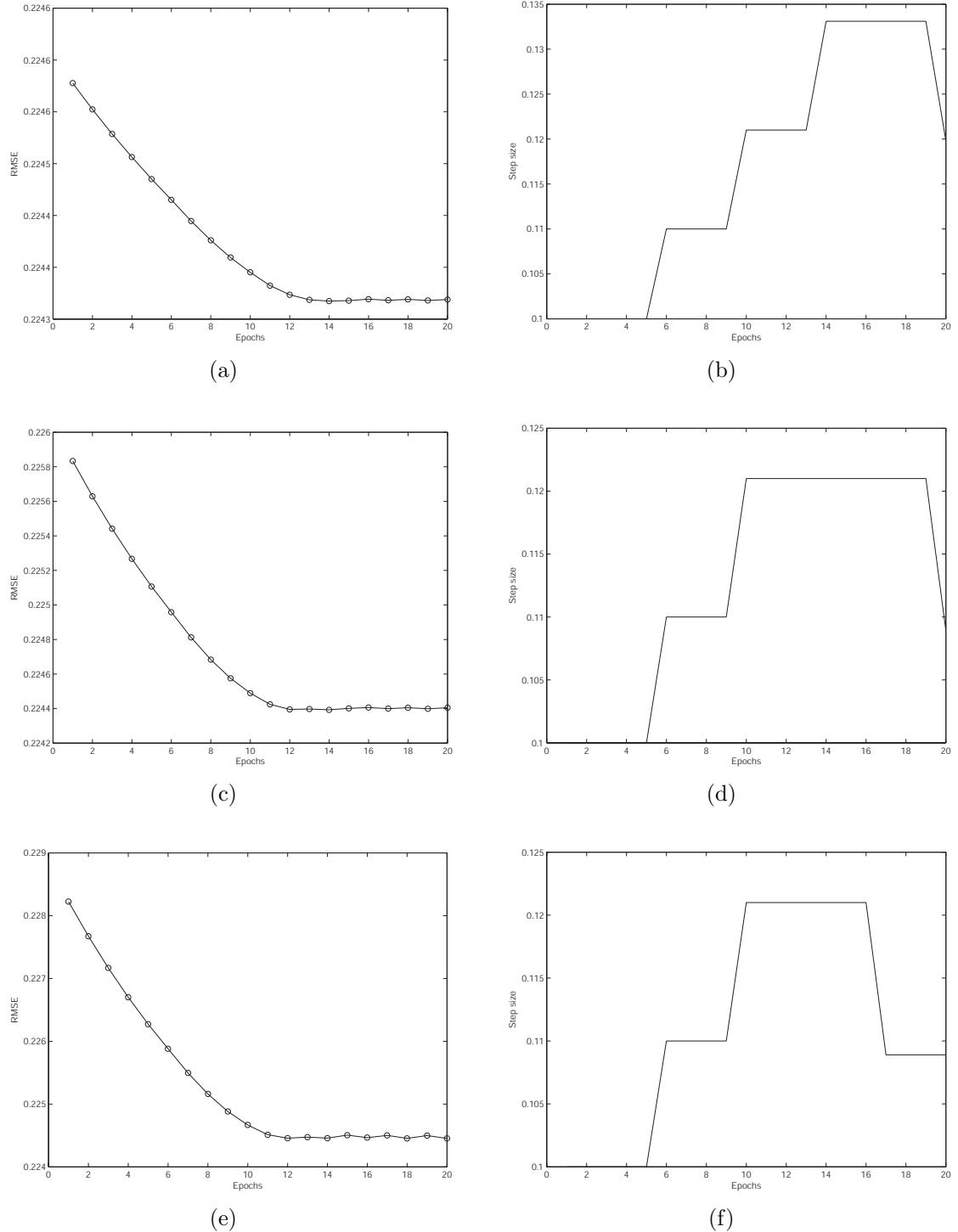


Fig. 4.30: *RMSE curves and training step size of pi-shaped MFs: (a), (c) and (e) RMSE curves to low, medium and high noises; (b), (d) and (f) The training step size for low, medium and high noises.*

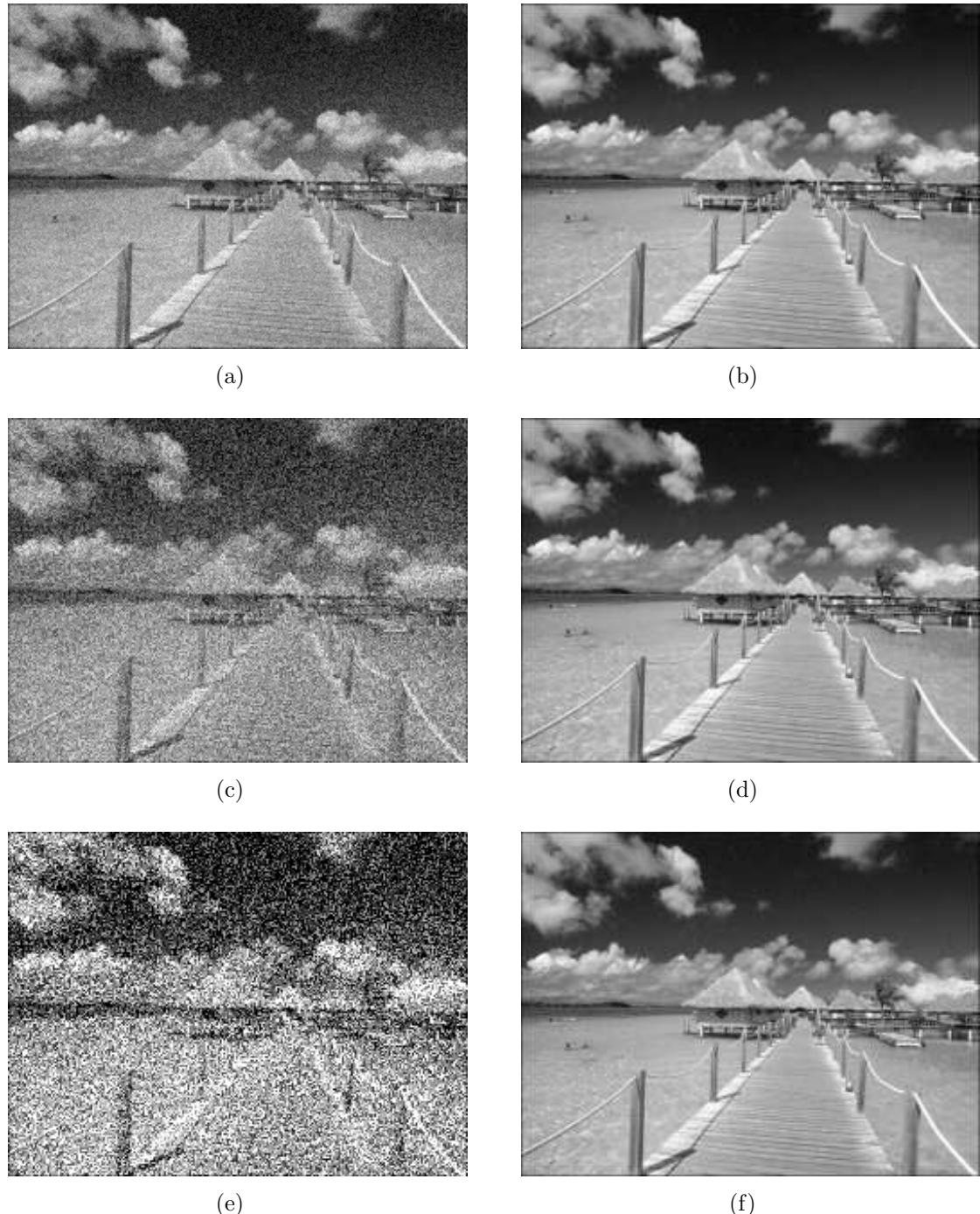


Fig. 4.31: *The image corrupted by different intensity noises and the filtered images with ANFIS of the pi-shaped MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.*

Table 4.1: *MSE between the original image and the restoration image with all types of MFs.*

Name of MFs	Intensity of noise (MSE)		
	Low	Medium	High
	175.3222	1577.9	4383.1
Triangle	61.8565	548.5101	1500.9
Gaussian	3.9071	29.1639	79.6379
Trapezoidal	4.7063	21.7074	56.3219
Bell	2.0298	11.3250	29.9477
Product of two sigmoid	1.4214	3.8072	8.7097
Difference between Two Sigmoidal	1.4214	3.8072	8.7097
Two-sided gaussian	1.1739	3.5559	8.8405
Pi-shaped	1.1888	2.0820	4.0841

the checking error (difference between the output from the training data and the checking data) starts increasing while the training error is still decreasing. In the following discussion about the parameter, we demonstrate the representable case of the image corrupted by high Gaussian noise and simplify the discussion to give up the case of the image corrupted by medium and low Gaussian noise.

Training Epoch Number

We set the training epoch number 50, 100, 200 and 400 with the bell MFs separately. Fig. 4.32(a) to 4.32(d) shows the RMSE with the intensity of high noise because the results of the RMSE with high noise is typical of all other intensity noise. We can find RMSE descending before 22 epochs. After that, RMSE oscillates but generally the tendency is still slightly descending. After 100 epochs, the curve of RMSE tend to stabilize in value. Even though it still reduces noise a little better and better after 100 epoches, the effect of removing noise is not good. On the other hand, it consumes much more time. Thus it is enough of training within 100 epoches.

Next the step size curve is demonstrated in Fig. 4.33. We can observe that when the step size increases, RMSE descends consistently; when the step size decreases, RMSE oscillates.

With the increasing training epoch number, the quality of the restoration image

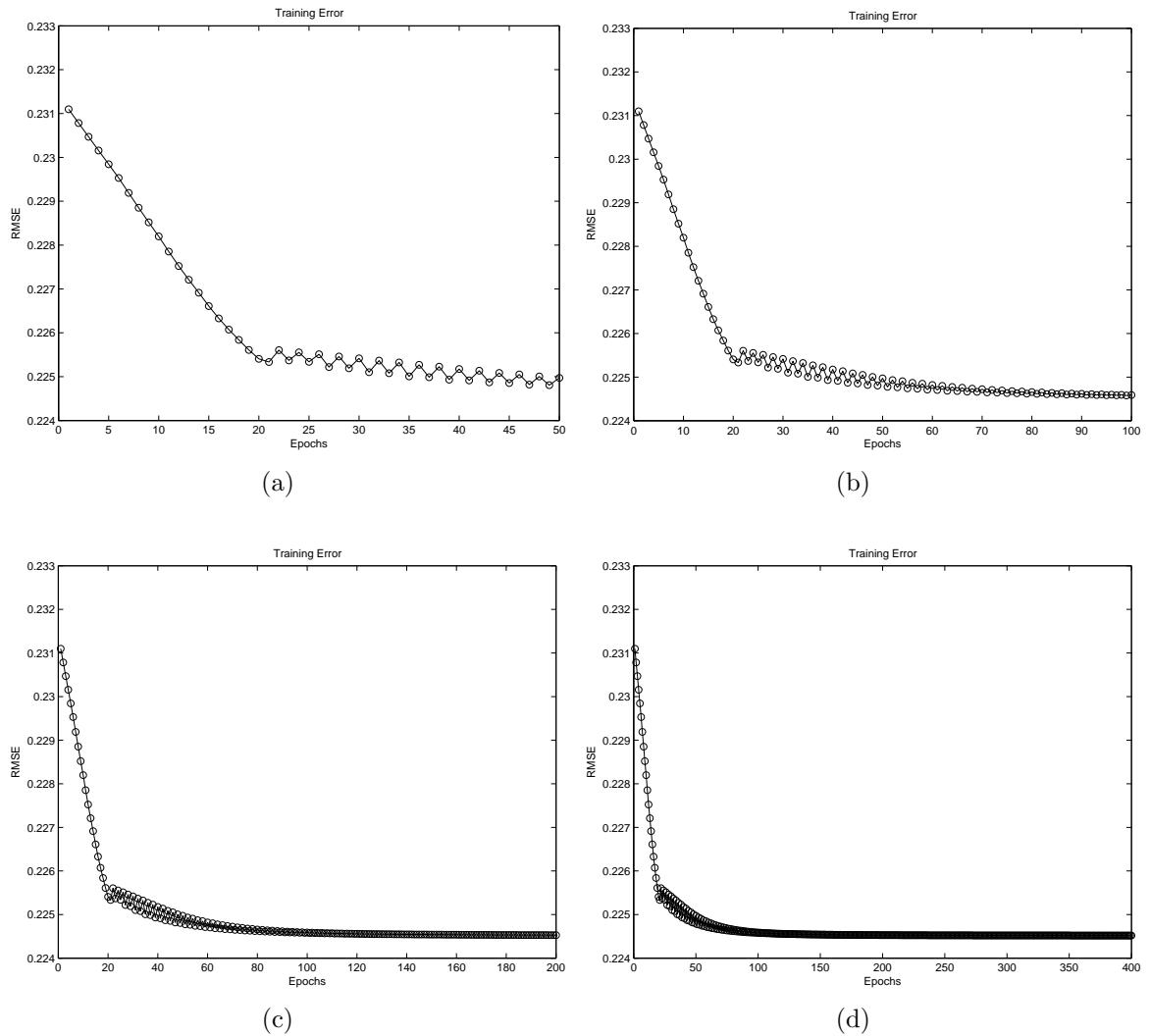


Fig. 4.32: The RMSE of ANFIS removing high Gaussian noise with the bell MFs for the different training epoch numbers: (a) 50 epochs; (b) 100 epochs; (c) 200 epochs; (d) 400 epochs.

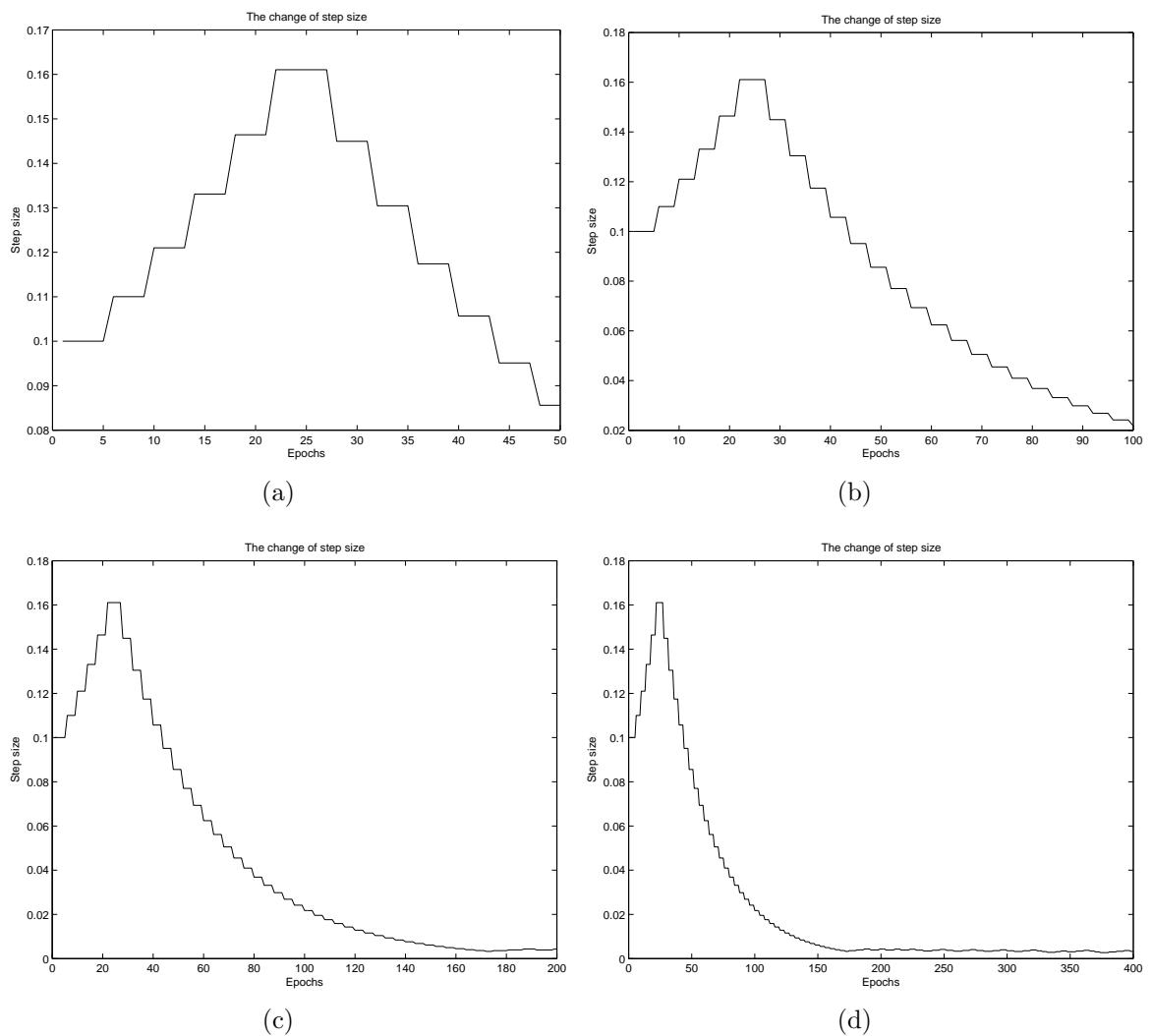


Fig. 4.33: *The change of step size curve in the training processing:* (a) 50 epochs; (b) 100 epochs; (c) 200 epochs; (d) 400 epochs.

Table 4.2: *MSE between the original image and the restoration image with the bell MFs in the different training epochs number.*

Epoch number	20	50	100	200	400
MSE	29.9477	13.1895	7.7370	6.9795	6.9171

Table 4.3: *MSE of filtered noise images contaminated with the different number of bell MFs for each input.*

Epoch number	2	3	4	5	6
MSE	29.9477	116.3464	4.0627	5.3917	7.2124

with ANFIS become better and better as shown in Fig 4.34(a) to 4.34(f). Meanwhile, when the RMSE descents, the effect of the restoration is better and better. We list the MSE between the original image and the restored image under high Gaussian noise in various training epoch number in Table 4.2. As the training epochs increase, the MSE decreases and the restoration image improves in quality.

The Number of Membership Function for Each Input

Now we discuss the effect of the number of bell MFs for each input to the image restoration corrupted by high Gaussian noise. We assume the number of MF for each input is 3, 4, 5 and 6 bell MFs separately. The MFs for which the number for each input are 3, 4, 5 and 6 before and after training, which reflects changes in premise nonlinear parameters are shown in Fig. 4.35 to 4.38. Fig. 4.39 shows the RMSE with the intensity of high noise. We list MSE of filtered noise image contaminated with high Gaussian noise in the different number of MF for each input in Table 4.3. The image contaminated with noise and restoration images with 2, 3, 4, 5 and 6 MFs for each input are shown in Fig A.21(a) to A.21(d).

Optimization Method Used in Training

ANFIS uses a hybrid learning algorithm or the backpropagation method to identify the MF parameters of output, Sugeno type FISs. We can choose a combination of least-squares and backpropagation gradient descent methods used for training FIS

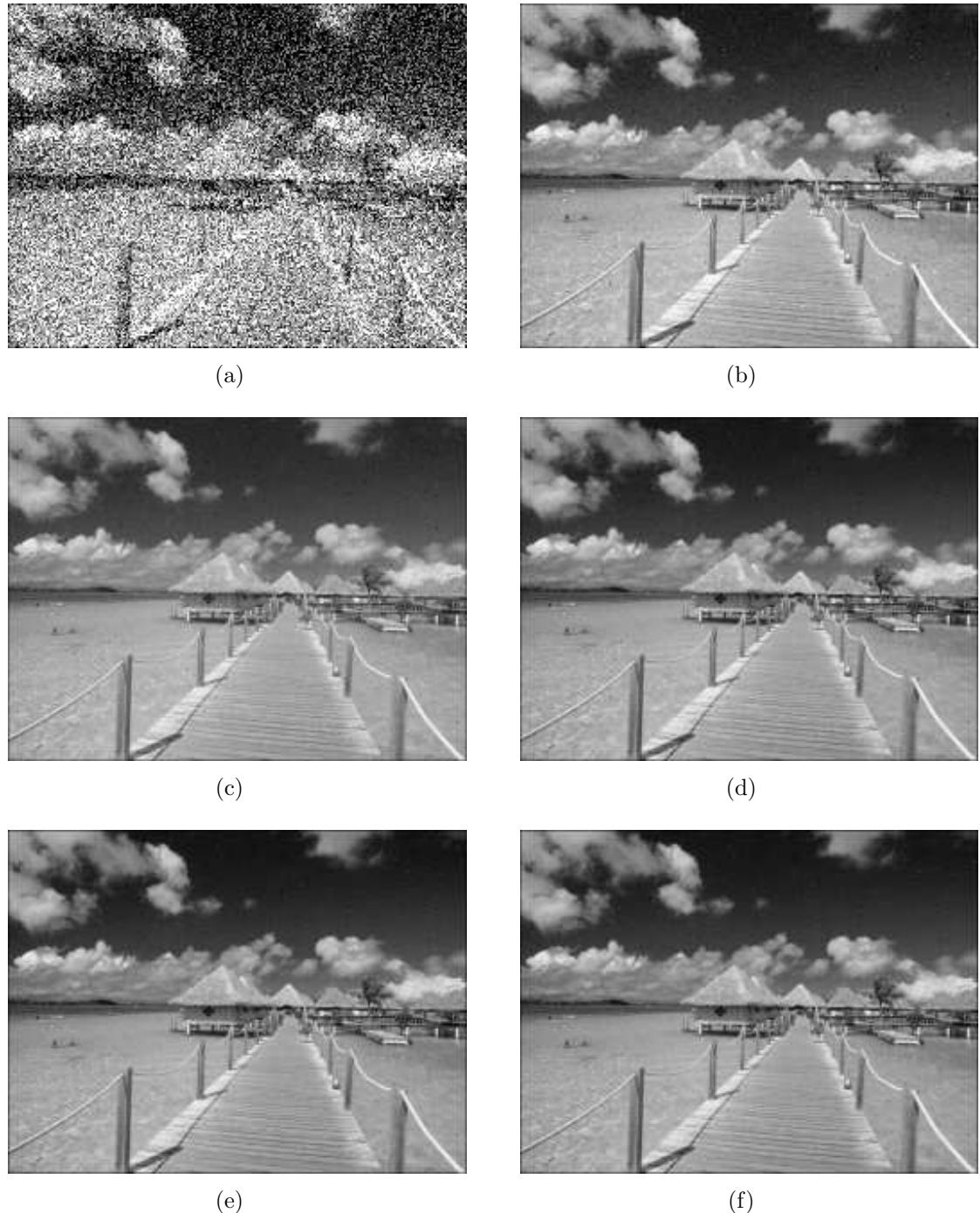


Fig. 4.34: The image corrupted by the high Gaussian noise and the results of removing noise with ANFIS of the bell MFs for the different training epoch numbers: (a) The image with high noise; (b) 20 epochs; (c) 50 epochs; (d) 100 epochs; (e) 200 epochs; (f) 400 epochs.

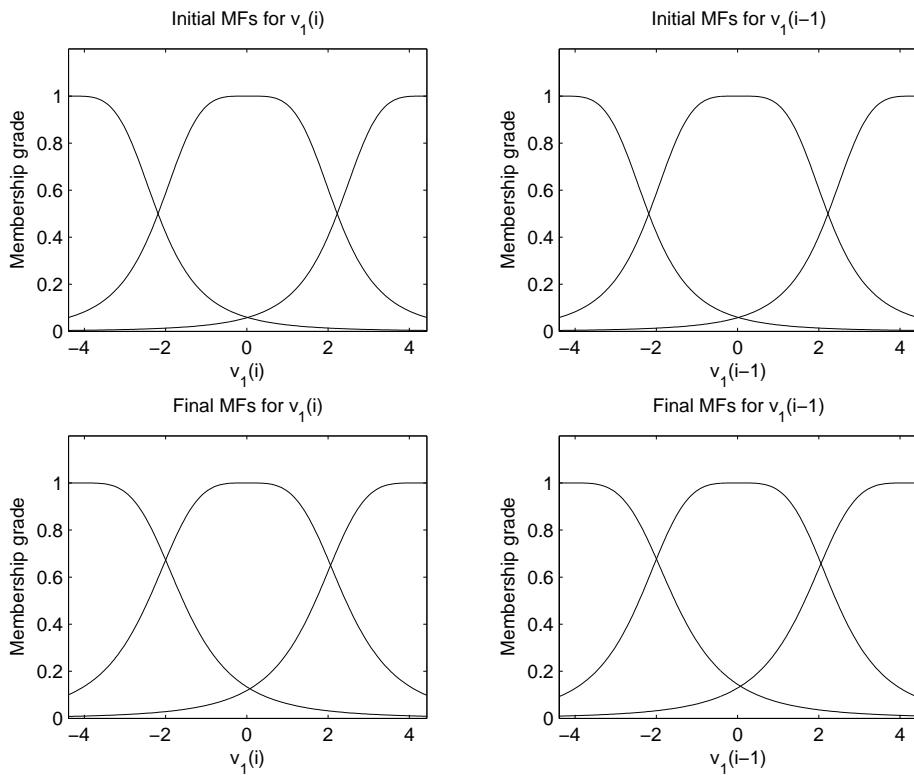


Fig. 4.35: *The changes in 3 MFs for each input before and after training.*

membership function parameters to model a given set of input/output data or just the backpropagation method.

MSE of filtered noise images by the backpropagation method and by using the hybrid learning algorithm contaminated with low, medium and high Gaussian noise are shown in Table 4.4. RMSEs and training step size using backpropagation method for low, medium and high noise are shown in Fig. 4.41 compared with Fig. 4.6 using the hybrid learning algorithm.

The restoration image with the backpropagation method compared with a hybrid

Table 4.4: *MSE of filtered noise images by the backpropagation method and by using the hybrid learning algorithm contaminated with different intensity Gaussian noises.*

	Low noise	Medium noise	High noise
MSE of the original and corrupted images	175.3222	1577.9	4383.1
MSE with the backpropagation method	50.0287	431.4803	1168.2
MSE with the hybrid learning algorithm	2.0298	11.3250	29.9477

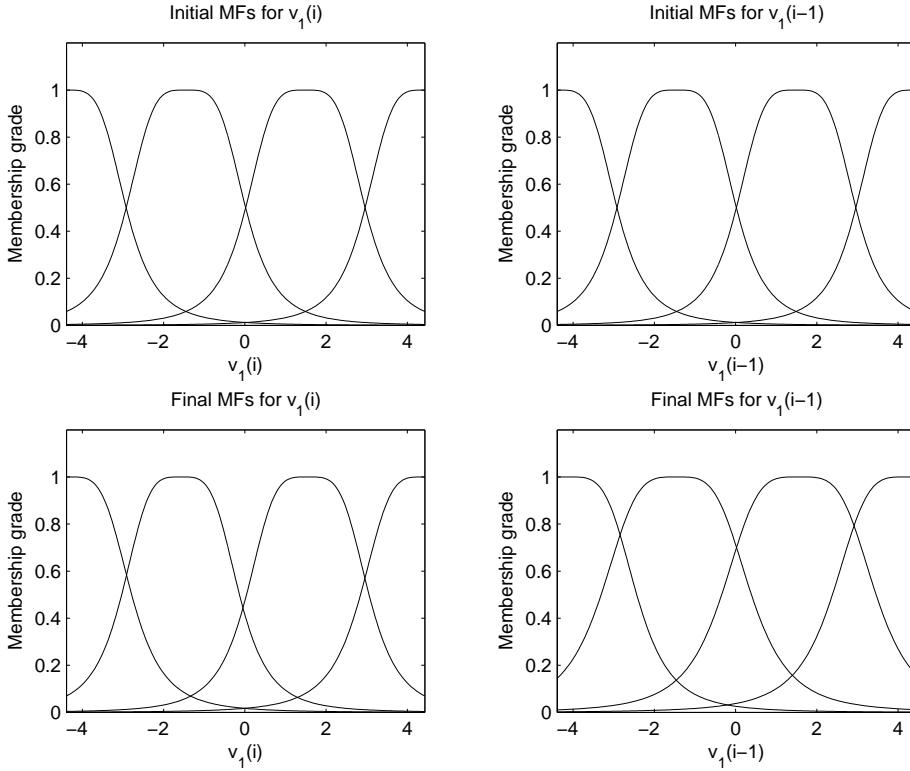


Fig. 4.36: *The changes in 4 MFs for each input before and after training.*

learning algorithm are shown in Fig 4.42(a) through 4.42(f) respectively.

From Table 4.4 and Fig. 4.42, we find that the effect for removing noise from the contaminated image by hybrid learning algorithm is much better than by the backpropagation method.

Output Membership Function Type

ANFIS usually uses *linear* or *constant* to identify the output MF type. We can choose either *linear* or *constant* used for training FIS membership function parameters to model a given set of input/output data.

MSE of filtered noise images by using the output MF type as *linear* and *constant* contaminated with the different intensity noise are shown in Table 4.5. RMSEs using the output MF type as *constant* for the different intensity noise are shown in Fig. 4.41 compared with Fig. 4.6 using the output MF type as *linear*.

The restoration image with the output MFs type as *constant* compared with as

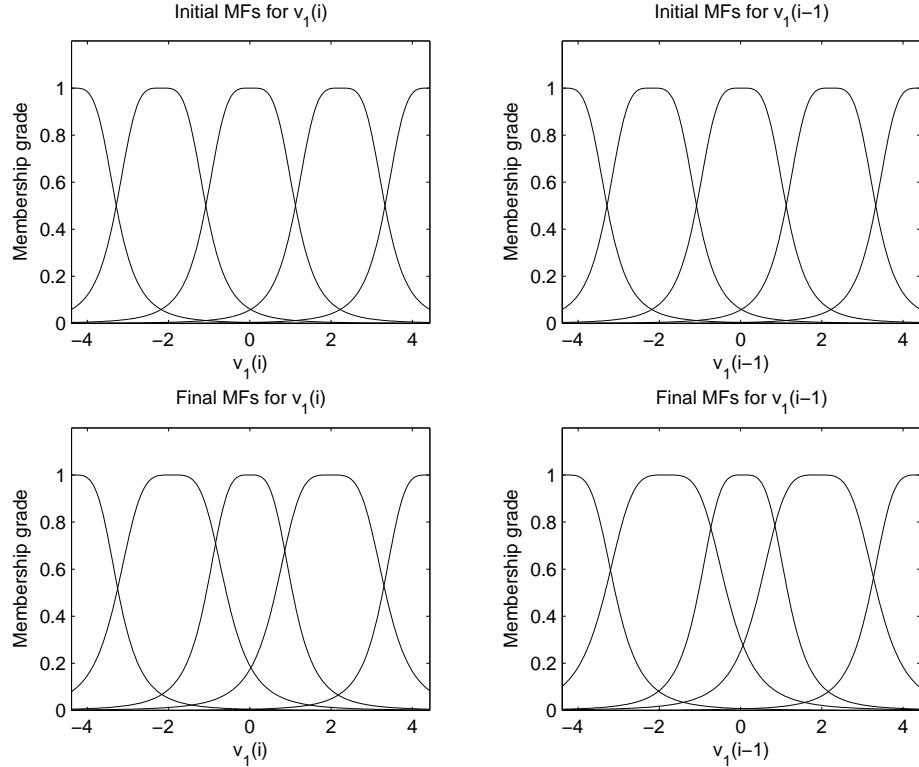


Fig. 4.37: The changes in 5 MFs for each input before and after training.

Table 4.5: MSE of filtered noise images by the output MF type as linear and constant contaminated with different intensity Gaussian noise.

	Low noise	Medium noise	High noise
MSE of the original and corrupted images	175.3222	1577.9	4383.1
MSE by the output MF type as <i>constant</i>	14.7049	130.0012	360.6174
MSE by the output MF type as <i>linear</i>	2.0298	11.3250	29.9477

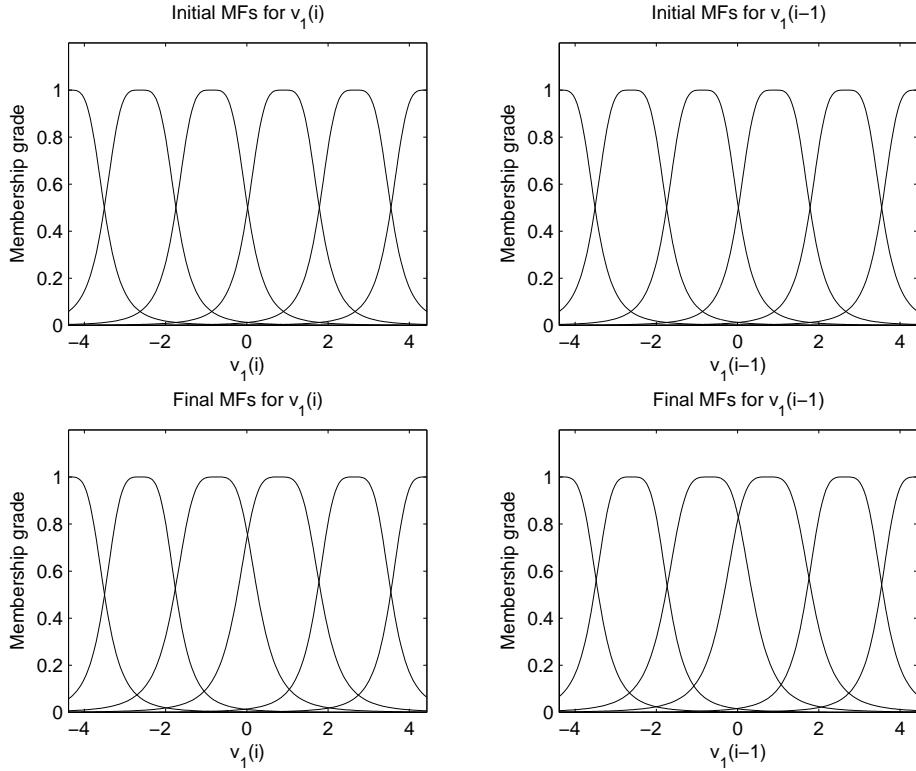


Fig. 4.38: *The changes in 6 MFs for each input before and after training.*

linear are shown in Fig 4.44(a) through 4.44(f) respectively.

From Table 4.5 and Fig. 4.44, we find that the effect for removing noise from the contaminated image by the output MF type of *linear* is much better than by the output MF type of *constant*.

Training Data and Checking Data

We can split the whole data into two halves. One half is called the training data and another half the checking (validation) data. The checking data is to detect overfitting of the training data set. The checking data has the same format as the training data. Overfitting can be detected when the checking error starts increasing while the training error is still decreasing.

We first show the RMSE curve Fig. 4.45 when the image is corrupted by high noise. Overfitting can not be detected because when the training error is slightly decreasing from 0.232 to 0.226, the checking error also descends slight from 0.2305 to

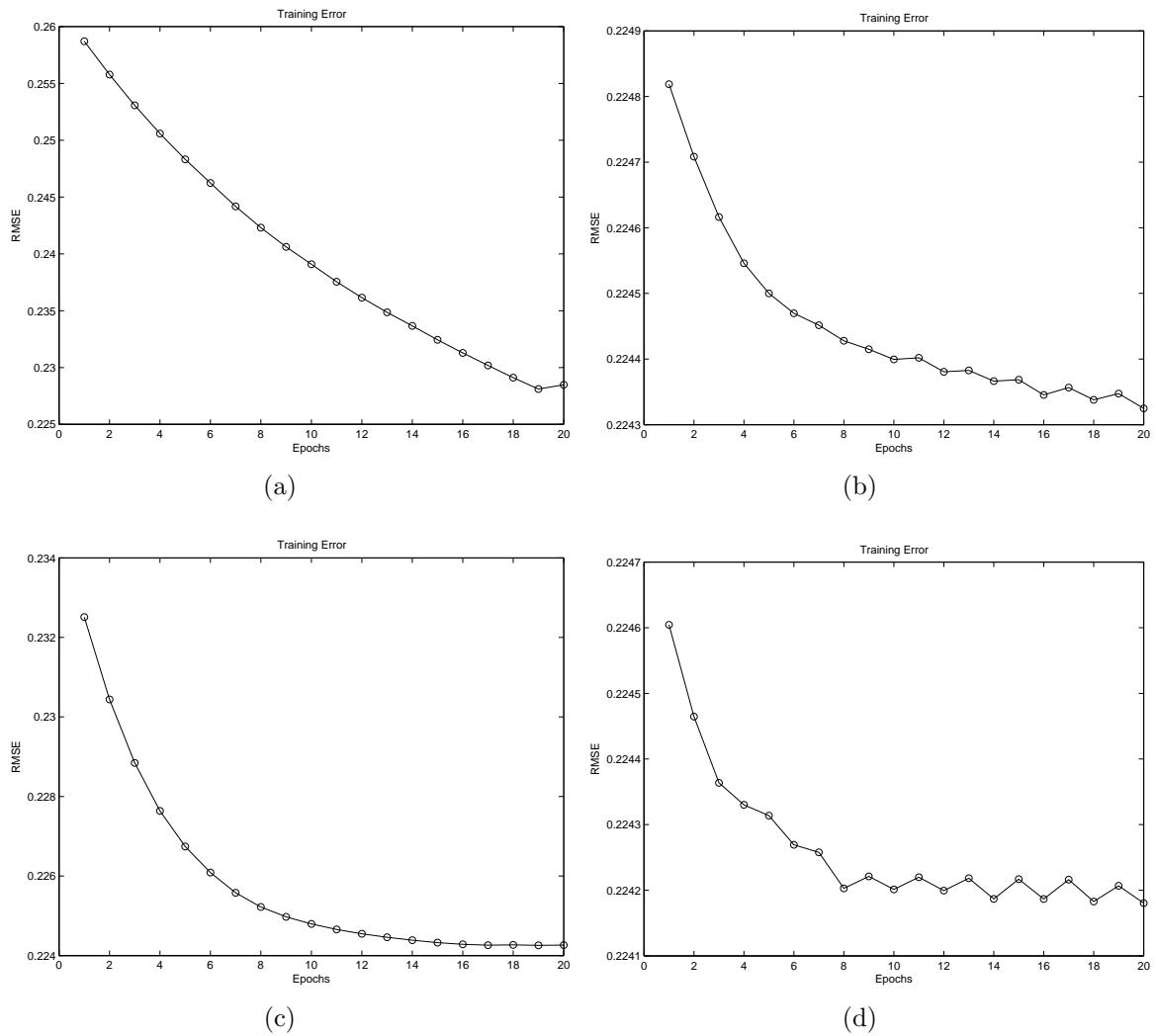


Fig. 4.39: The RMSE of ANFIS removing high Gaussian noise with the different number of bell MFs for each input: (a) 3 MFs; (b) 4 MFs ; (c) 5 MFs; (d) 6 MFs.

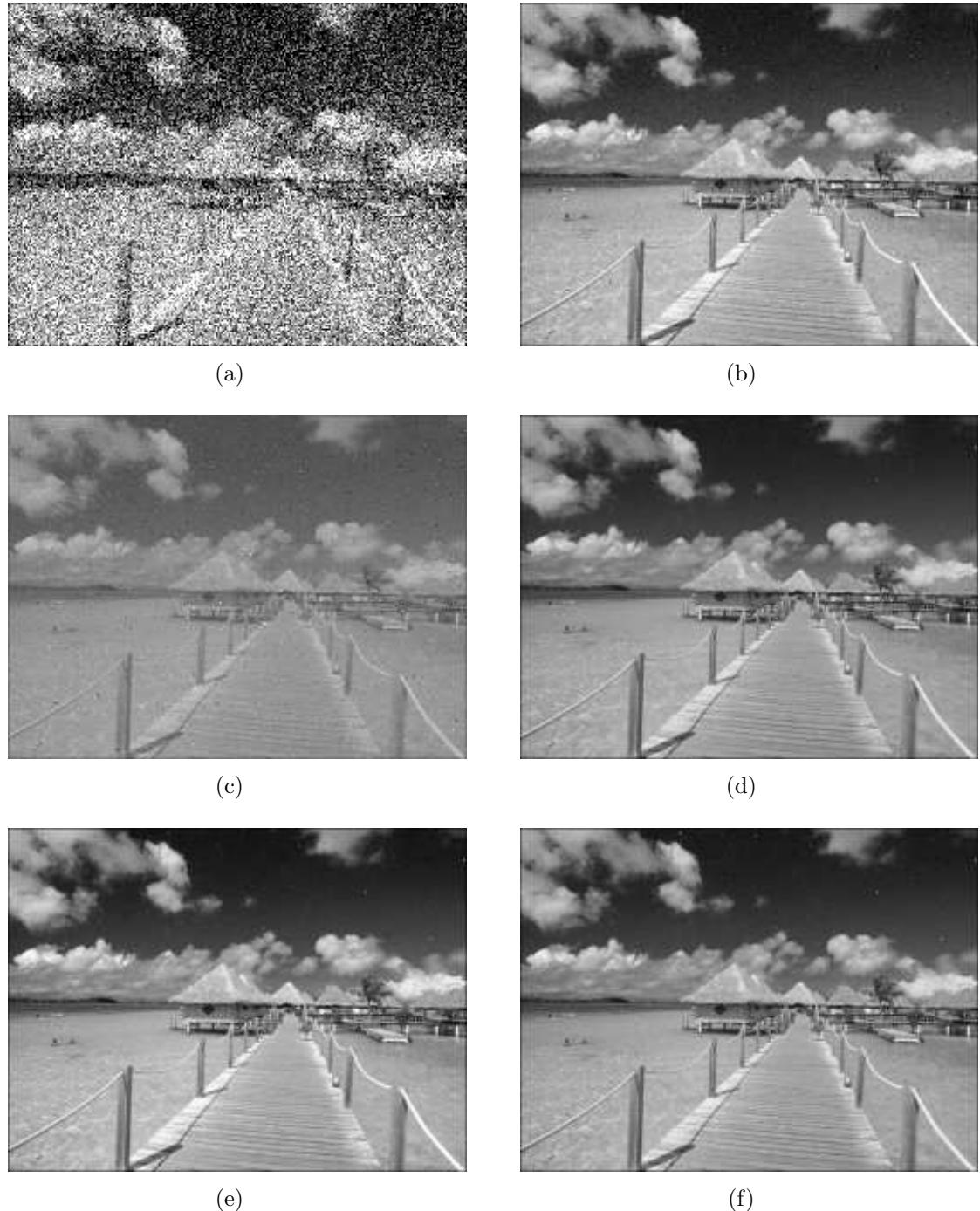


Fig. 4.40: *The images corrupted by the high Gaussian noise and the results of removing noise with ANFIS in the different number of the bell MFs for each input: (a) The image with high noise; (b) 2MFs; (c) 3MFs; (d) 4MFs; (e) 5MFs; (f) 6MFs.*

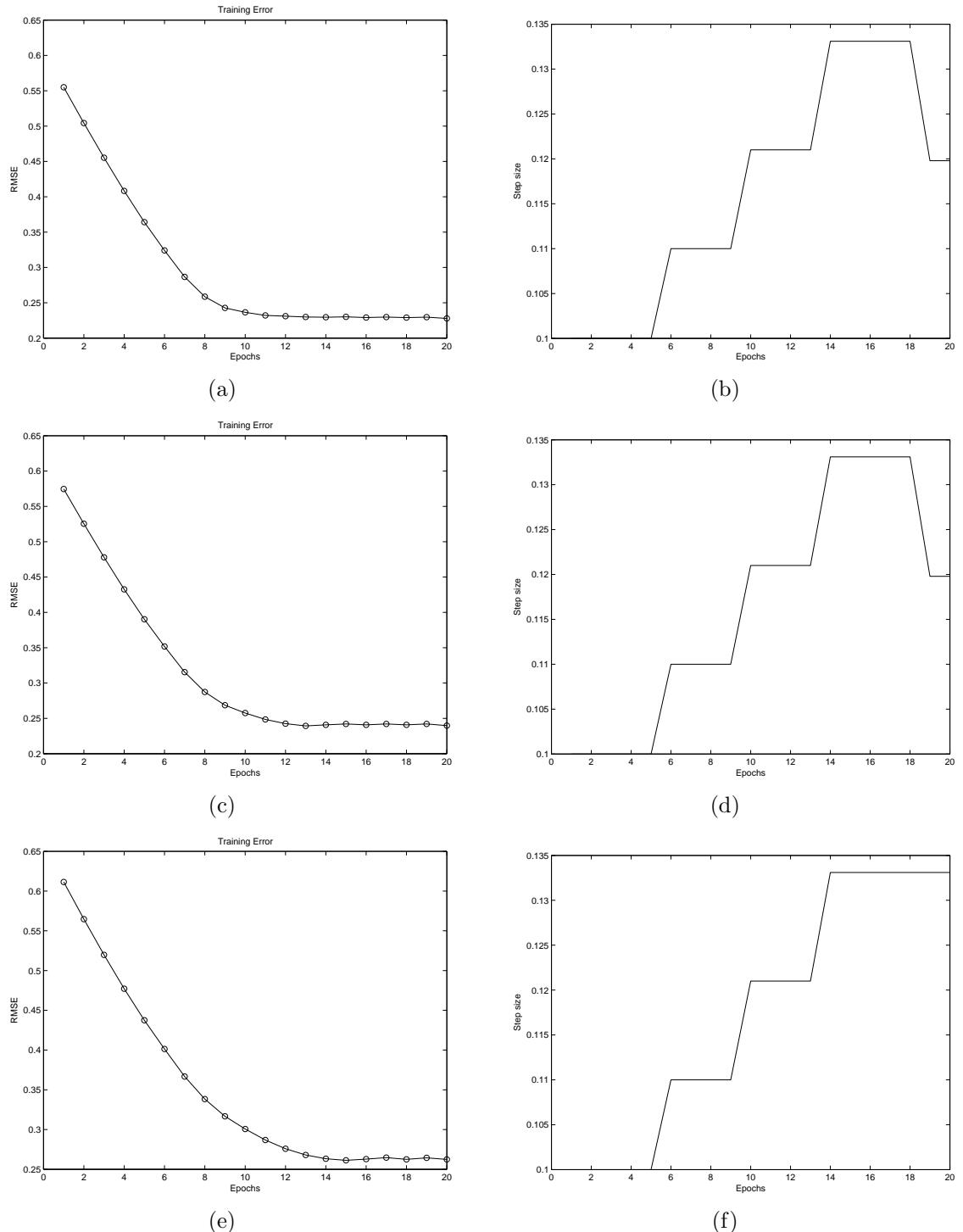


Fig. 4.41: RMSE curves and training step size of backpropagation method: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.

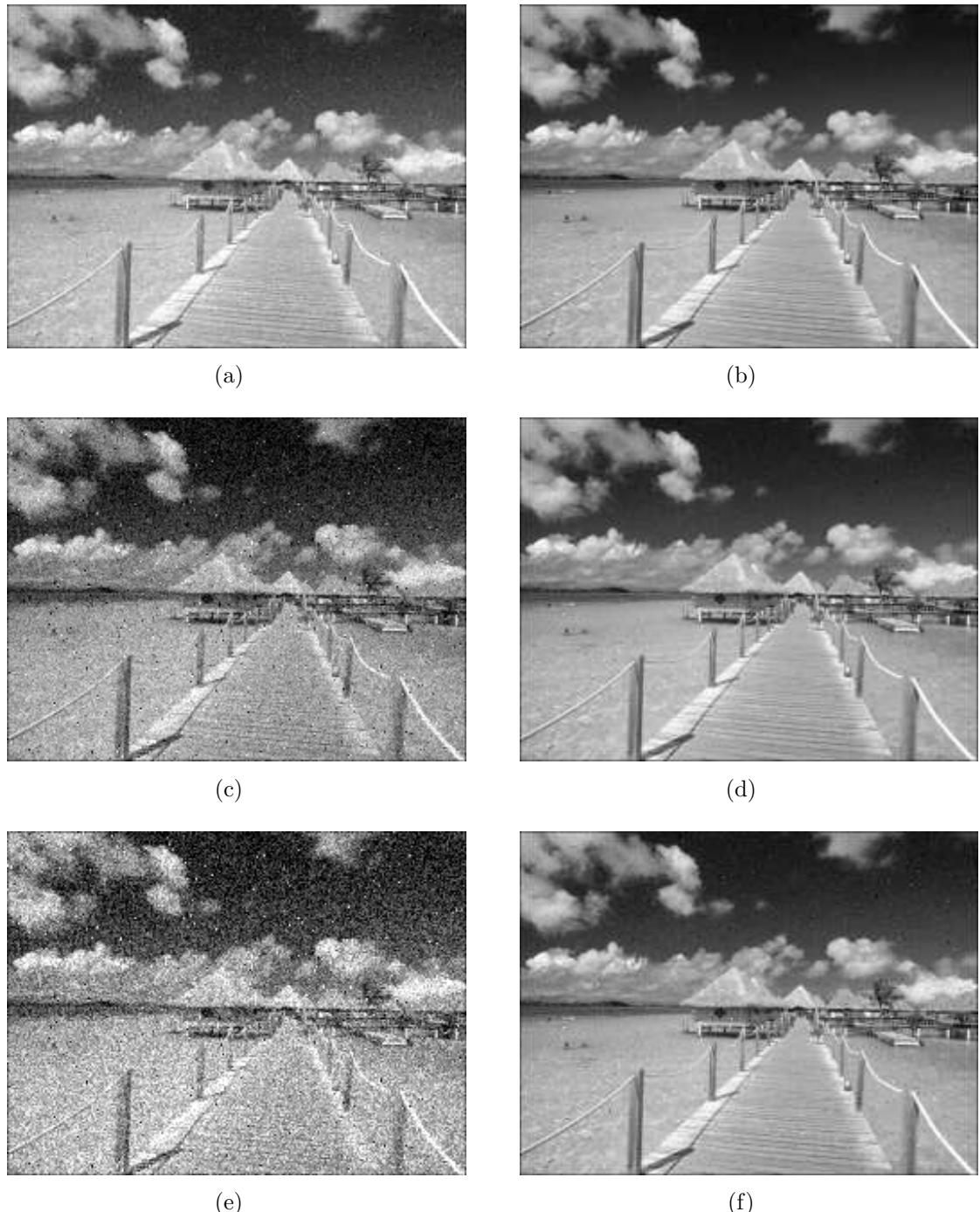


Fig. 4.42: *The images corrupted by different intensity noises restored by the back-propagation method and by a hybrid learning algorithm with ANFIS of the bell MFs: (a), (c) and (e) Backpropagation method restored from low, medium and high noise respectively; (b), (d) and (f) Hybrid learning algorithm restored from low, medium and high noise respectively.*

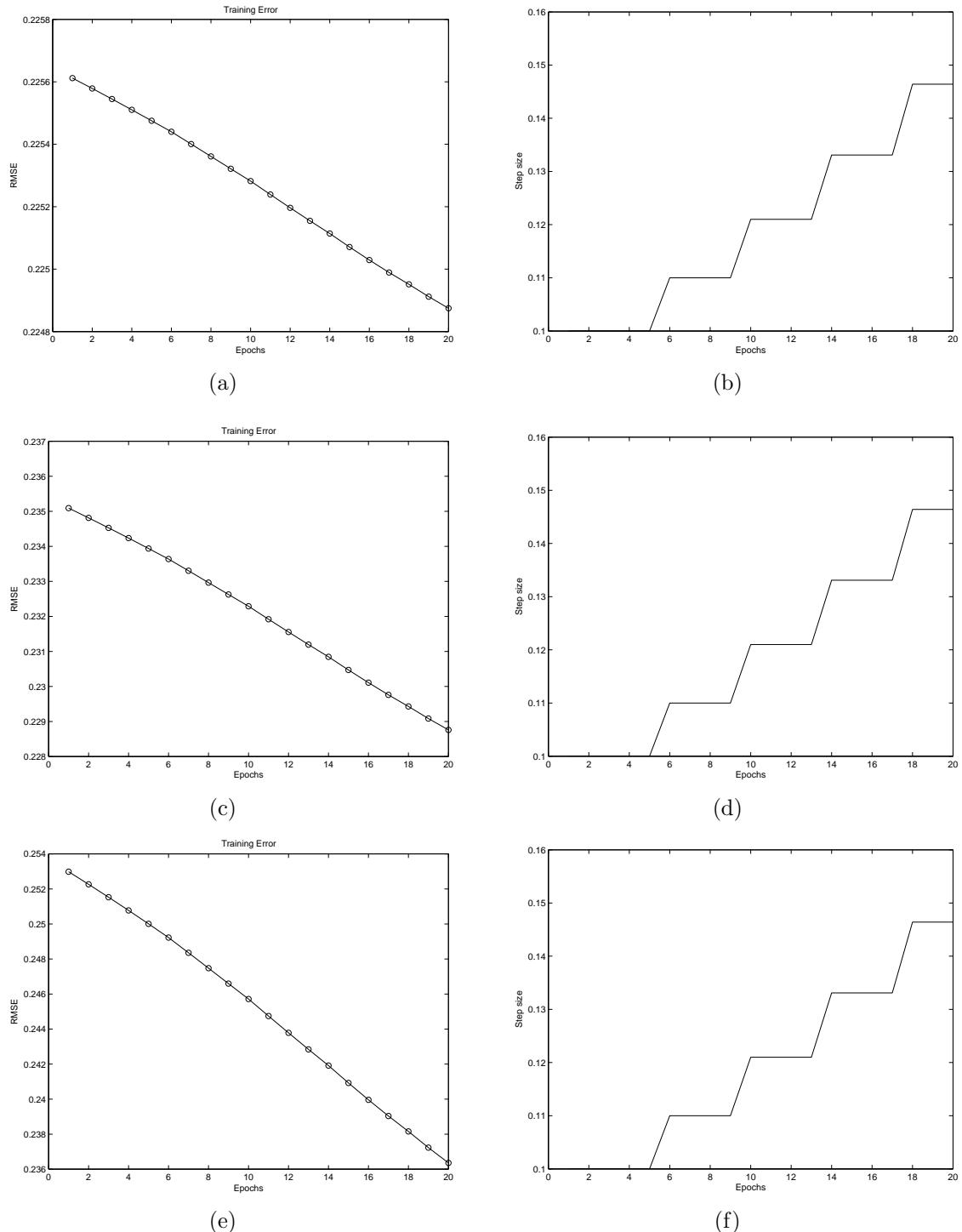


Fig. 4.43: RMSE curves and training step size of the output MF type as constant: (a), (c) and (e) RMSE curves to low, medium and high noise; (b), (d) and (f) The training step size for low, medium and high noise.

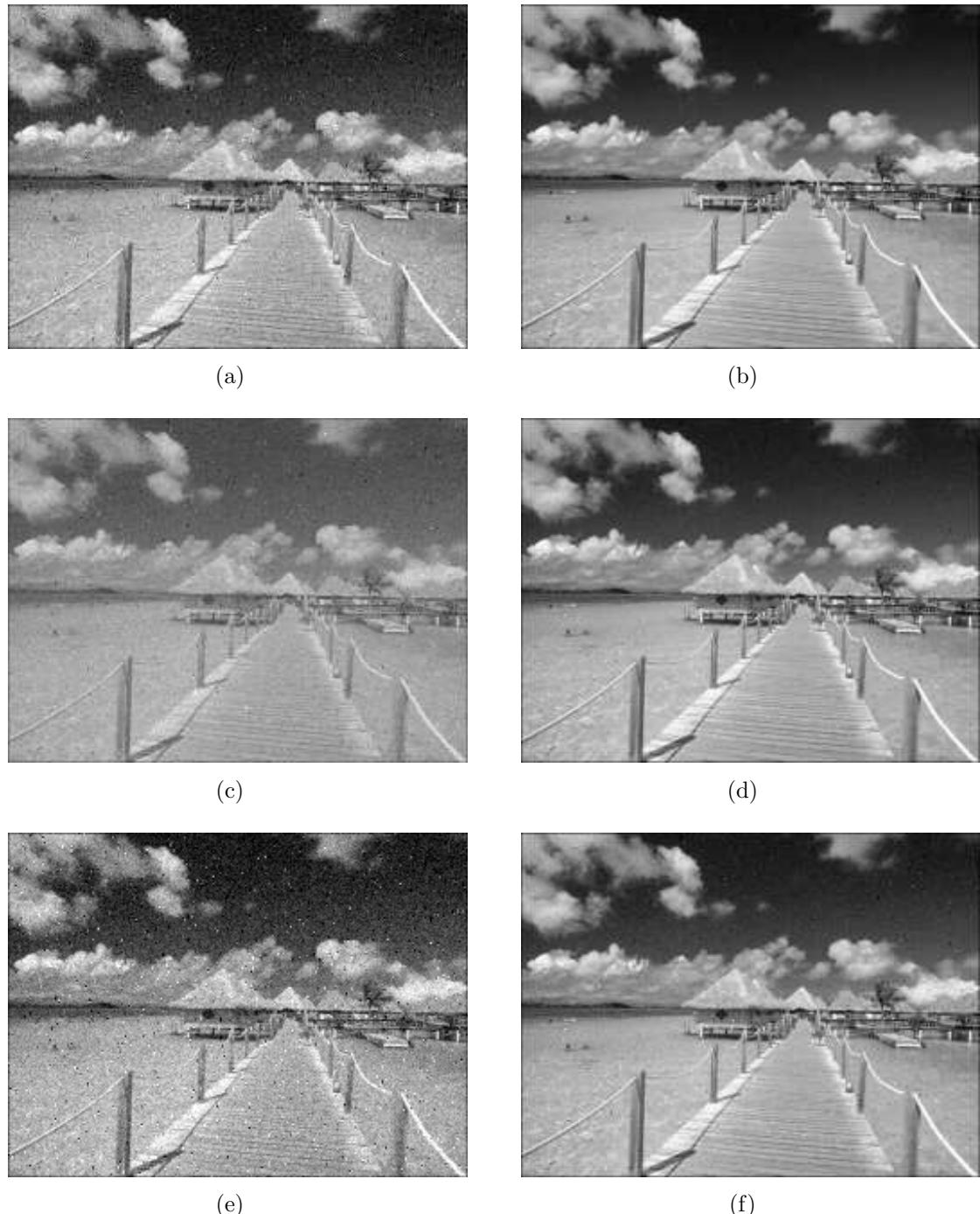


Fig. 4.44: The images corrupted by the different intensity noise restored by the output MF type of constant and linear with ANFIS of the bell MFs: (a), (c) and (e) using the output MF type of constant restored from low, medium and high noise respectively; (b), (d) and (f) using the output MF type of linear restored from low, medium and high noise respectively.

0.225.

We also shows the changes in the bell MFs of the training data and checking data before and after training in Fig 4.46.

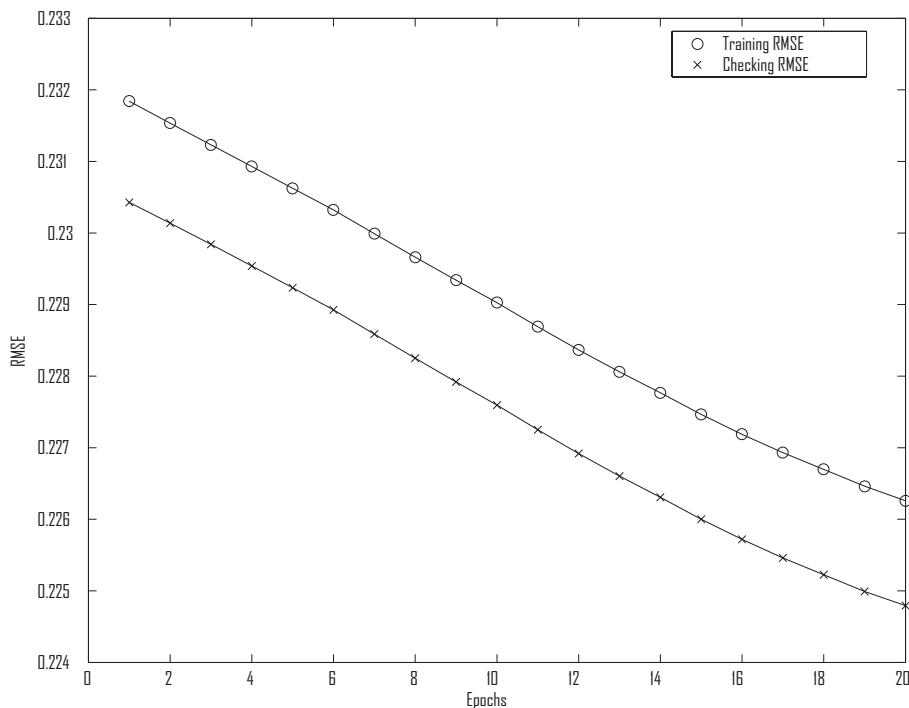


Fig. 4.45: *The changes in RMSEs of the training data and the checking data.*

Summary of All Membership Functions

With the increase of training epoch, the effect become better and better. Even though it still reduces noise a little better and better after 100 epoches, the effect of removing noise is not much. However, on the other hand, it consumes much more time. So it is enough of training within 100 epoches. The result from the number of MFs for each input reflects the complexity of ANFIS for choosing parameters. When the number of MFs for each input is 2 or 3, it is not enough to reflect the complex structure of the data, therefore the MSE of filtered image is large. However, when we choose the number of MFs for each input as 5 or 6, it produces the redundancy for the structure of data, therefore the MSE slightly increases. Choosing the number of MFs for each input as 3 is the best matching with the complex structure of data. It is

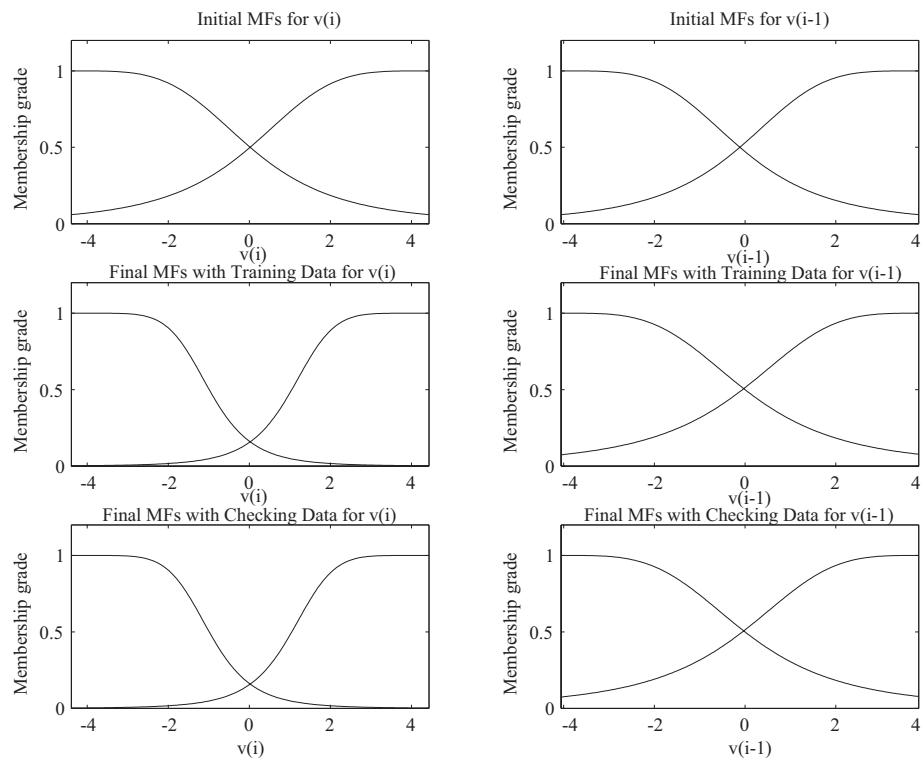


Fig. 4.46: *The changes in the MFs of the training data and checking data before and after training.*

important for the structure of the system to match the available data. Therefore, to build the model with ANFIS, the choice of data must express the whole properties for the system. The choice for the system structure should have enough parameters to reflect all the characteristics. However, the numbers of parameters should also be restrained. It is by no means true that the more complex the structure, the better the effect. The structure should match with the data. We need to decide the structure according to the experience or by changing it to observe the effect in special application. For the optimization method, the effect of choosing a combination of least-squares and backpropagation gradient descent methods is much better than that of the backpropagation method. Choosing the *linear* output MF type is much better than choosing the *constant* output MF type. We can observe that the changes of RMSE between the checking data and the training data are all decreasing and match well after training. It proves that no overfitting happens in the ANFIS training process. The difference between the checking RMSE and the training RMSE is due to the fact that the image pixel value is discrete and jumps especially at the edge of objects.

4.2 Application of ANFIS to the Nonlinear Passage Dynamics of Order 3 to Restore an Image Corrupted by Gaussian Noise

In the experiment, the unknown nonlinear passage dynamics of order 3 instead of the unknown nonlinear passage dynamics of order 2 in the first experiment is assumed to be defined as

$$\begin{aligned} v_0(i) &= f(v_1(i), v_1(i-1)) \\ &= \frac{8 \sin(v_1(i)) v_1(i-1) v_1(i-2)}{1 + (v_1(i-1))^2 + (v_1(i-2))^2} \end{aligned} \quad (4.3)$$

where $v_1(i)$ is a noise source and $v_1(i-1)$ and $v_1(i-2)$ is the one unit delay and two unit delay of the noise source, $v_0(i)$ is defined as the resultant of the nonlinear passage dynamics $f(\cdot)$ owing to $v_1(i)$, $v_1(i-1)$ and $v_1(i-2)$, i is from 1 to the number of the pixels in the image.

We choose SNRs that are equal to 12.2625, 2.7201 and -1.7169 to represent the intensities of low, medium and high Gaussian noises respectively.

We still use the RGB image ‘*Matirapoint*’ for comparison. All settings are the same as in a nonlinear passage dynamics of order 2 and then transferred to the gray image for experiments except the ANFIS input variables changed from 2 to 3.

4.2.1 Application of ANFIS with the Default Bell MFs

We only discuss the default bell MF of the unknown nonlinear passage dynamics of order 3. We also list MSE between the original image and the restoration image with all types of MFs for the nonlinear passage dynamics of order 3 to limit the size of thesis and discussion on parameters of ANFIS. For more details of the results with the other types of MFs, we show them in Appendix A.

We first investigate the behavior of these signals in the frequency domain before we go on to the next. Fig. A.1(a) to A.1(d) display the spectral density distributions of $s(i)$, $v_1(i)$, $v_0(i)$ and $x(i)$, respectively, from the first 256 points. Like the first experiment, the spectra of the information signal $s(i)$ and the distorted noise $v_0(i)$ overlap each other in a large frequency area. Therefore, it is impossible to apply common frequency domain filtering methods to remove $v_0(i)$ from $x(i)$.

Because the differences of the MFs in the restoration processing are almost the same to low, medium and high noise, we just show the changes of the bell MFs for high noise in Fig A.2 before and after training. A comparison of the nonlinear passage dynamics of order 3 with those of order 2 reveals that the only difference is the number of the MFs for the input changing from 4 to 6.

The image corrupted by the intensity of low, medium and high Gaussian noise and the results of removing noise with ANFIS are shown in Fig A.3(a) through A.3(f) respectively. The result is that the restoration effect is good compared with the original image in Fig. 4.1(b) no matter how badly the image is corrupted by noise.

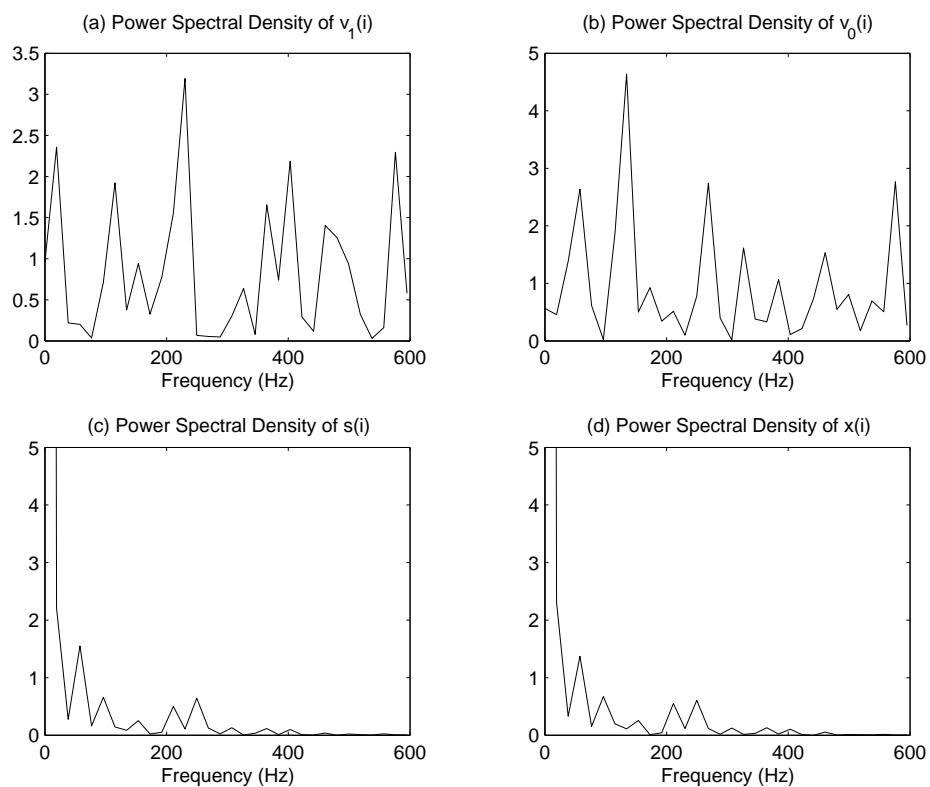


Fig. 4.47: Spectral density distributions: (a) Information signal $s(i)$; (b) Source noise signal $v_1(i)$; (c) Distorted noise signal $v_0(i)$; (d) Measurable output signal $x(i)$.

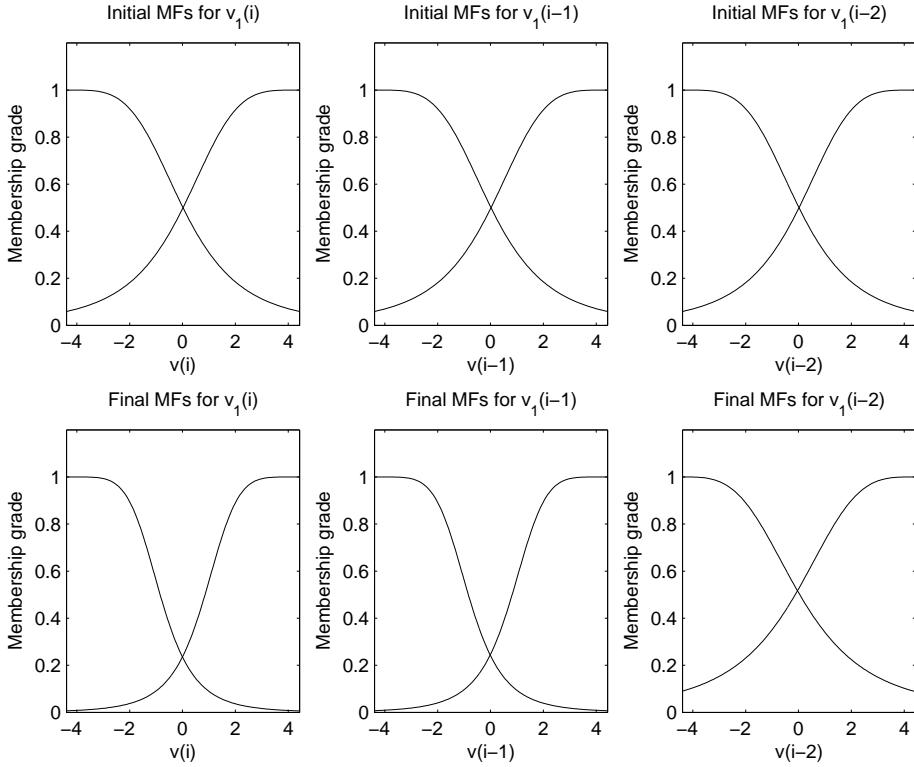


Fig. 4.48: *The changes in bell MFs before and after training.*

4.2.2 Summary of All Membership Functions

Similar to the nonlinear passage dynamics of order 2, all MSE results processed with different MFs are listed in descending order in Table 4.6 for comparison. We find the order has some differences compared with that of the nonlinear passage dynamics of order 2. The image restoration is the best with the pi-shaped MF. The difference between two sigmoidal MF, the product of two sigmoid MF, and the two-sided gaussianMF are slight worse than the pi-shaped MF. The Gaussian MF and the trapezoidal MF are poorer with respect to the effectiveness of the above mentioned MFs in removing noise from the corrupted images. The poorest MF is triangle MF. These results can also obtained by examining the restoration images when they are contaminated with same noise.

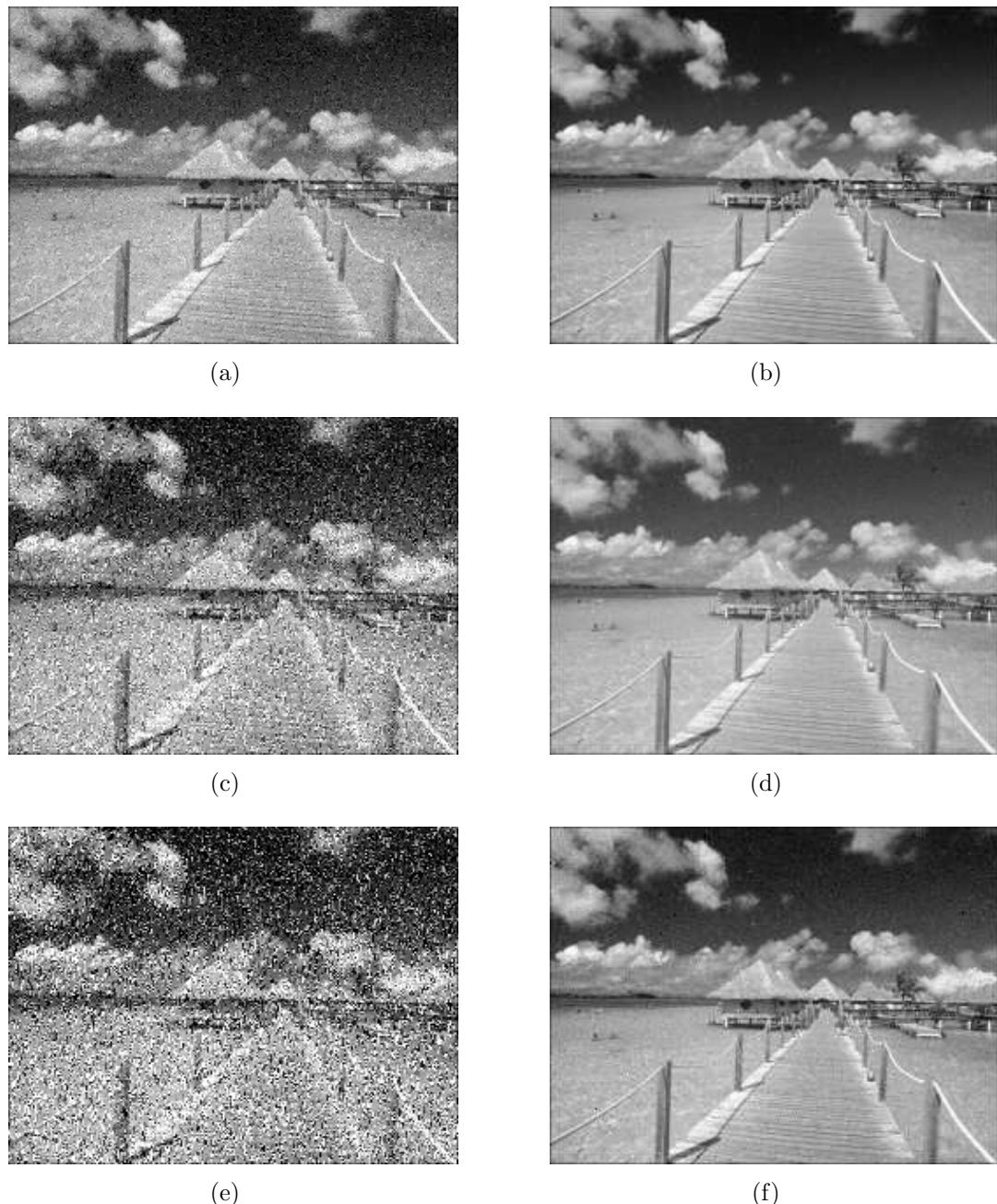


Fig. 4.49: The image corrupted by different intensity noises and the results of removing noises with ANFIS of bell MFs: (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

Table 4.6: *MSE between the original image and the restoration image with all types of MFs for the nonlinear passage dynamics of order 3.*

Name of MFs	Intensity of noise (MSE)		
	Low	Medium	High
	194.4897	1750.4	4862.2
MSE between the original image and the corrupted image	194.4897	1750.4	4862.2
Triangle	67.2136	587.5628	1615.1
Gaussian	7.1746	50.5327	137.4984
Trapezoidal	7.5388	40.1960	108.3532
Bell	4.1796	21.2186	55.4988
Product of two sigmoid	4.1881	18.1754	46.4375
Difference between Two Sigmoidal	4.1881	18.1754	46.4375
Two-sided gaussian	4.5509	17.7161	46.2868
Pi-shaped	3.7880	17.2451	41.9444

Table 4.7: *MSE between the original image and the restoration image with the bell MFs in the different training epoch numbers.*

Epoch number	20	50	100	200	400
MSE	55.4988	44.3938	44.0614	43.8616	43.7009

4.2.3 Discussion of Parameters of ANFIS

The MSE between the original image and the restored image under high Gaussian noise in different training epoch number is listed in Table 4.7.

The MSE of filtered noise image contaminated with high Gaussian noise in the different number of MF for each input in Table 4.8.

MSE of filtered noise images by the backpropagation method and by using the hybrid learning algorithm contaminated with low, medium and high Gaussian noise

Table 4.8: *MSE of filtered noise images contaminated with the different number of bell MFs for each input.*

Epoch number	2	3	4
MSE	55.4988	284.5468	21.9081

Table 4.9: *MSE of filtered noise images by the backpropagation method and by using the hybrid learning algorithm contaminated with different intensity Gaussian noise.*

	Low noise	Medium noise	High noise
MSE between original and corrupted images	194.4897	1750.4	4862.2
MSE with the backpropagation method	128.3111	495.6110	1833.2
MSE with the hybrid learning algorithm	4.1796	21.2186	55.4988

Table 4.10: *MSE of filtered noise images by the output MF type as linear and constant contaminated with different intensity Gaussian noise.*

	Low noise	Medium noise	High noise
MSE between original and corrupted images	194.4897	1750.4	4862.2
MSE by the output MF type as <i>constant</i>	14.7049	130.0012	360.6174
MSE by the output MF type as <i>linear</i>	13.3121	114.9718	318.2671

are shown in Table 4.9.

MSE of filtered noise image by using the output MF type as *linear* and *constant* contaminated with the different intensity noise are shown in Table 4.10.

We can split the whole data into two halves and overfitting can not be detected because when the training error is slightly decreasing, the checking error also descend a little.

With the increasing of training epoch, the effect become better and better. However, the results from the number of MFs for each input reflects the complexity of ANFIS for choosing parameters. When the number of MFs for each input is 2, it is not enough to reflect the complex structure of data, therefore the MSE of filtered image is big. However, when we choose the number of MFs for each input as 3, it produces redundancy for the structure of data, so the MSE has a great increasing. Choosing the number of MFs for each input as 3 is the best matching with the complex structure of data. It is important that the structure of the system match the appropriate data. Therefore, to build the model with ANFIS, the choice of data must express the whole properties for the system. The choice for the system structure should have enough parameters to reflect all the characteristics. However, the

number of parameters should also be restrained. It is by no means valid to say that the more complex the structure, the better the effect. The structure should match with the data. We need to decide the structure according to the experience or by changing it to observe the effect in special application. For the optimization method, the effect of choosing a combination of least-squares and backpropagation gradient descent methods is much better than that of the backpropagation method. Choosing the *linear* output MF type is much better than choosing the *constant* output MF type. We can observe that the change of RMSE between the checking data and the training data are all decreasing and matches well after training. This proves that no overfitting happens in ANFIS training process. The difference between the checking RMSE and the training RMSE results from the fact that the image pixel value is discrete and jumps especially at the edge of objects.

4.3 Application of ANFIS to the Nonlinear Passage Dynamics of Order 2 to Restore an Image Corrupted by Salt and Pepper Noise

Like the general rule we assume before, we generally assign the number of MFs per input as two, output MF type as *linear*, type of MF as *bell*, and training epoch number as 20. Usually the bell MFs are used as defaulted MFs. In general, the initial step size, step size decrease rate, and step size increase rate are 0.1, 0.9 and 1.1 respectively. A combination of least-squares and backpropagation gradient descent methods are used for training MF parameters to model a given set of input/output data. The output membership function type is *linear*. In the experiment, the unknown nonlinear passage dynamics are assumed to be defined as in Eq.(4.1).

We analyze the restoration of an image contaminated with high salt & pepper noise for which SNR is -1.4564 because the changes between the different intensities of noise after restoration reach almost the same effect. In the other word, its restoration effect is much better than that of the image contaminated to the same extent with Gaussian noise. We will discuss the reasons later.

Fig A.26(a) shows the original RGB (red, green and blue) color image ‘*Maldives*’. We use the same method to transfer it to the gray iamge. The resulting image after processed is shown in Fig A.26(b).

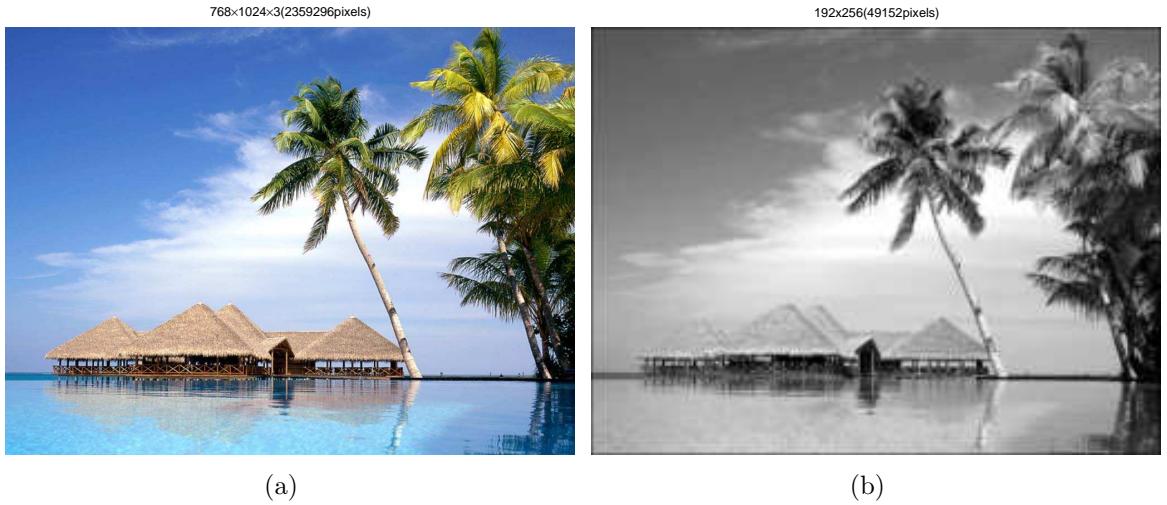


Fig. 4.50: *The original image: (a) The original color image; (b) The original gray image.*

The measurable source noise is salt & pepper noise for which the probability of the distorted noise signal $v_0(i)$ is 0.3017 shown in Fig. A.27(a). The distorted noise $v_0(i)$ caused by the source noise $v_1(i)$ and produced by the nonlinear dynamics of order 2 in Eq.(4.1) is shown in Fig. A.27(b). We show the estimated distorted signal $y_0(i)$ and the estimated error in Fig A.27(c) and Fig. A.27(d) respectively. We can find Fig. A.27(d) showing black because the the estimated error between the estimated distorted noise $y(i)$ by ANFIS and the distorted noise $v_0(i)$ is near zero matrix (less than 10^{-3}).

We now investigate the behavior of these signals in the frequency domain before we go on to the next. Fig. A.46(a) to A.46(d) display the spectral density distributions of $s(i)$, $v_1(i)$, $v_0(i)$ and $x(i)$, respectively, from the first 256 points. Obviously, the spectra of the information signal $s(i)$ and the distorted noise $v_0(i)$ overlap each other considerably. This makes it impossible to apply common frequency domain filtering methods to remove $v_0(i)$ from $x(i)$. Fig A.29 is the ANFIS surface $\hat{f}(.)$ after 20 epochs of batch learning.

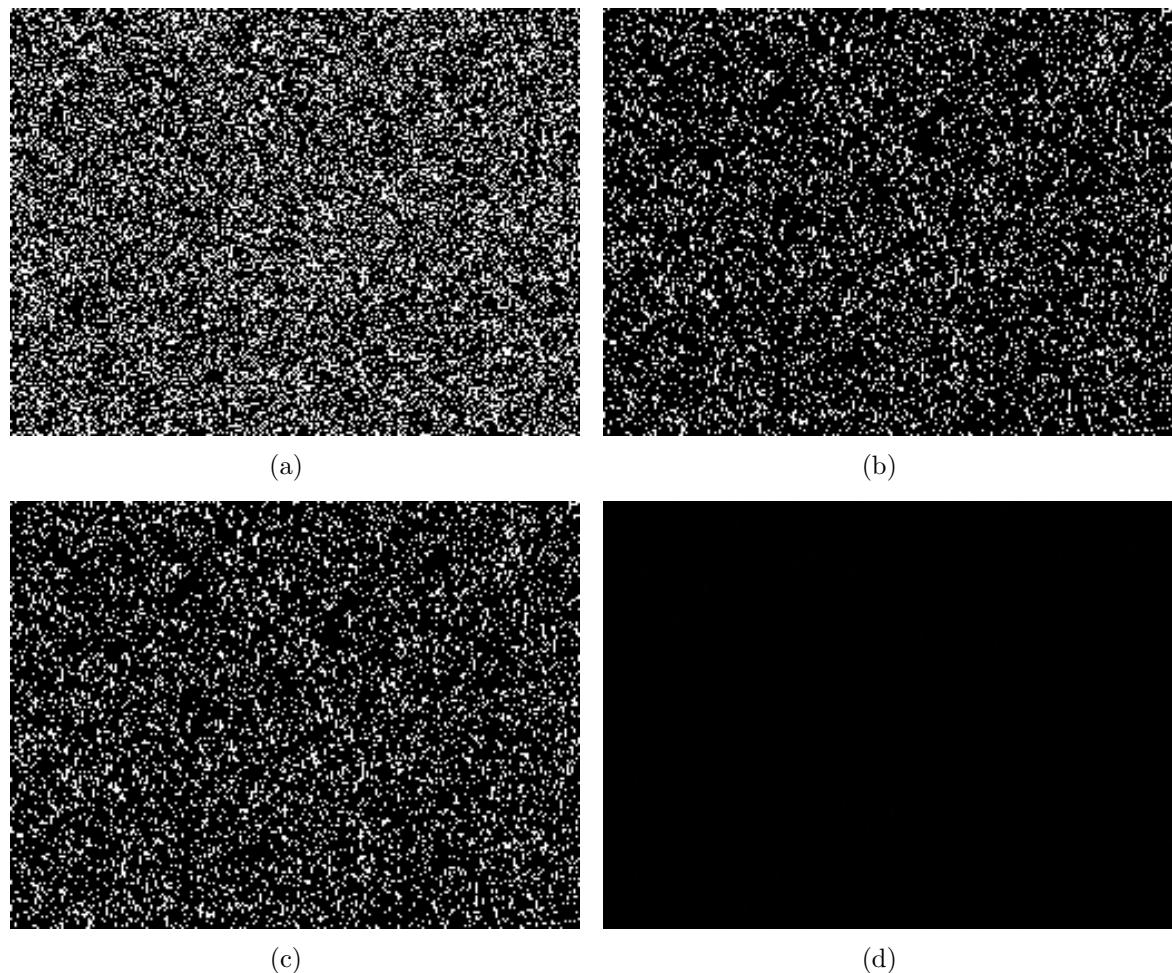


Fig. 4.51: *The signals being used in ANFIS: (a) The measurable source noise $v_1(i)$; (b) The distorted noise $v_0(i)$; (c) The estimated distorted noise $y(i)$ by ANFIS; (d) The error between the estimated distorted noise $y(i)$ by ANFIS and the distorted noise $v_0(i)$.*

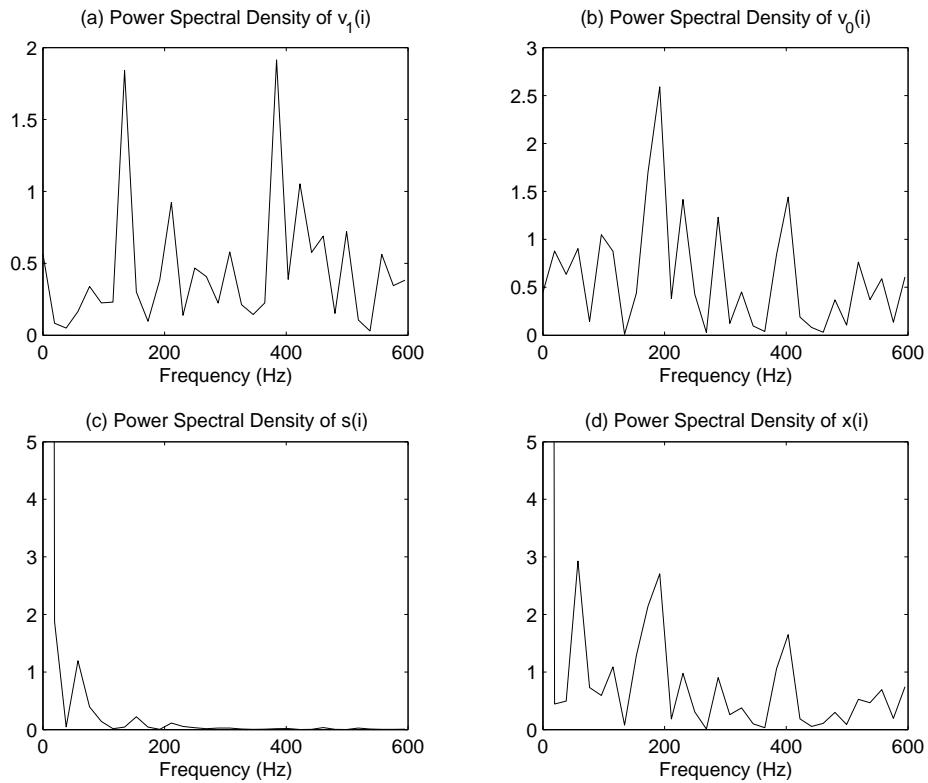
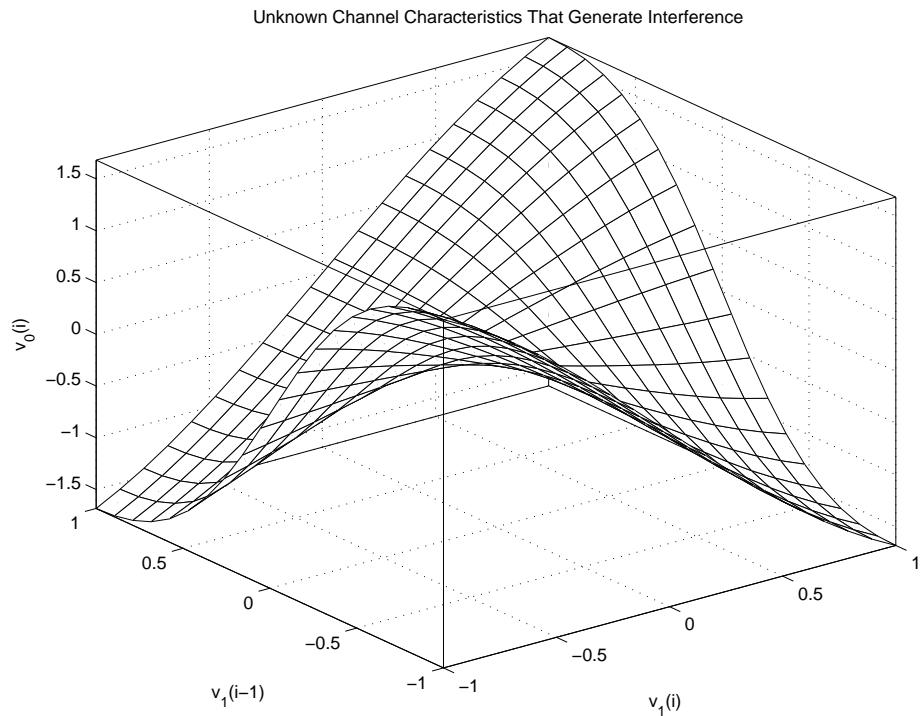


Fig. 4.52: Spectral density distributions.

Fig. 4.53: The characteristics of ANFIS function \hat{f} .

In order to visually display the extent of the error between the estimated distorted noise $y(i)$ and the actual distorted noise $v_0(i)$, we change the two dimensional pixel matrix into one dimensional vector shown in Fig. A.30. We find the error is particularly small not beyond 10^{-2} .

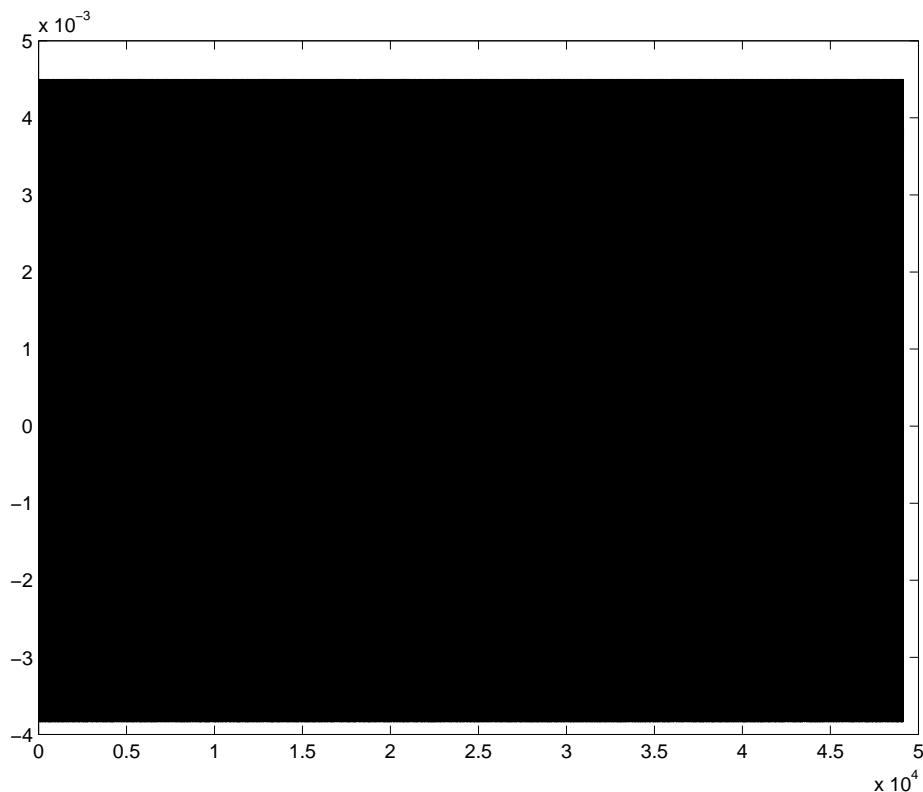


Fig. 4.54: *The estimated errors with ANFIS corrupted by high salt & pepper noise.*

4.3.1 Application of ANFIS with the Default Bell MFs

We only discuss the default bell MF of the unknown nonlinear passage dynamics of order 2. We also list MSE between the original image and the restoration image with all types of MFs for the nonlinear passage dynamics of order 2 to limit the size of thesis and discussion on parameters of ANFIS. For more details of the results with the other types of MFs, we show them in Appendix A.

First we choose the bell membership function as MFs of this ANFIS. There are two inputs, $v_1(i - 1)$ and $v_0(i)$, for nonlinear passage dynamics of orders 2 and two MFs for each input. We choose probabilities are 0.1008, 0.1812 and 0.3017 correspondent

to low, medium and high salt & pepper noise respectively. The images corrupted by salt & pepper noise and the results of removing noise with different PDF with ANFIS are shown in Fig A.31(a) through A.31(f) respectively.

Because the change is very small, we just compare the case when the image corrupted by the heavy noise for which the probability is 0.3017 and ignore the affection of the other light noise.

4.3.2 Summary of All Membership Functions

All MSE results processed with different MFs are listed in descending order in Table 4.11. We find that the effect of removing salt & pepper noise is much better than that of removing Gaussian noise. The reason is that salt & pepper noise has a distinguishing characteristic — its pixels values are always equal to *minimum* 0 or *maximum* 255 in 8 bit image. This gives ANFIS a easy mark to identify a pixel value as noise or signal. Unlike identifying the Gaussian noise, two difficulties arise in distinguishing the real white color pixels, for which the value is 255, from the salt noise, for which the value also is 255 and distinguishing the real black color pixels, for which the value is 0, from the pepper noise, for which the value also is 0.

The image restoration is the best with the pi-shaped MF and Two-sided Gaussian MFs. The difference between two sigmoidal MF, the product of two sigmoid MF, and the triangle MF are the worst three MFs. The Gaussian MF and the trapezoidal MF are between the above mentioned MFs.

4.3.3 Discussion on Parameters of ANFIS

Next, now we discuss the parameters of training epoch number, the number membership function for each input, output membership function type, the optimization method, and the training data and checking data.

We list the MSE between the original image and the restored image under salt & pepper noise in the training epoch number 50,100, 200 and 400 with the bell MFs in Table 4.12.

We list the MSE of filtered noise image contaminated with salt & pepper noise in

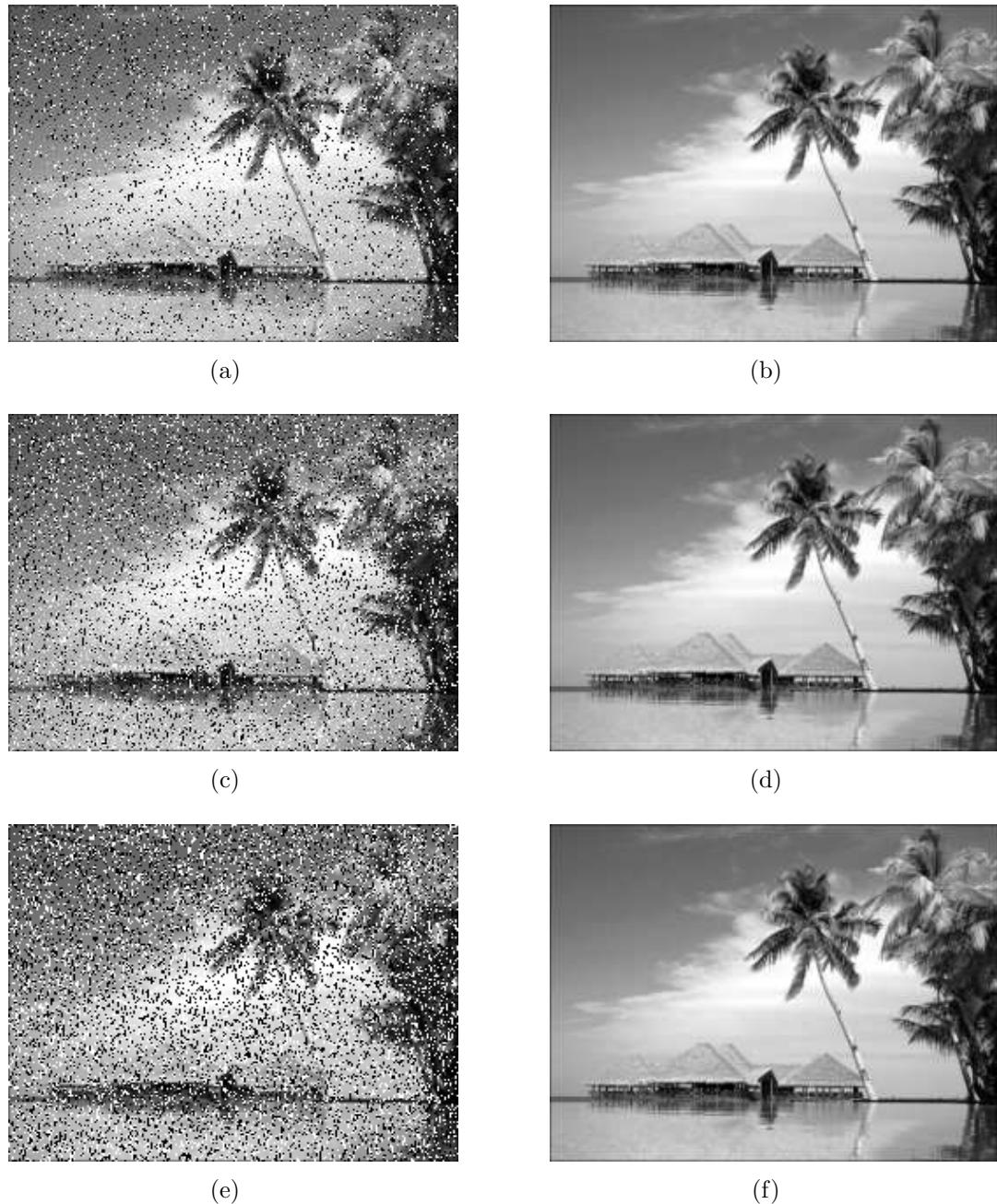


Fig. 4.55: The image corrupted by salt & pepper noises and the results of removing noises with ANFIS of bell MFs: (a) The image contaminated with low salt & pepper noise ($PDF=0.1008$); (b) The restored image; (c) The image contaminated with medium salt & pepper noise ($PDF=0.1812$); (d) The restored image; (e) The image contaminated with heavy salt & pepper noise ($PDF=0.3017$); (d) The restored image.

Table 4.11: *MSE between the original image and the restoration image with all types of MFs.*

Name of MFs	Intensity of noise (probability=0.3017)
Product of two sigmoid	1.0216
Difference between Two Sigmoidal	1.0216
Triangle	1.0204
Gaussian	0.7352
Trapezoidal	0.7495
Bell	0.7175
Pi-shaped	0.4987
Two-sided Gaussian	0.4619

Table 4.12: *MSE between the original image and the restoration image with the bell MFs in different training epoch numbers.*

Epoch number	20	50	100	200	400
MSE	0.7175	0.7171	0.7203	0.7203	0.7203

the different number of MF for each input in Table 4.13. When the number of bell MFs for each input is 4, the MSE is the least. The next lowest MSE occurs when the number of bell MFs for each input is 5.

The MSE of filtered noise images by the backpropagation method and by using the hybrid learning algorithm contaminated with high salt & pepper noise are shown in Table 4.14.

The MSE of filtered noise images by using the output MF type as *linear* and *constant* contaminated with salt & pepper noise are shown in Table 4.15. In Table 4.15, we find that the effect for removing noise from the contaminated image by the output MF type of *constant* is better than by the output MF type of *linear*.

Table 4.13: *MSE of filtered noise images contaminated with the different number of bell MFs for each input.*

Epoch number	2	3	4	5	6
MSE	0.7175	0.5260	0.1251	0.2532	0.9724

Table 4.14: *MSE between the backpropagation method and the hybrid learning algorithm.*

The optimization method	Intensity of noise (probability=0.3017)
The backpropagation method	945.9702
The hybrid learning algorithm	0.7175

Table 4.15: *MSE of filtered noise images by the output MF type as linear and constant contaminated with salt & pepper noise.*

The optimization method	Intensity of noise (probability=0.3017)
The output MF type as <i>constant</i>	0.3415
The output MF type as <i>linear</i>	0.7175

We can split the whole data into two halves and overfitting can not be detected because when the training error is slightly decreasing, the checking error also descend a little.

4.4 Application of ANFIS to the Nonlinear Passage Dynamics of Order 3 to Restore an Image Corrupted by Salt & Pepper Noise

In the experiment, we use the same general rule in Chapter 4. The unknown nonlinear passage dynamics are assumed to be defined as in Eq.(4.2).

We analyze only the restoration of image contaminated with high salt & pepper noise which SNR is -1.4564 because the changes between the different intensity noise after restoration are almost similar to the same effect. In the other words, it restoration effect is much better than that of an image contaminated to the same extent by Gaussian noise. We will discuss the reasons later.

Fig A.26(a) shows the original RGB (red, green and blue) color image ‘*Maldives*’. We use the same method to transfer it to the gray image. The resulting image after processed is shown in Fig A.26(b).

Fig. A.46(a) to A.46(d) display the spectral density distributions of $s(i)$, $v_1(i)$,

$v_0(i)$ and $x(i)$, respectively, from the first 256 points. Obviously, the spectra of the information signal $s(i)$ and the distorted noise $v_0(i)$ overlap each other considerably. This makes it impossible to apply common frequency domain filtering methods to remove $v_0(i)$ from $x(i)$.

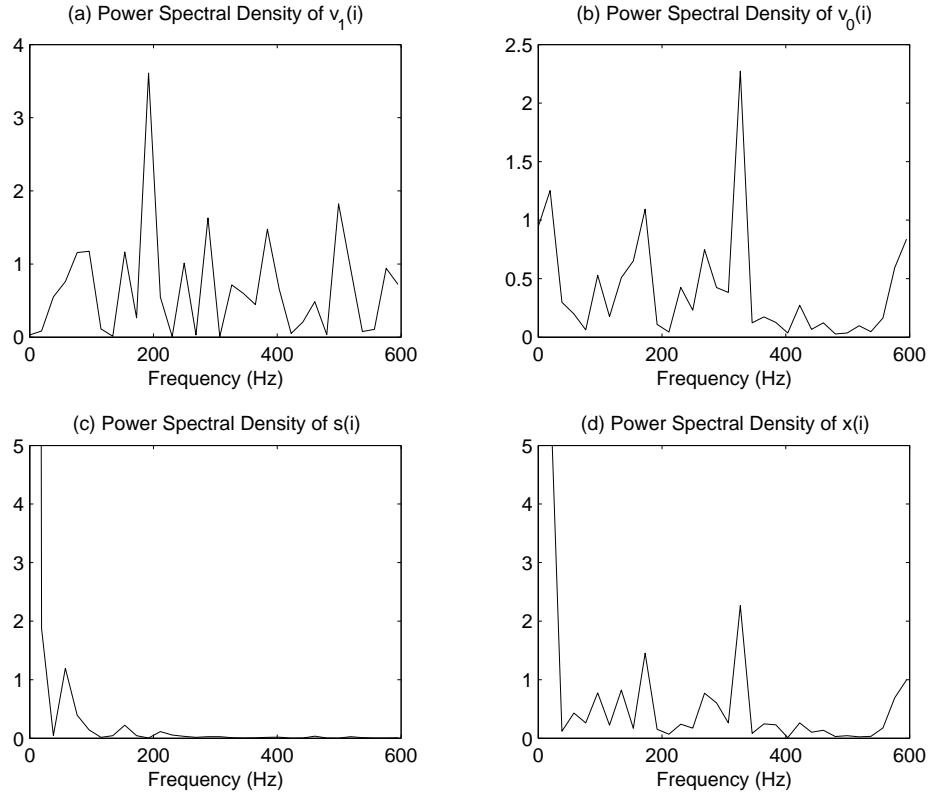


Fig. 4.56: *Spectral density distributions.*

4.4.1 Application of ANFIS with the Default Bell MFs

We only discuss the default bell MF of the unknown nonlinear passage dynamics of order 3. We also list MSE between the original image and the restoration image with all types of MFs for the nonlinear passage dynamics of order 3 to limit the size of thesis and discussion on parameters of ANFIS. For more details of the results with the other types of MFs, we show them in Appendix A.

We choose the bell membership function as MF of this ANFIS. There are three inputs, $v_1(i - 2)$, $v_1(i - 1)$ and $v_0(i)$, for nonlinear passage dynamics of orders 3 and two MFs for each input. We select probabilities are 0.2793 separately. The images

corrupted by salt & pepper noise and the results of removing noise with different PDF with ANFIS are shown in Fig A.47(a) through A.47(b) respectively.



Fig. 4.57: *The image corrupted by salt & pepper noise and the results of removing noise with ANFIS of bell MFs: (a) The image contaminated with salt & pepper noise (PDF=0.2793); (b) The restored image.*

Summary of All Membership Functions

All MSE results processing with different MFs in the nonlinear passive dynamics of order 3 are listed in descending order in Table 4.16. We find that the effect of removing salt & pepper noise is much better than that of removing Gaussian noise. The reason is that salt & pepper noise has a distinguishing characteristic — its pixels values are always equal to *minimum* 0 or *maximum* 255 in 8 bit image. This gives ANFIS a easy mark to identify a pixel value as noise or signal. Unlike identifying the Gaussian noise, the only two difficult things are to distinguish the real white color pixels, which have the value 255, from the salt noise, for which the value also is 255 and to distinguish the real black color pixels, which have the value 0, from the pepper noise, for which the value also is 0.

The image restoration is the best with the pi-shaped MF and Two-sided Gaussian MFs. The difference between two sigmoidal MF, the product of two sigmoid MF, and the triangle MF are the worst three MFs. The Gaussian MF and the trapezoidal MF are between the above mentioned MFs.

Table 4.16: *MSE between the original image and the restoration image with all types of MFs in a nonlinear passive dynamics of order 3.*

Name of MFs	Intensity of noise (probability=0.2793)
Product of two sigmoid	0.8796
Difference between Two Sigmoidal	0.8796
Triangle	1.0332
Gaussian	1.0077
Trapezoidal	0.8830
Bell	0.8703
Pi-shaped	1.0185
Two-sided Gaussian	0.8894

Table 4.17: *MSE between the original image and the restoration image with the bell MFs in the different training epoch numbers.*

Epoch number	20	50	100	200	400
MSE	0.8703	0.9096	0.9236	0.9236	0.9236

4.4.2 Discussion on Parameters of ANFIS

Like removing Gaussian noise, now we discuss the parameters of training epoch number, the number membership function for each input, output membership function type, the optimization method, and the training data and checking data.

We list MSE between the original image and the restored image under salt & pepper noise in the training epoch number 50,100, 200 and 400 with the bell MFs in Table 4.17

We list MSE of filtered noise image contaminated with salt & pepper noise in the different number of MF for each input in Table 4.18.

MSE of filtered noise images by the backpropagation method and by using the

Table 4.18: *MSE of filtered noise images contaminated with the different number of bell MFs for each input.*

Epoch number	2	3	4
MSE	0.8703	0.6853	0.2351

hybrid learning algorithm contaminated with high salt & pepper noise are shown in Table 4.19.

Table 4.19: *MSE between the backpropagation method and the hybrid learning algorithm.*

The optimization method	Intensity of noise (probability=0.3017)
The backpropagation method	341.4441
The hybrid learning algorithm	0.8703

MSE of filtered noise images by using the output MF type as *linear* and *constant* contaminated with the high salt & pepper noise are shown in Table 4.20.

Table 4.20: *MSE of filtered noise images by the output MF type as linear and constant contaminated with salt & pepper noise.*

The optimization method	Intensity of noise (probability=0.3017)
The output MF type as <i>constant</i>	0.7413
The output MF type as <i>linear</i>	0.8703

We can split the whole data into two halves and overfitting can not be detected because when the training error is slightly decreasing, the checking error also descend a little.

Chapter 5

Comparisons between ANFIS and Conventional Filters

In this chapter, we discuss the conventional filters to remove the distorted noise $v_0(i)$ generated by the nonlinear passive dynamics of 2 only because the filtered results of the distorted noise $v_0(i)$ generated by the nonlinear passive dynamics of 3 have the same properties. We separate the types of noise as Gaussian and salt & pepper as before.

We discuss the removal of Gaussian noise and salt & pepper noise using the conventional filters and wavelet. The variances of low noise, medium noise and high noise are 175.3222, 1577.9 and 4383.1 as discussed in Section 4.1. The probability of salt & pepper is 0.3017 as stated in Section 4.3.

5.1 Spatial Filtering

We compare the results of spatial filter, such as arithmetic mean filter, geometric mean filter, harmonic mean filter, contraharmonic mean filter, median filter, max and min filters, midpoint filter, alpha-trimmed mean filter and SD-ROM filter with the results of ANFIS.

5.1.1 Arithmetic Mean Filter

We use arithmetic mean filter to filter the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with arithmetic mean filter and with bell MFs in ANFIS are shown in Fig. 5.1 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We find that the filtered images are much poorer than those filtered with ANFIS. The filtered image corrupted by salt & pepper noise is particularly awful.

5.1.2 Geometric Mean Filter

The geometric mean filter is introduced to filter the images contaminated with low, medium and high Gaussian noise and heavy salt & pepper noise separately. The filtered images with geometric mean filter and with bell MFs in ANFIS are shown in Fig. 5.2 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise and are much poorer than those filtered with ANFIS. The filtered image corrupted by salt & pepper noise is terrible.

5.1.3 Harmonic Mean Filter

Now we choose harmonic mean filter to handle the images contaminated with low, medium and high Gaussian noise and heavy salt & pepper noise separately. The filtered images with harmonic mean filter and with bell MFs in ANFIS are displayed in Fig. 5.3 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We find the filtered images are much poorer than those filtered with ANFIS.

5.1.4 Contraharmonic Mean Filter

Contraharmonic mean filter is used to cope with the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered

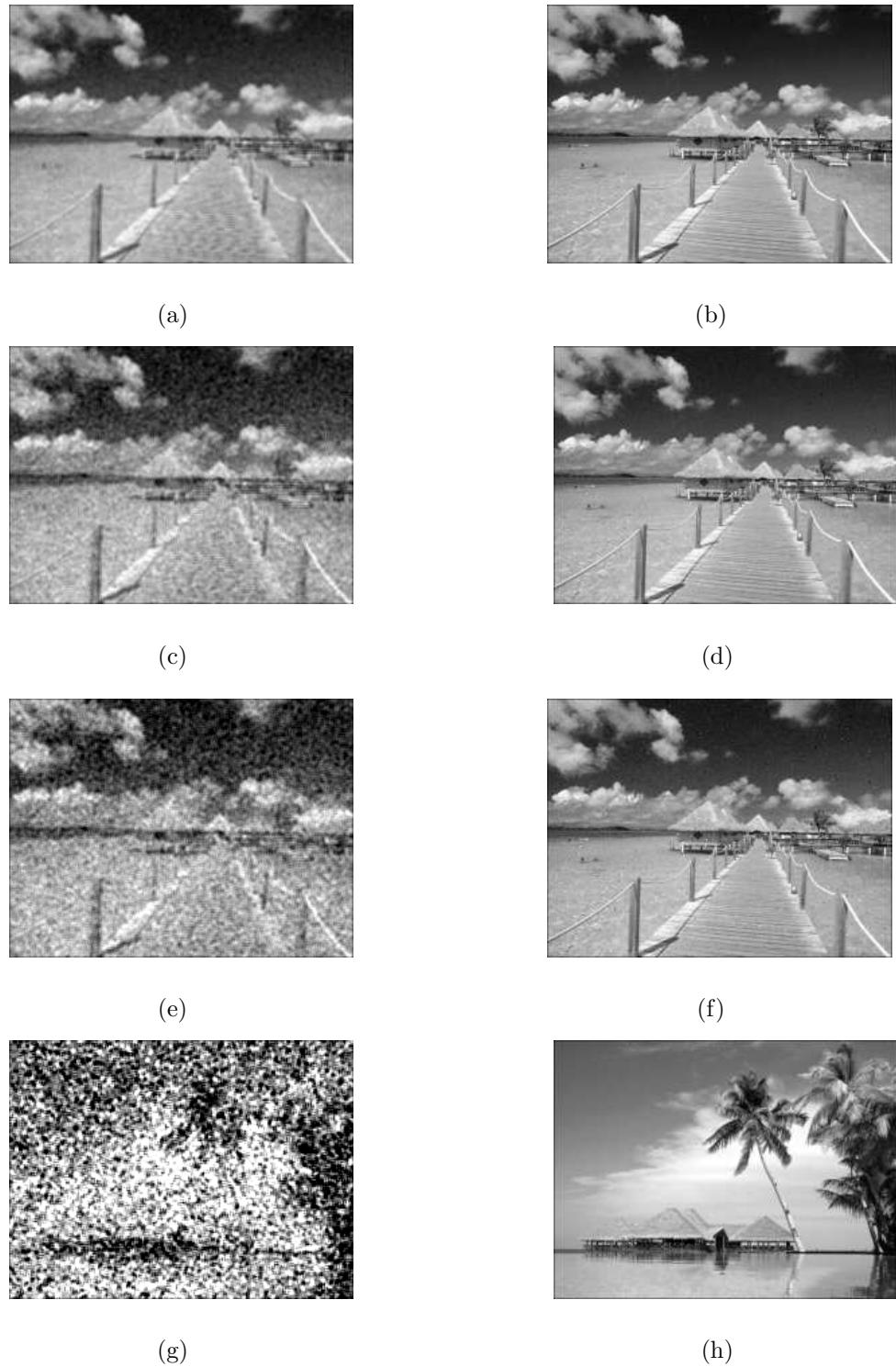


Fig. 5.1: The images filtered with arithmetic mean filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by: (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.

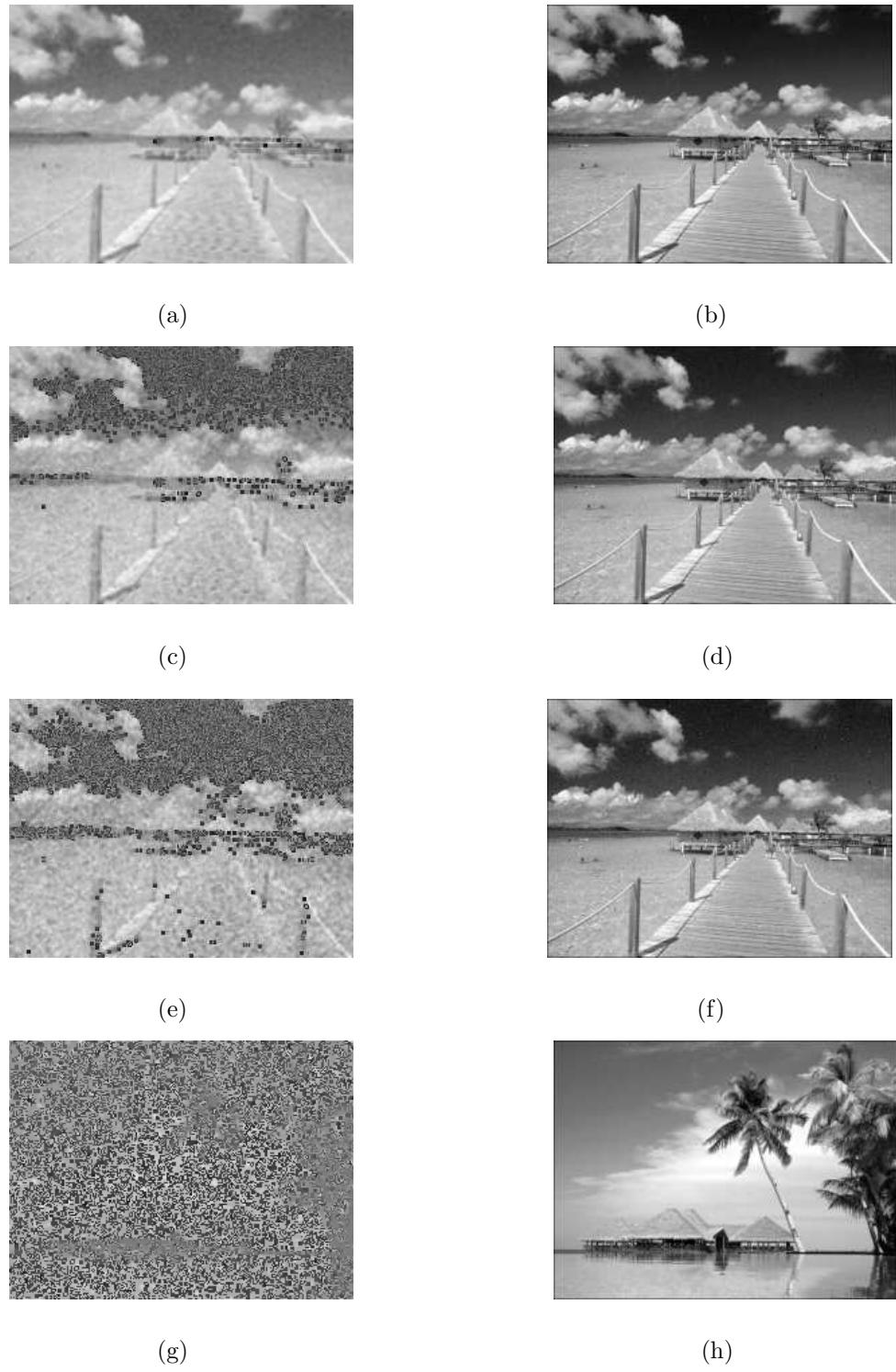


Fig. 5.2: The images filtered with geometric mean filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.

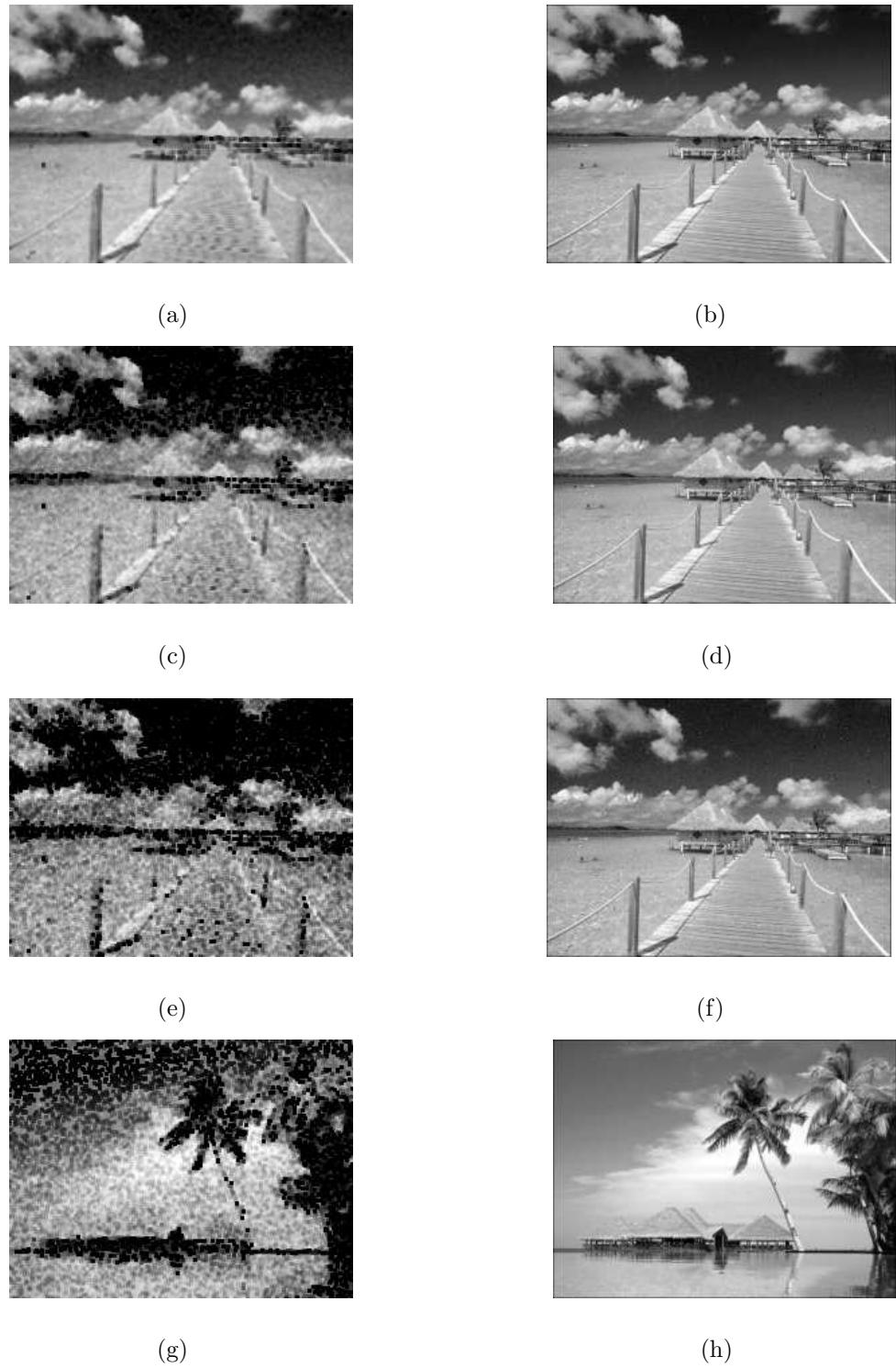


Fig. 5.3: The images filtered with harmonic mean filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.

images with contraharmonic mean filter and with bell MFs in ANFIS are shown in Fig. 5.4 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise and much poorer than those filtered with ANFIS although they are better than other conventional filters.

5.1.5 Median Filter

The most popular median filter is used to restore the images contaminated with low, medium and high Gaussian noise and heavy salt & pepper noise separately. The filtered images with median filter and with bell MFs in ANFIS are shown in Fig. 5.5 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We also can find the filtered images are much poorer than those filtered with ANFIS.

5.1.6 Max Filter

Now we choose the max filter to deal with the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with max filter and with bell MFs in ANFIS are shown in Fig. 5.6 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We find that the filtered images are much poorer than those filtered with ANFIS.

5.1.7 Min Filter

Next the min filter is used to recover the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with min filter and with bell MFs in ANFIS are shown in Fig. 5.7 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We find that the filtered images are much poorer than those filtered with ANFIS.

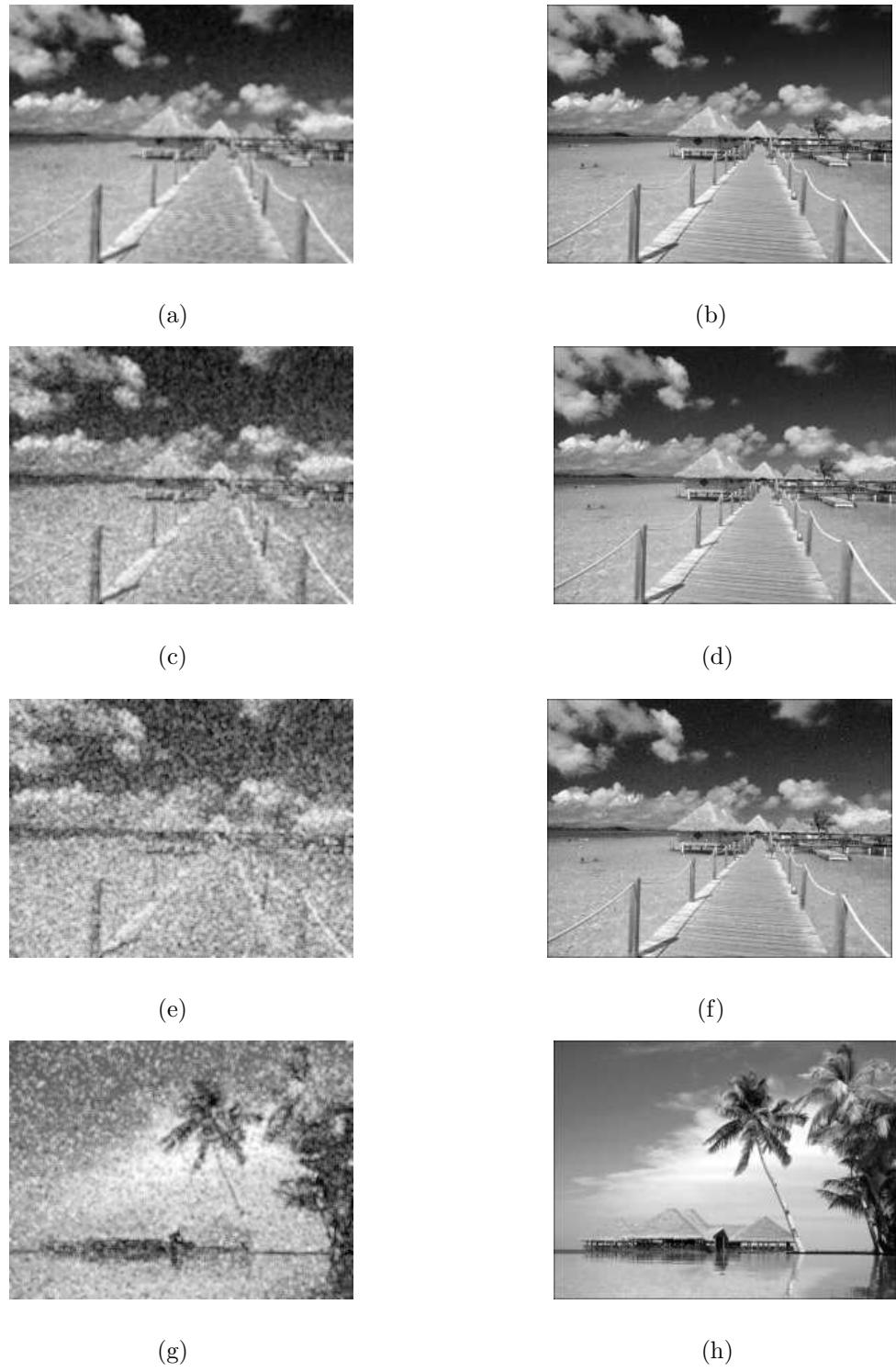


Fig. 5.4: The images filtered with contraharmonic mean filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (g) High Gaussian noise; (h) Salt & pepper noise.

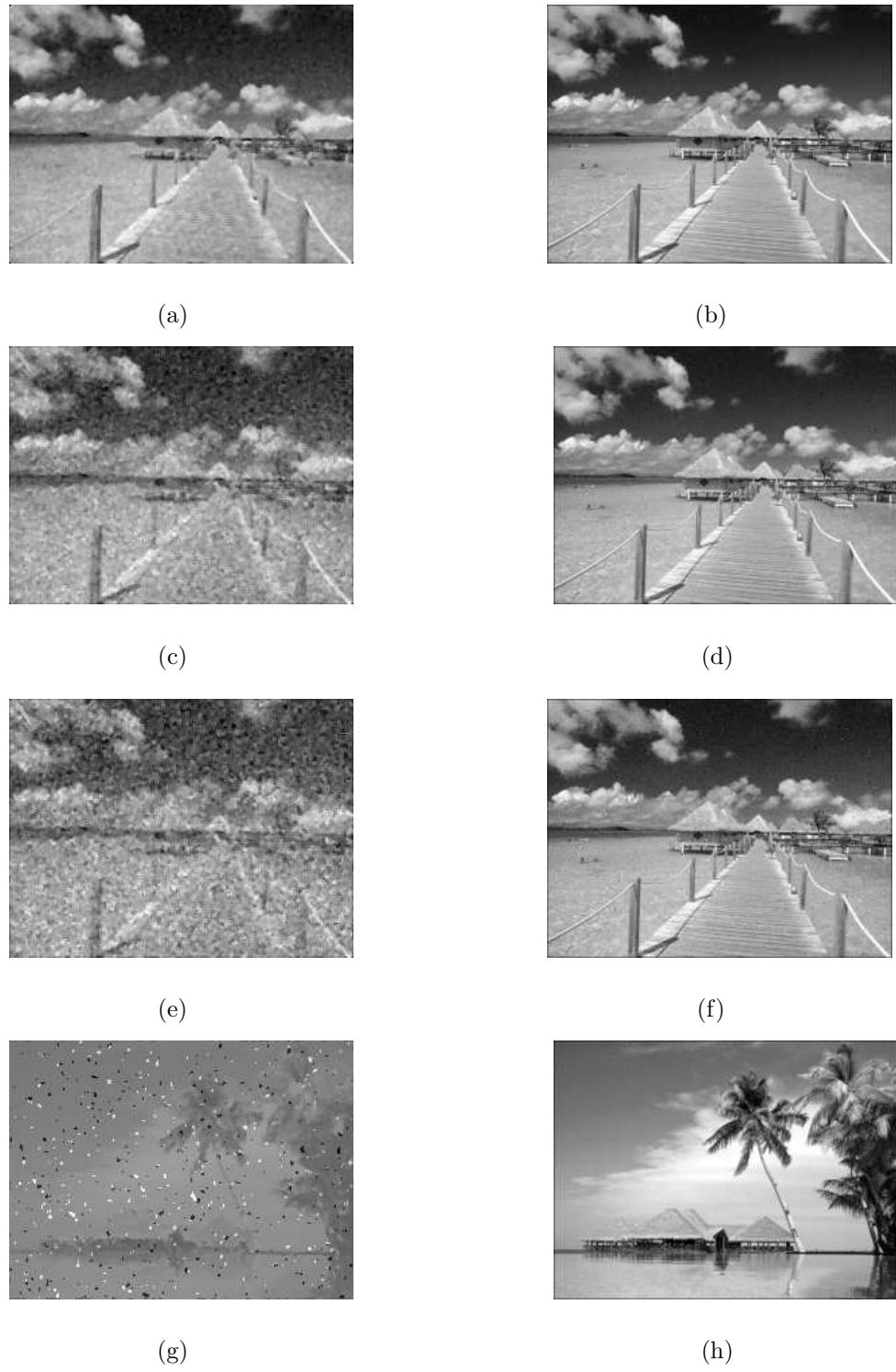


Fig. 5.5: *The images filtered with median filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.*

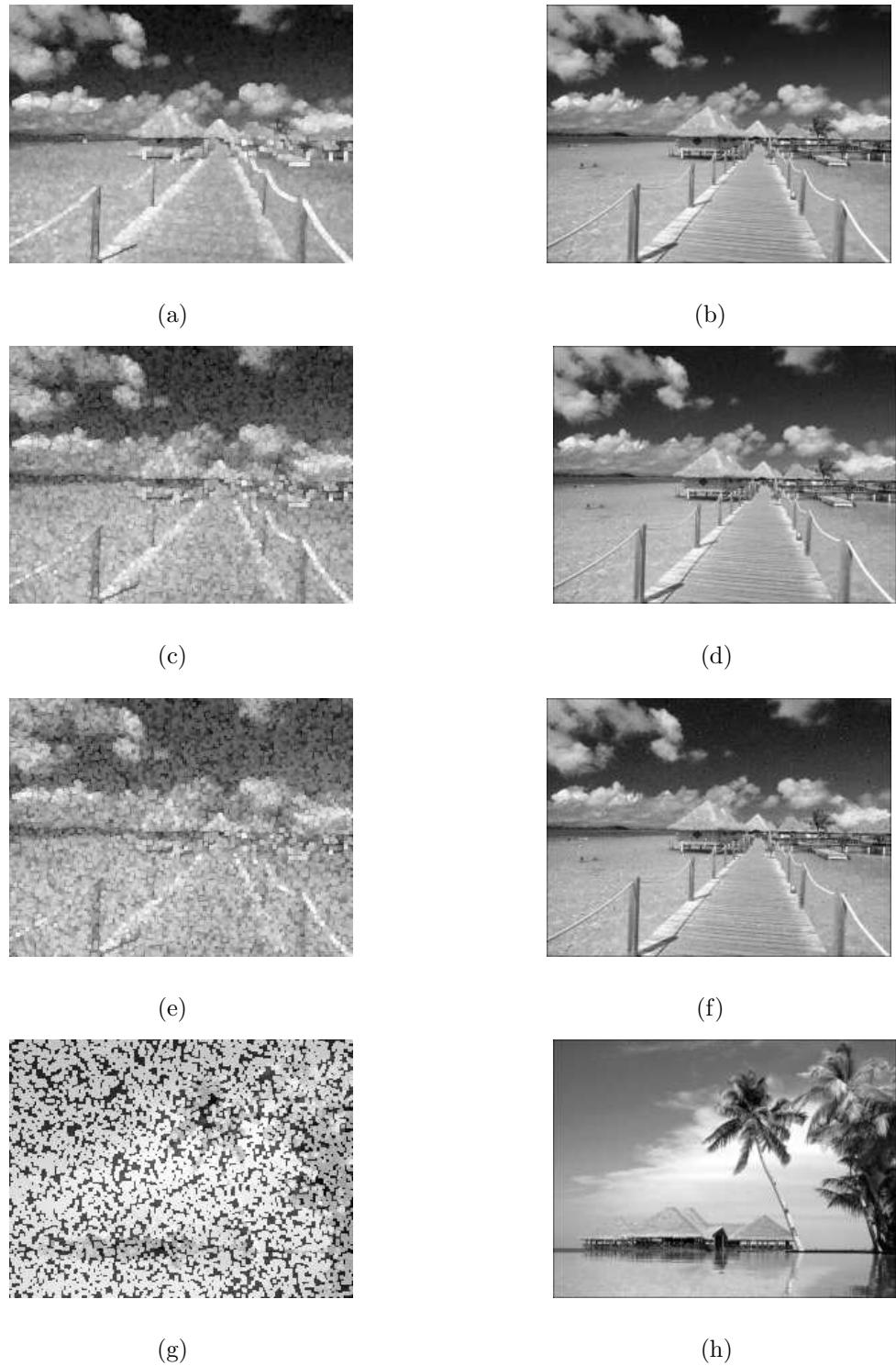


Fig. 5.6: *The images filtered with max filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.*

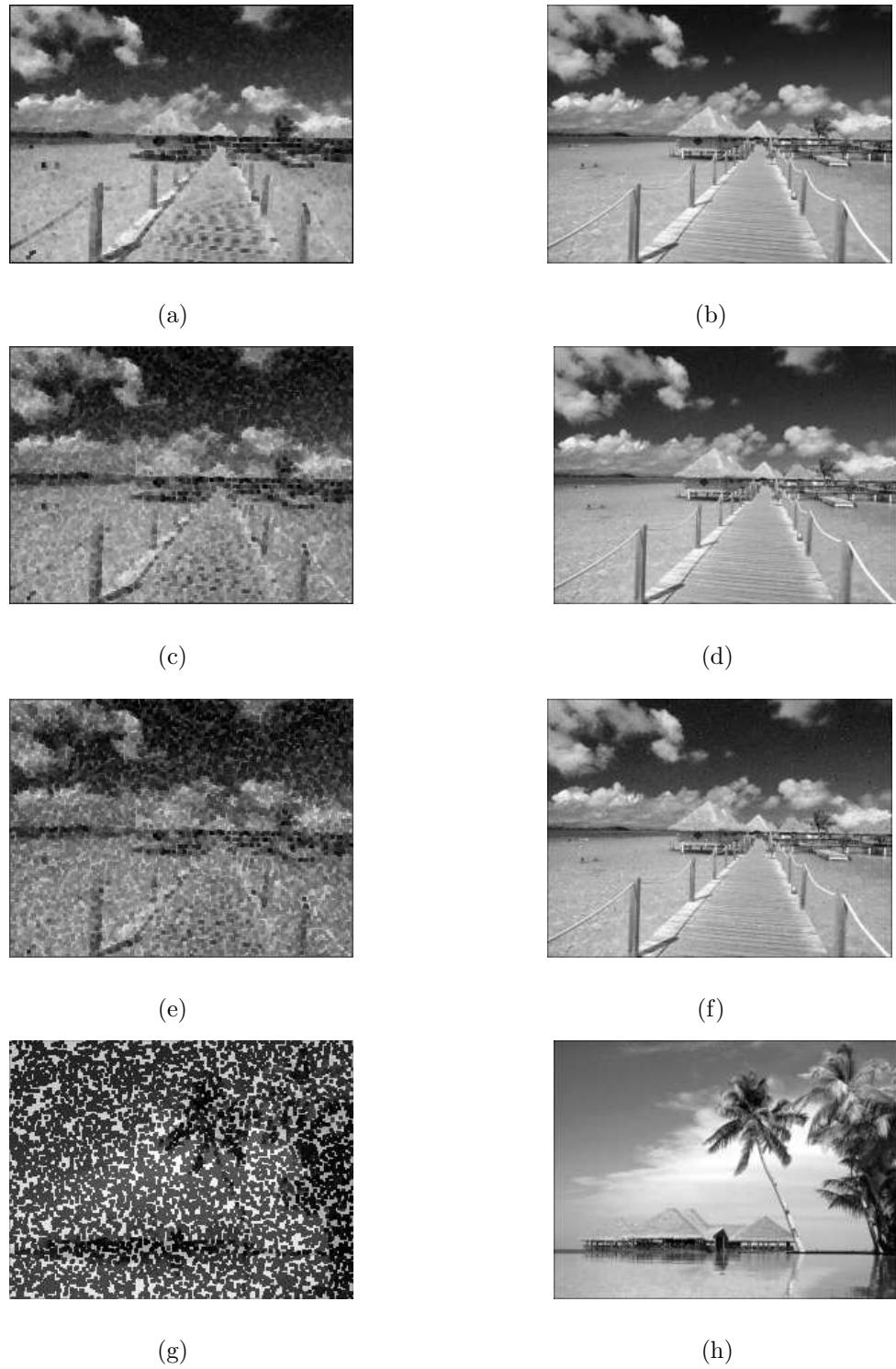


Fig. 5.7: *The images filtered with min filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with min filter corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (g) Salt & pepper noise.*

5.1.8 Midpoint Filter

Now the midpoint filter is used to recover the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with midpoint filter and bell MFs in ANFIS are shown in Fig. 5.8 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We find that the filtered images are much poorer than those images filtered with ANFIS.

5.1.9 Alpha-trimmed Mean Filter

Then we use alpha-trimmed mean filter to explore and restore the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with alpha-trimmed mean filter and with bell MFs in ANFIS are shown in Fig. 5.9 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We still find that the filtered images are much poorer than those in using with ANFIS.

5.1.10 SD-ROM Filter

Finally SD-ROM filter is used to cope with the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with SD-ROM filter and with bell MFs in ANFIS are shown in Fig. 5.10 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. Again we find that the filtered images are much poorer than those filtered with ANFIS.

5.2 Frequency Domain Filtering

Here we design some frequency domain filter with frequency transformation method, frequency sampling method and windowing method and Gaussian lowpass filter and

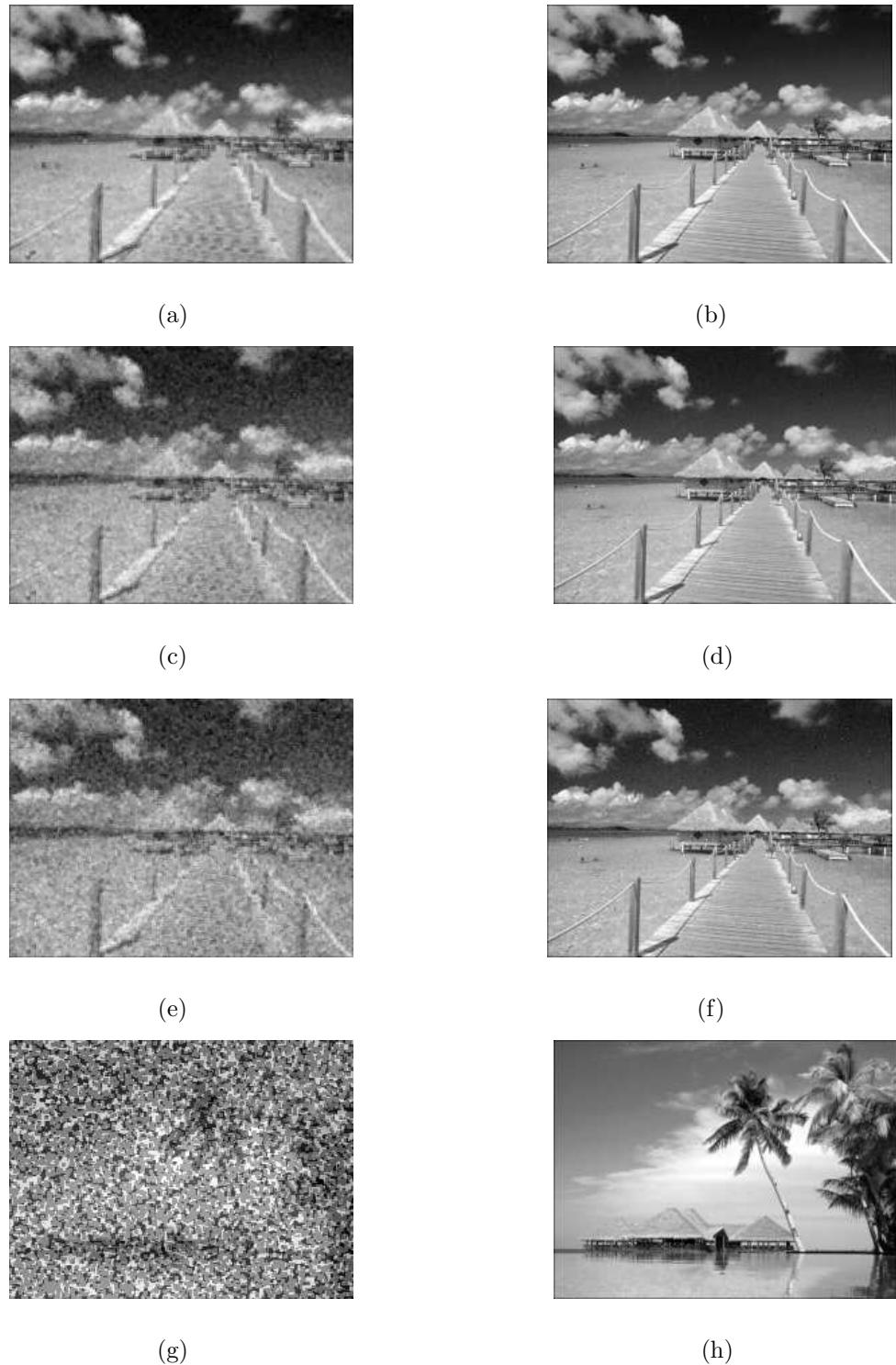


Fig. 5.8: *The images filtered with midpoint filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.*

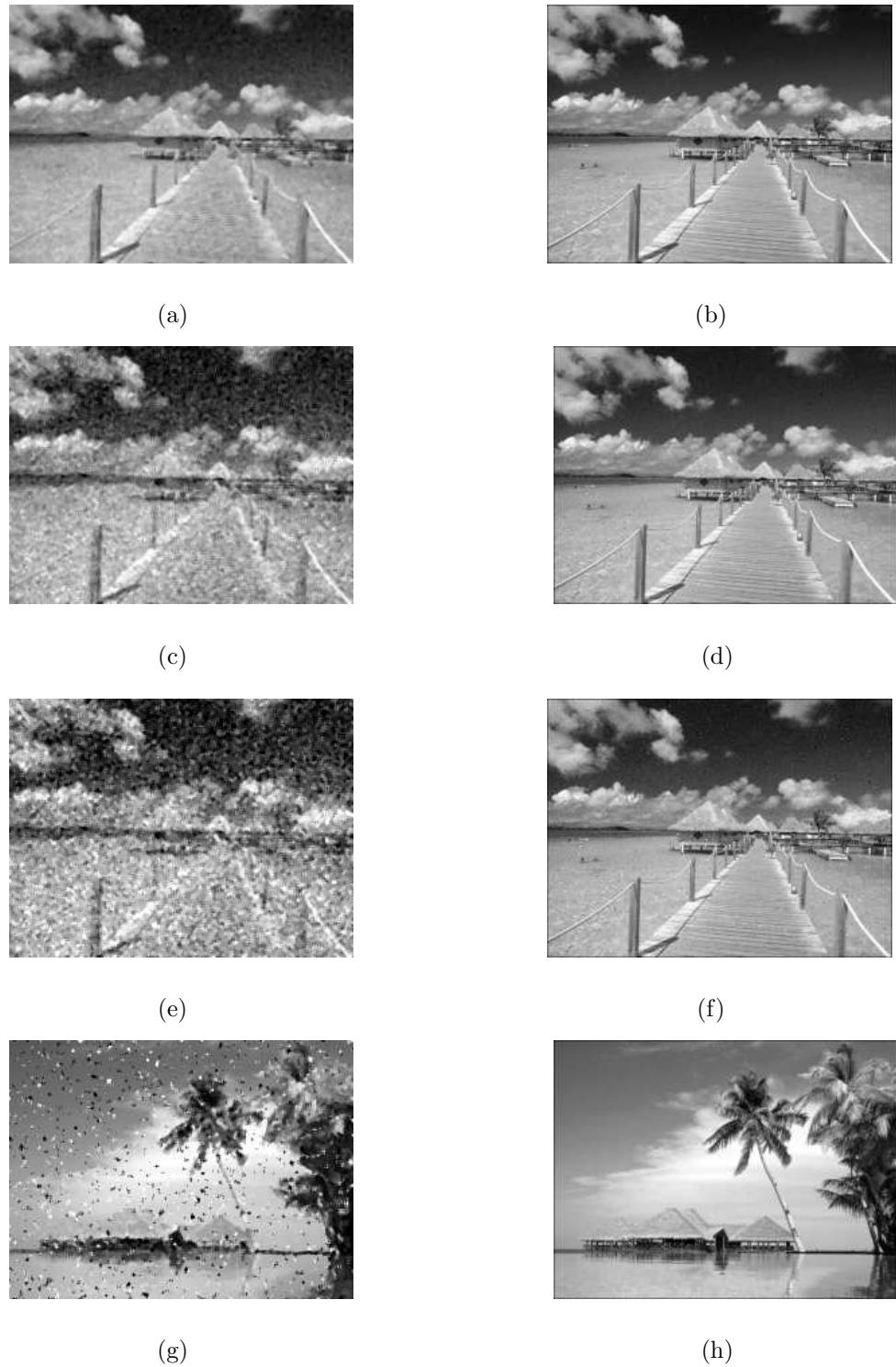


Fig. 5.9: *The images filtered with alpha-trimmed mean filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.*

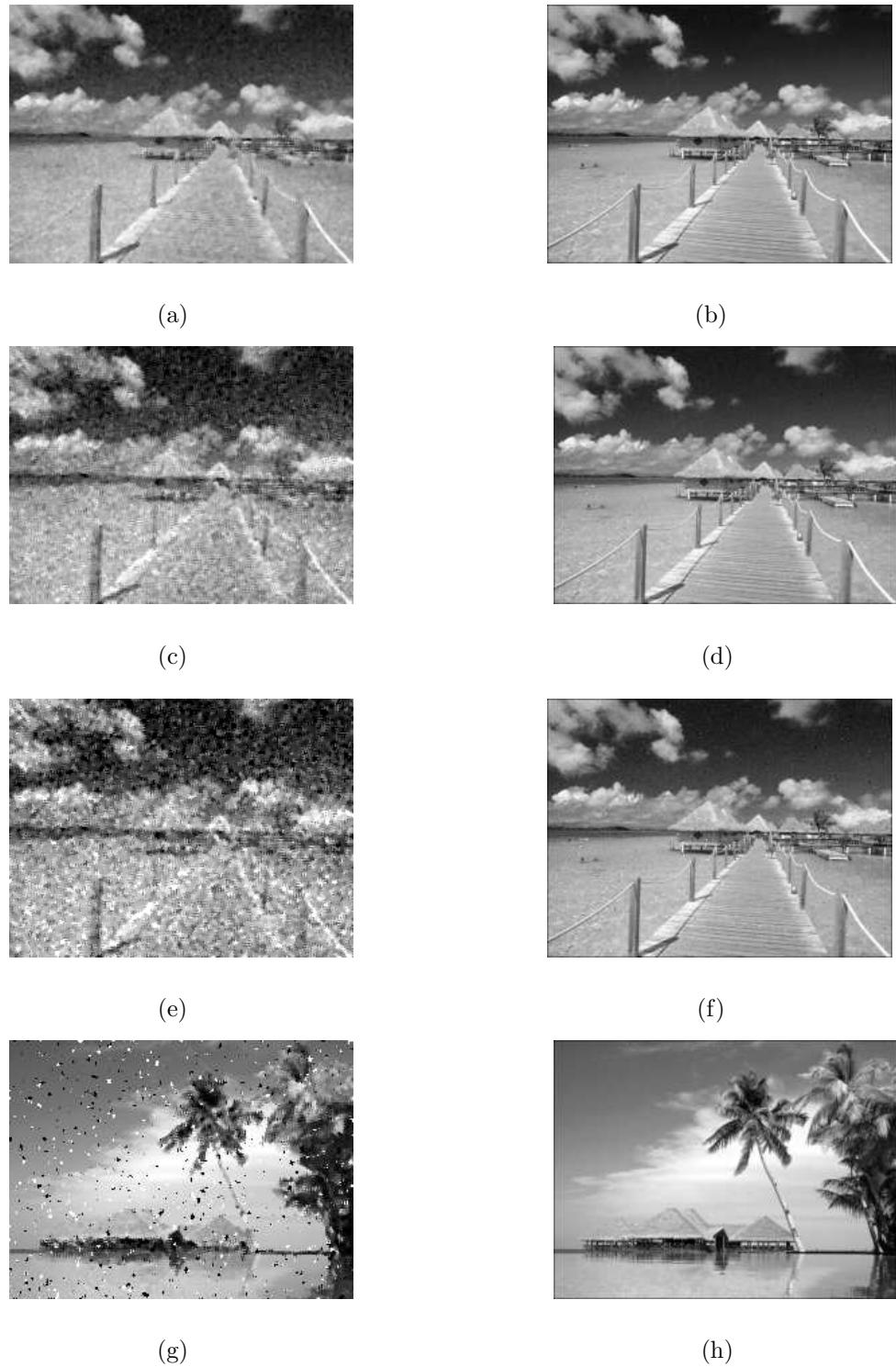


Fig. 5.10: *The images filtered with SD-ROM filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.*

Butterworth lowpass filter. We also compare the results of these filters with those of ANFIS.

5.2.1 Frequency Domain Filter Designed with Frequency Transformation Method

Now we design the tenth order frequency domain filter with frequency transformation method to handle the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with frequency transformation method and with bell MFs in ANFIS are shown in Fig. 5.11 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We find that the filtered images are much poorer than those filtered using ANFIS.

5.2.2 Frequency Domain Filter Designed with Frequency Sampling Method

We design the frequency domain filter with frequency sampling method to handle the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with frequency sampling method and with bell MFs in ANFIS are shown in Fig. 5.12 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We find that the filtered images are much poorer than those filtered with ANFIS.

5.2.3 Frequency Domain Filter Designed with Windowing Method

The frequency domain filter with windowing method is designed to handle the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with windowing method and with bell MFs in ANFIS

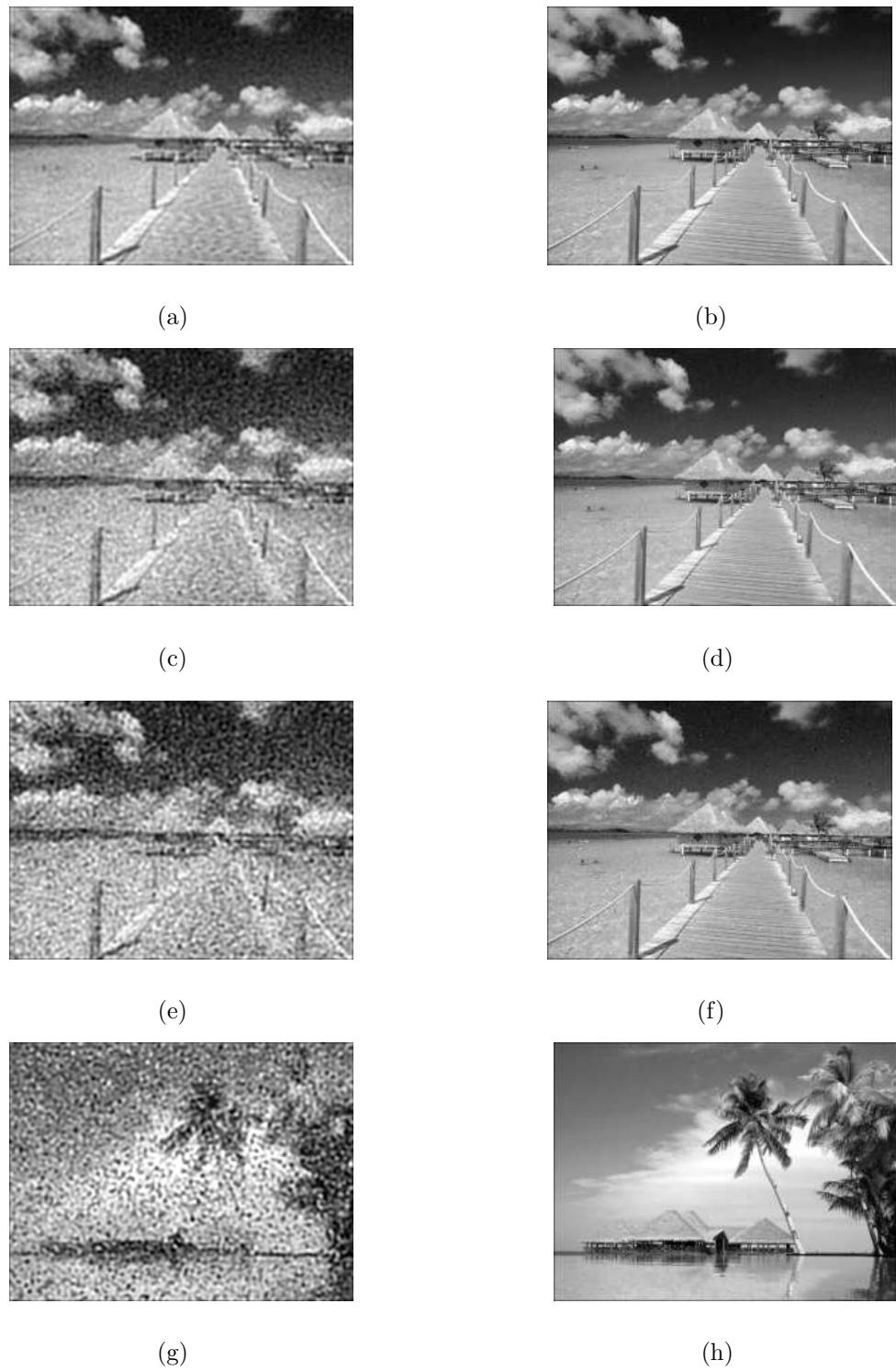


Fig. 5.11: The images filtered with frequency transformation method corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.

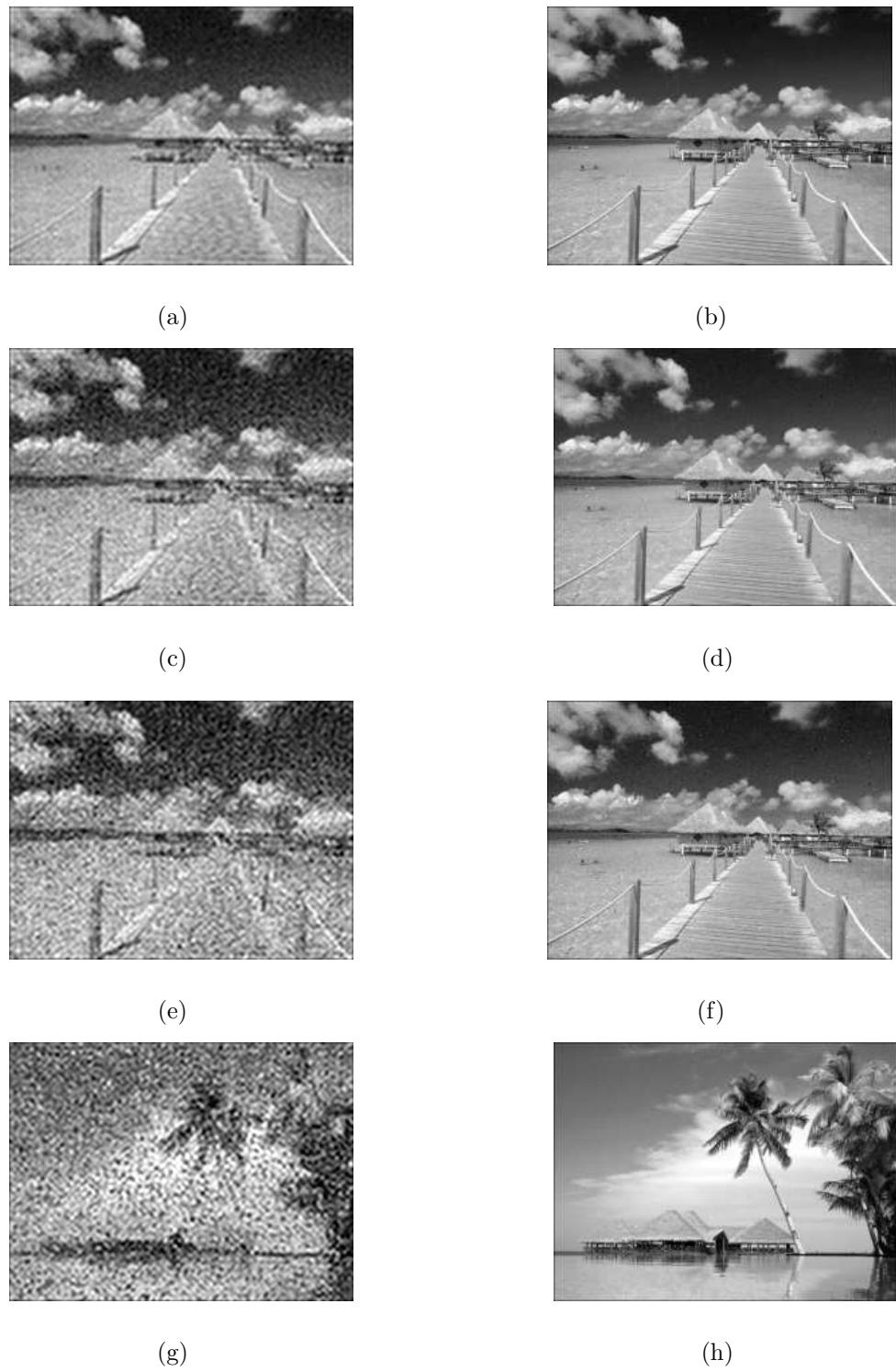


Fig. 5.12: *The images filtered with frequency sampling method corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.*

are shown in Fig. 5.13 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We find that these filtered images are much worse than those filtered with ANFIS.

5.2.4 Gaussian Lowpass Filter

Then we choose the Gaussian lowpass filter to cope with the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with Gaussian lowpass filter and with bell MFs in ANFIS are shown in Fig. 5.14 and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. Again we find that the filtered images are much worse than those filtered with ANFIS.

5.2.5 Butterworth Lowpass Filter

The Butterworth lowpass filter is used to deal with the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with Butterworth lowpass filter and with bell MFs in ANFIS are shown in Fig. 5.15 for comparison and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We find the filtered images are much worse than those filtered with ANFIS.

5.3 Wiener Lowpass Filter

Now we use a Wiener lowpass filter to recover the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The filtered images with Wiener lowpass filter and with bell MFs in ANFIS are shown in Fig. 5.16 and the MSEs of filtered images are listed in Table 5.1 corresponding to the above four types of noise. We find the filtered images are much worse than those filtered with ANFIS.

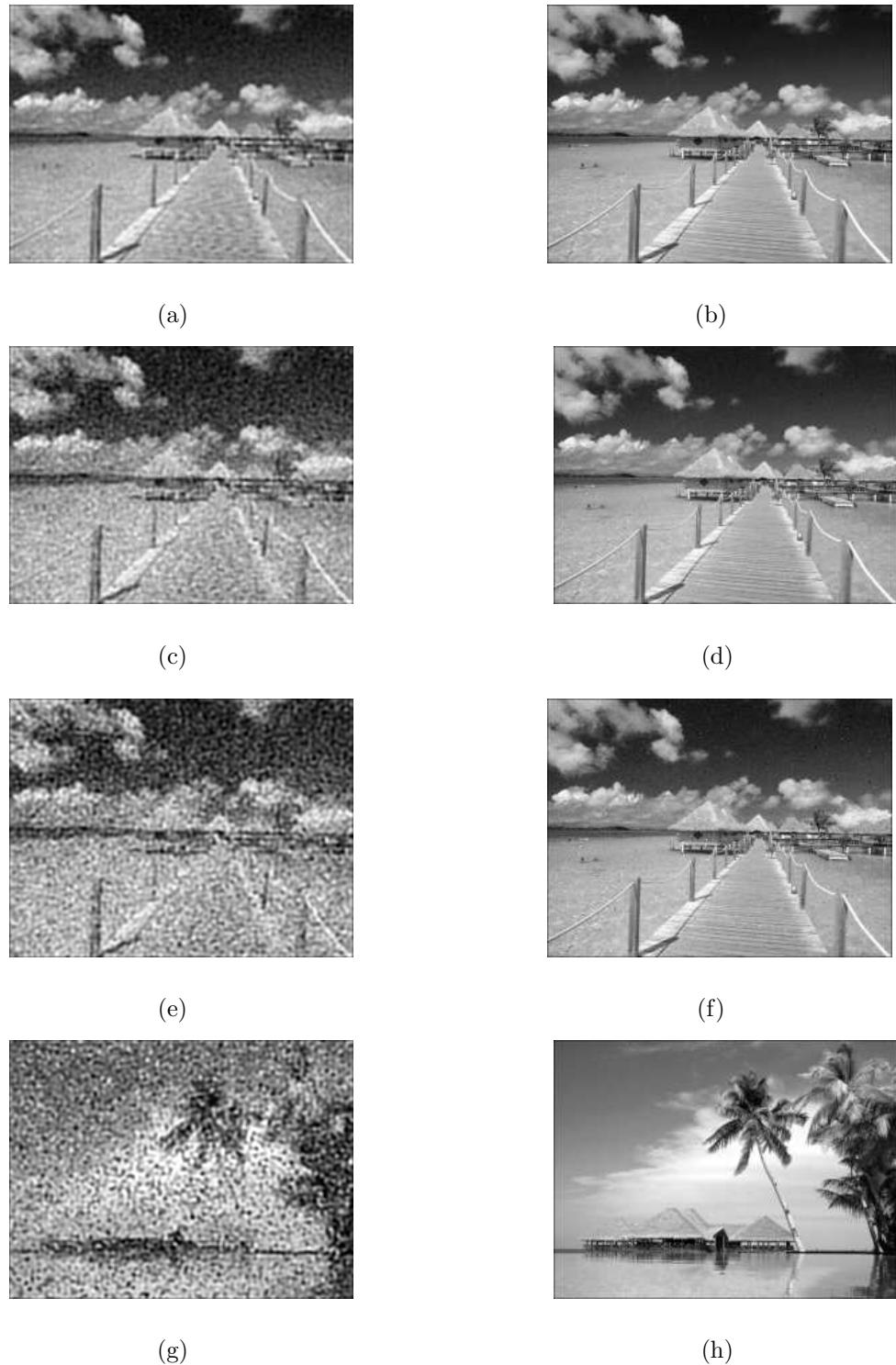


Fig. 5.13: The images filtered with windowing method corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.

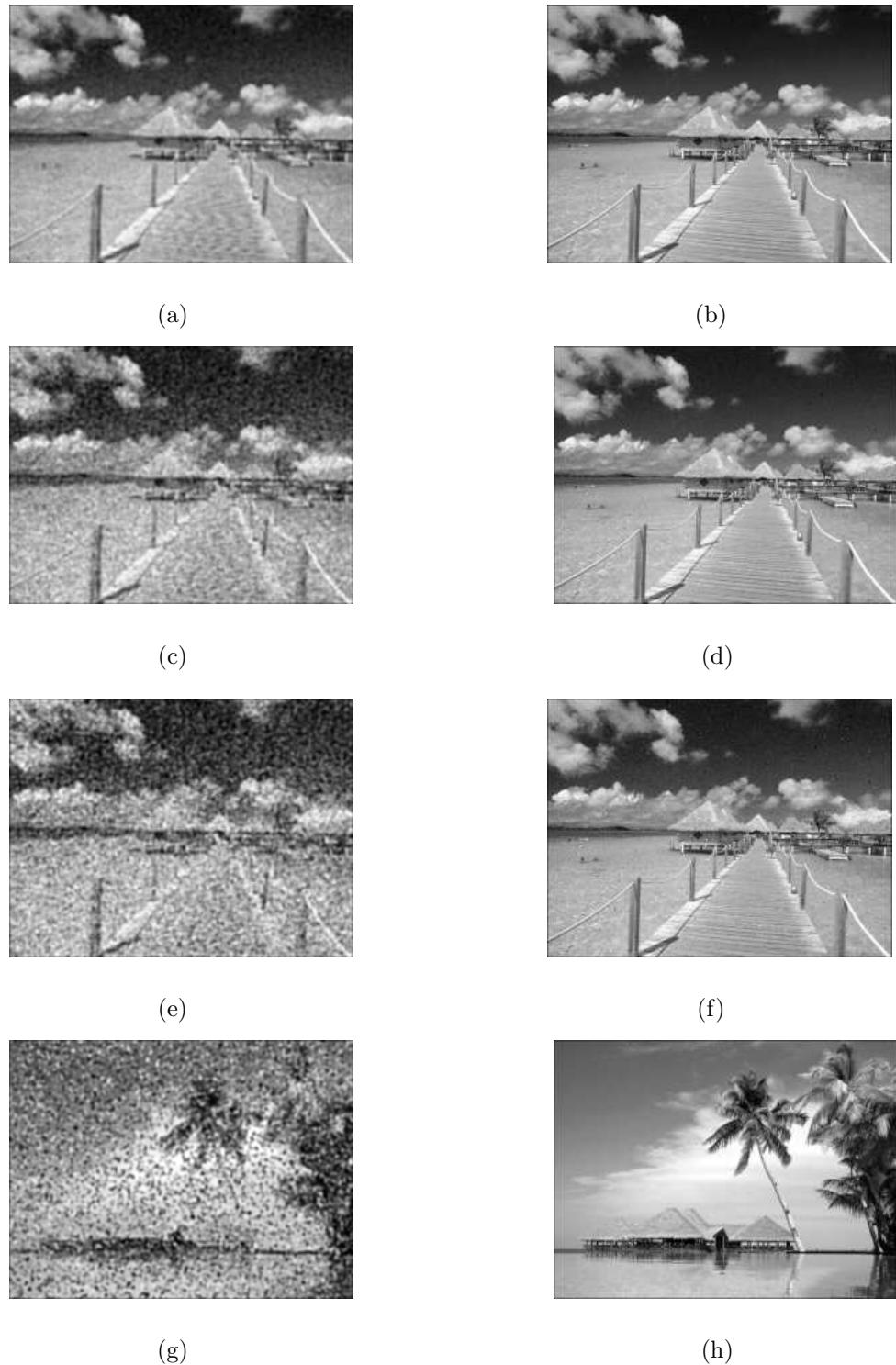


Fig. 5.14: The images filtered with Gaussian lowpass filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.

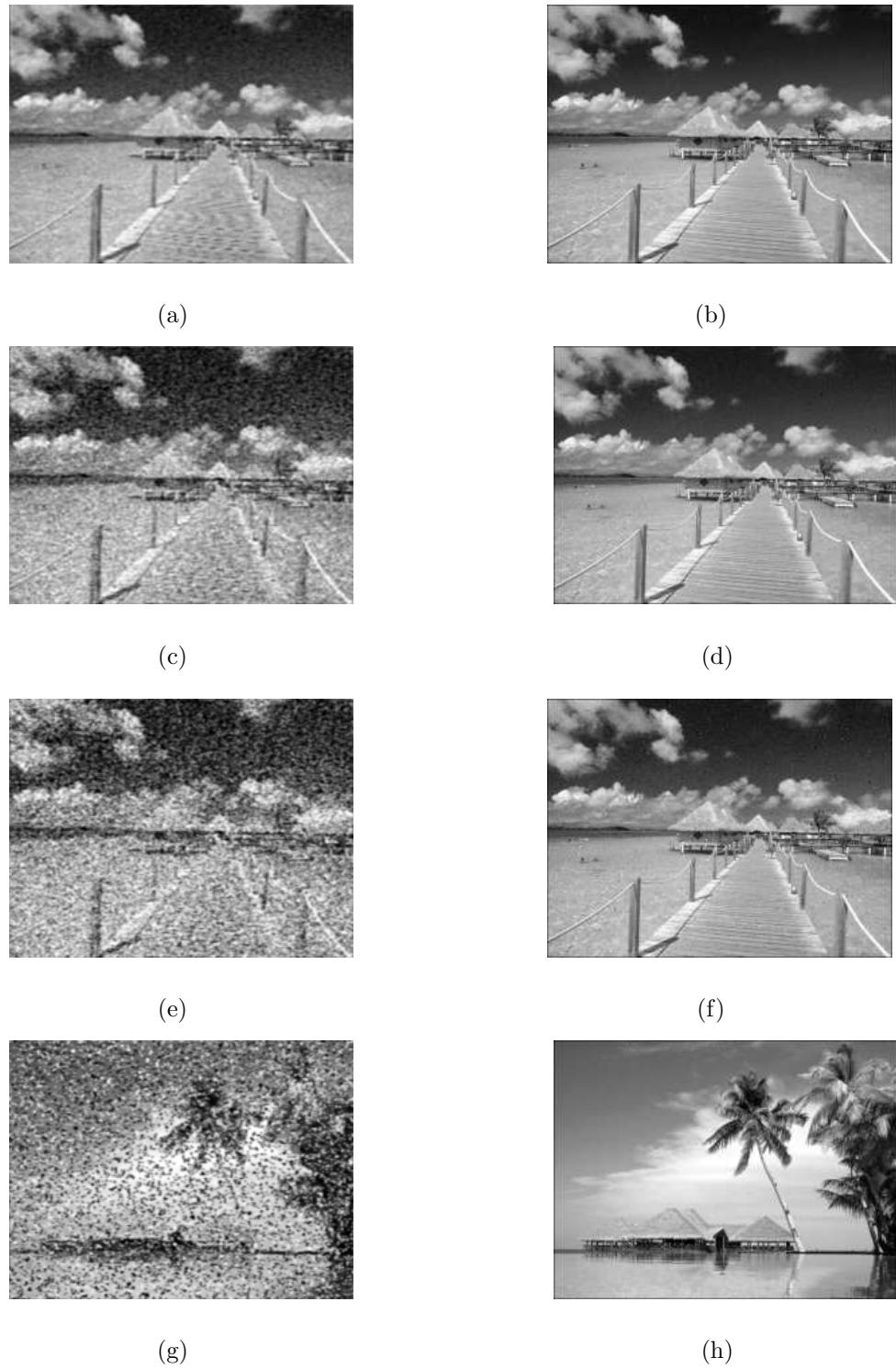


Fig. 5.15: *The images filtered with Butterworth lowpass filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.*

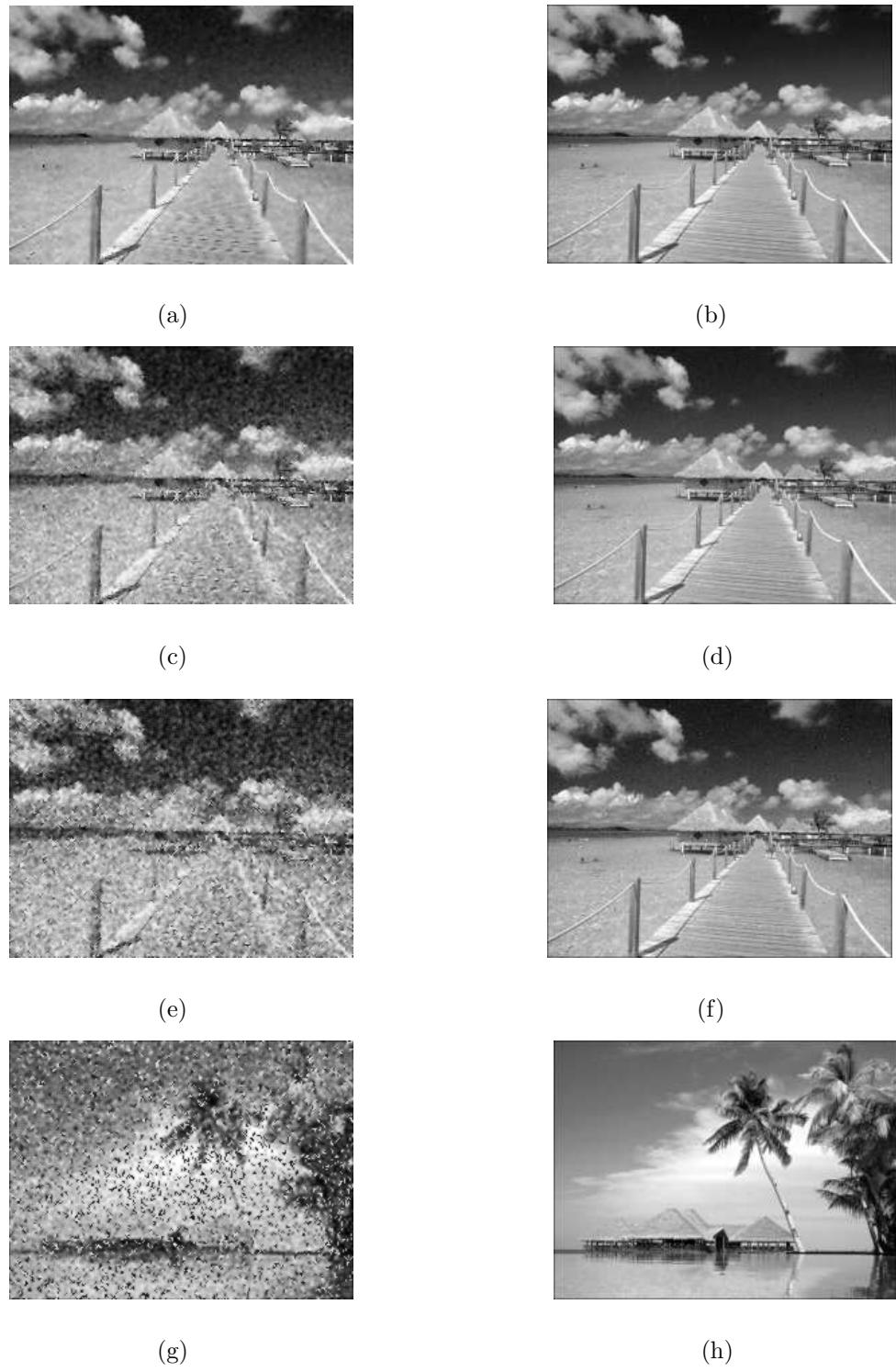


Fig. 5.16: *The images filtered with Wiener lowpass filter corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.*

5.4 Wavelet Denoising

The wavelet is designed to restore the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The wavelet function is chosen as ‘*coiflets*’ and the image is decomposed to 2 level. The denoised images with soft threshold in wavelet and with bell MFs in ANFIS are shown in Fig. 5.17 for comparison and the MSEs of soft and hard threshold are listed in Table 5.1 corresponding to the above four types of noise. Again we find the quality of the filtered images are much poorer than those filtered with ANFIS.

5.5 Wavelet Packet Denoising

Finally we try a wavelet packet to cope with the images contaminated with low, medium and high Gaussian noise and salt & pepper noise separately. The wavelet function is chosen as ‘*syslets*’ and the image is decomposed to 1 level. The denoised images with soft threshold in wavelet packet and with bell MFs in ANFIS are shown in Fig. 5.18 for comparison and the MSEs of soft and hard threshold are listed in Table 5.1 corresponding to the above four types of noise. We find the filtered images are much poorer than those filtered with ANFIS.

5.6 Summary

It is evident that the conventional filtering techniques — spatial filters, optimal Wiener filter, frequency domain filters, wavelet, or wavelet packet — have much poorer results for denoising the images corrupted by different types of noise compared with the results recovered with ANFIS in Table 5.1. The reason is that all conventional filtering techniques blindly recover the images contaminated with noise. However, the image restoration with ANFIS has the ability to learn the knowledge stored in the training data.

Because ANFIS have not only the simplifying function of fuzzy reasoning but also

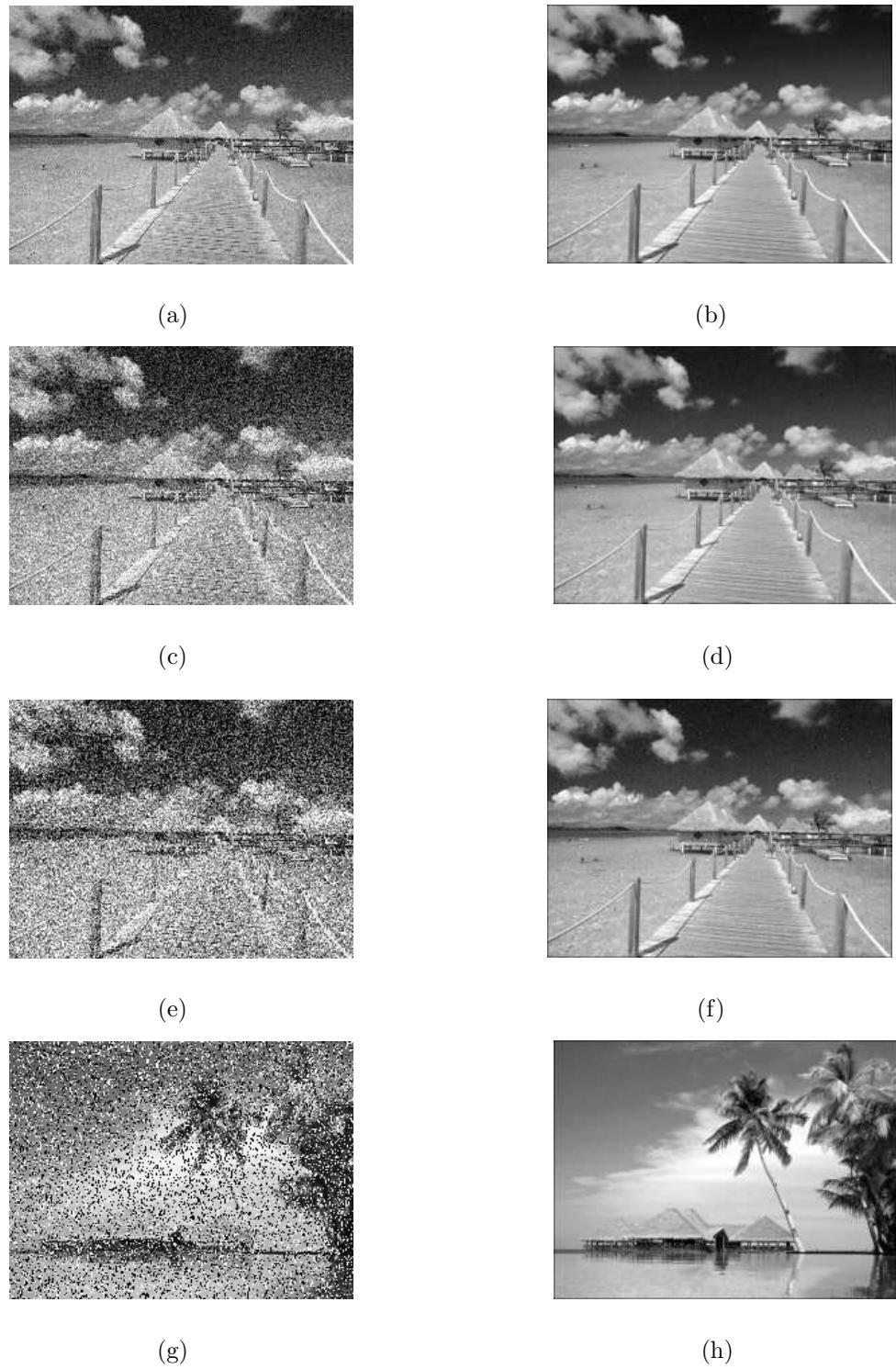


Fig. 5.17: *The images filtered with soft threshold in wavelet corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.*

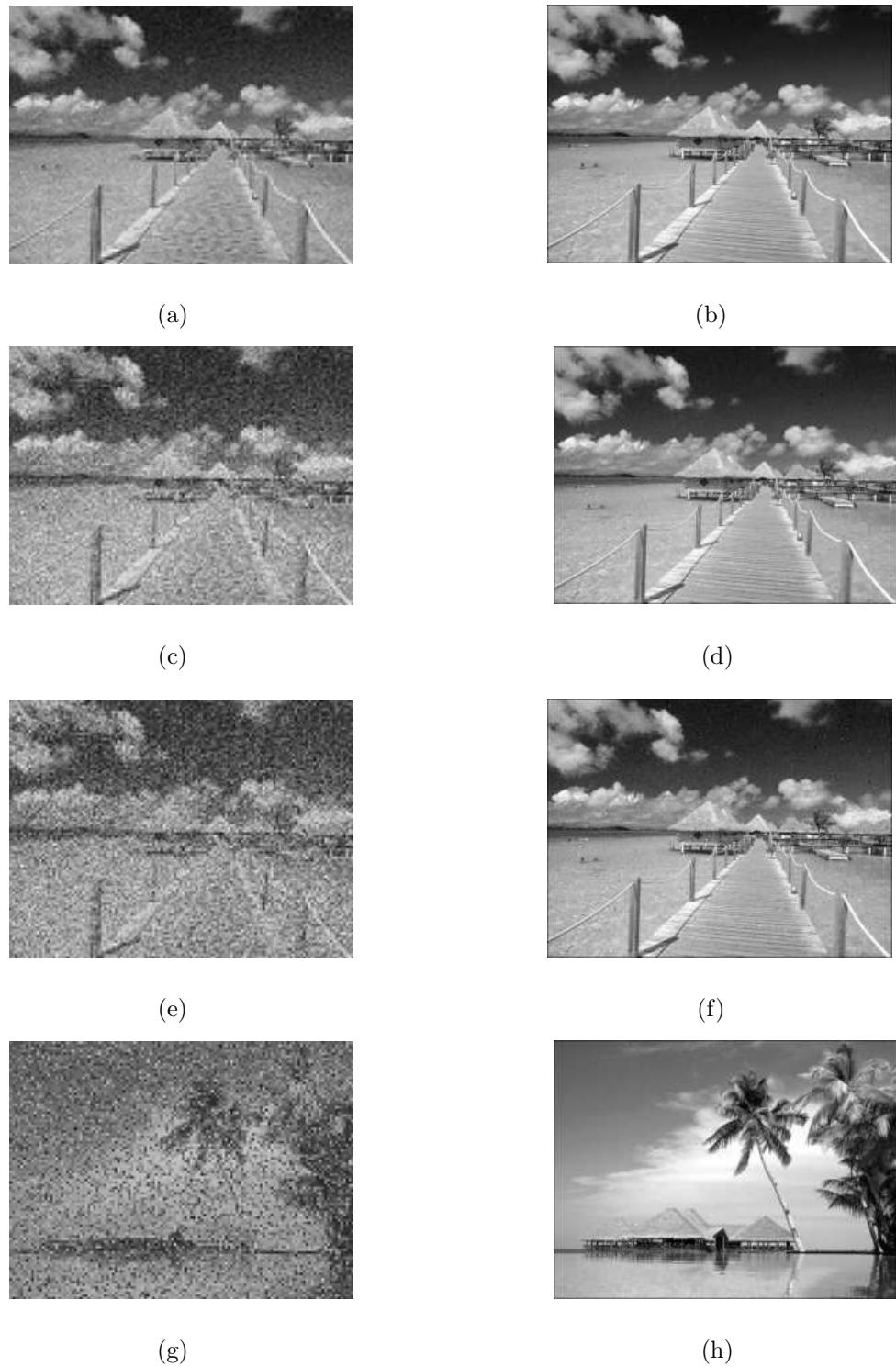


Fig. 5.18: *The images filtered with soft threshold in wavelet packet corrupted by: (a) Low Gaussian noise; (c) Medium Gaussian noise; (e) High Gaussian noise; (g) Salt & pepper noise; and with bell MFs in ANFIS corrupted by. (b) Low Gaussian noise; (d) Medium Gaussian noise; (f) High Gaussian noise; (h) Salt & pepper noise.*

the self-learning ability of neural networks, it has the strong capacity of eliminating pseudo signals (noise). The clarity processed by ANFIS in the above figures is much better than those processed by any traditional filters in this chapter. Compared the data in Table 5.1 with in Table 4.1, they further reveal that the precision of ANFIS is higher two magnitudes than the conventional filtering systems under the standard of MSE. It is adequately proved from the above facts that ANFIS is successfully applied to the area of image processing. Its practical future is very wide in image noise cancellation.

However, it also consumes much more time than conventional filters and wavelet filtering noise. We list the elapsed time of filtered noisy image contaminated with different types of noise by ANFIS (bell MFs) compared with the conventional techniques in Table 5.2. We find CPU spends almost as ten times to filter noise with ANFIS as with the conventional techniques because ANFIS need consume much more time to learn information from each pixel in the image from Table 5.2. Also CPU consumes much more time to remove noise with ANFIS of nonlinear passage dynamics of order 2 than nonlinear passage dynamics of order 3 because the learning parameters increase from 2 to 3 in each pixel.

Table 5.1: *MSE of noisy image contaminated with different types of noise filtered by the conventional techniques.*

Name of filter	MSE			
	Gaussian noise (variance)			Salt & pepper
	Low	Medium	High	(Probability)
	175.3222	1577.9	4383.1	0.3017
Arithmetic mean filter	250.9947	403.2217	708.2673	6665.1
Geometric mean filter	252.0248	1356	3251.6	45462
Harmonic mean filter	275.9819	730.7467	1310.7	1845.5
Contraharmonic mean filter	225.1598	433.5892	903.4906	885.5909
Median filter	226.1985	561.0192	1211	5036.1
Alpha-trimmed mean filter	207.8163	475.0034	951.5577	928.4560
SD-ROM filter	276.5266	608.8525	1203.8	1188
Max filter	528.3509	702.4051	1190.9	34650
Min filter	756.7153	808.9610	1199.2	34145
Midpoint filter	349.2780	440.8521	671.4911	16824
Frequency transformation method	212.5613	440.0593	825.2965	1669.2
Frequency sampling method	240.8762	479.4706	890.5405	1780.7
Windowing method	197.4664	423.3473	816.0861	1678
Gaussian lowpass filter	186.3891	363.2081	681.3593	1444.2
Butterworth lowpass filter	129.0895	385.5263	827.8351	1727.9
Wiener lowpass filter	89.8830	352.7439	801.3708	1971.8
Wavelet with soft threshold	146.7896	1191.5	3027.8	5422.2
Wavelet with hard threshold	169.5155	1470.5	3559.4	6119
Wavelet packet by soft threshold	211.6517	539.3796	1092	2129.7
Wavelet packet by hard threshold	211.6517	539.8328	1093.7	2131

Table 5.2: *The elapsed time of filtered noisy image contaminated with different types of noise by ANFIS (bell MFs) and by the conventional techniques (Unit: second).*

Name of filter	Elapsed time	
	Gaussian noise	Salt & pepper noise
	Variance=4383.1	Probability=0.3017
ANFIS with passage dynamics of order 2	183.5630	173.6890
ANFIS with passage dynamics of order 3	298.0090	293.8130
Arithmetic mean filter	2.1030	2.2130
Geometric mean filter	2.8540	2.7640
Harmonic mean filter	11.2460	7.9910
Contraharmonic mean filter	4.4460	4.6470
Median filter	2.1930	2.1030
Max filter	2.1030	2.1930
Min filter	2.1030	2.3730
Midpoint filter	2.1530	2.2430
Alpha-trimmed mean filter	10.0650	10.2440
SD-ROM filter	11.5360	11.2760
Frequency transformation method	1.4720	1.4620
Frequency sampling method	1.4720	1.5420
Windowing method	1.5020	1.5820
Gaussian lowpass filter	1.5320	1.5930
Butterworth lowpass filter	4.0750	3.2650
Wiener lowpass filter	1.6720	1.6030
Wavelet with soft threshold	2.4240	2.4530
Wavelet with hard threshold	2.4340	2.5330
Wavelet packet with soft threshold	1.9530	1.9030
Wavelet packet with hard threshold	2.0030	1.9820

Chapter 6

Conclusions and Future Works

Adaptive noise cancellation was first presented by Widrow and Glover in 1975 [?] and handled with one dimensional signals. We tried to expand this method to two dimensional signals — images to filter noise. In this chapter, we first summarize the conclusions we found from the results of the above experiments. Then we will comment on the direction that the study of noise cancellation can take in the future.

6.1 Conclusions

In this thesis, we used ANFIS as adaptive noise cancellation to filter out noise and restored the images contaminated with different types of noise. The nonlinear passive dynamics of order 2 and 3 were discussed because these two cases are typical in the corrupted images. For higher order components, their effect dramatically reduce with the increasing of delay time in the whole noise. Therefore, we do not discuss the nonlinear passive dynamics of order higher than 3.

Gaussian noise and salt & pepper noise were added to the images and then removed with ANFIS. We found that the restoration effect for removing salt & pepper noise was much better than that for cancelling Gaussian noise. This is because salt & pepper noise is impulse noise and has a distinct identification that its values are equal to the maximum, 255 for salt noise or equal to the minimum, 0 for pepper noise in 8 binary gray images. This distinguishing mark of salt & pepper makes it easy for ANFIS to learn the knowledge and remove this kind of noise from the corrupted

images. On the other hand, the values of Gaussian noise are between 0 and 255 which makes it difficult for ANFIS to distinguish noise from the signal. Therefore, this is the reason why ANFIS need learn more knowledge and also make much more mistakes in removing Gaussian noise from the contaminated images compared with that for salt & pepper noise.

Eight different MFs are used to remove noise and restore the image. They are bell MFs, triangle MFs, Gaussian MFs, trapezoidal MFs, two-sided Gaussian MFs, product of two sigmoidal MFs, difference between two sigmoidal MFs, and pi-shaped MFs. In removing salt & pepper noise, all types of MFs display the excellent abilities and the differences of these MFs are very small reflected in the MSEs of filtered images because of the distinct property of this noise. In removing Gaussian noise, the differences of these MFs' abilities are quite large. Pi-shaped MFs, two-sided Gaussian MFs, product of two sigmoidal MFs, and difference between two sigmoidal MFs all show their excellent capacities of filtering different type of noises no matter what the intensity of noise is. However, the capability of triangle MFs is the poorest for cancelling Gaussian noise.

We also discuss the other parameters in ANFIS. In the case of salt & pepper noise, only when choosing the different optimization methods, the difference is particularly big. The effect of choosing the combination of least-squares and backpropagation gradient descent methods is much better than that of the backpropagation method. The choices of other parameters such as the training epochs, the MF number for each input, the output MF type are not important. However, on the other hand, the choice for these parameters all have an important effect on the results of removing Gaussian noise. The conclusions we reach all obtained by removing Gaussian noise with ANFIS as follows:

- With the increment of the training epoch, the result for cancelling noise is better and better. After 100 epochs, the curve of RMSE tend to stable in a value. Even though it still reduces noise a little better and better after 100 epoches, the effect of removing noise is not much. However, on the other hand, it consumes much more time. Considering from saving time, choosing the training epoch from 50

to 100 is better than that out of this area.

- The effect of choosing the combination of least-squares and backpropagation gradient descent methods is much better than that choosing the backpropagation method.
- The *linear* output MF remove Gaussian noise much better than the *constant* output MF.
- The step size of training data for all MF in ANFIS is related to RMSE. If RMSE decreases, the step size increases. However, if RMSE oscillates, the step size decreases.
- We do not find overfitting in ANFIS for cancelling noise no matter which type of noise is in the nonlinear passive dynamics of order 2 and 3 because the case do not occur when the error of training data decreases and the error checking data increases at the meantime.
- The result from the number of MFs for each input reflects the complexity of ANFIS for choosing parameters. For example in the nonlinear passive dynamics of order 2 corrupted by Gaussian noise, when the number of MFs for each input is 2 or 3, it is not enough to reflect the complex structure of data, therefore the MSE of filtered image is big. However, when we choose the number of MFs for each input as 5 or 6, it produces the redundancy for the structure of data, therefore the MSE slightly increases. If we choose the number of MFs for each input as 4, it is the best to reflect the complex structure of data. It is important that the structure of the system must match the data. Therefore, to build the model with ANFIS, the choice of data must express the whole properties of the system. The choice for the system structure should have enough parameters to reflect all the characteristics. However, the numbers of parameters should also be restrained. It is by no means true that the more complex the structure, the better the effect. The structure should match with the data. We need to decide

the structure according to the experience or by changing it to observe the effect in special application.

Finally the results by ANFIS for cancelling noise were compared with those of conventional filters and wavelet. We listed the filtered results with spatial filters in arithmetic mean filter, geometric mean filter, harmonic mean filter, contraharmonic mean filter, median filter, alpha-trimmed mean filter, SD-ROM filter, max filter and min filter. Then, the results were also listed with frequency domain filter designed by frequency transformation method, frequency sampling method, and windowing method, Gaussian lowpass filter, and Butterworth lowpass filter. We also compared the denoising results of optimization Wiener lowpass filter, the wavelet with hard threshold and soft threshold and wavelet packet. These conventional filters and wavelet can not absolutely compare with the excellent ability of ANFIS as adaptive noise cancellation.

Because ANFIS have not only the simplifying function of fuzzy reasoning but also the self-learning ability of neural networks, it has the strong capacity of eliminating pseudo signals (noise). The clarity processed by ANFIS in the above figures is much better than those processed by any traditional filters in Chapter 4. Compared the data in Table 5.1 with in Table 4.1, they further reveal that the precision of ANFIS is higher two magnitudes than the conventional filtering systems under the standard of MSE. It is adequately proved from the above facts that ANFIS is successfully applied to the area of image processing. Its practical future is very wide in image noise cancellation.

Of course, it is not to say that ANFIS can replace the conventional method filtering noise. Unlike conventional filters and wavelet filtering noise in full blind condition, ANFIS needs to know more messages, such as the source noise and guessing the passive dynamics.

It also consumes much more time than conventional filters and wavelet filtering techniques. We find CPU spends almost as ten times to filter noise with ANFIS as with the conventional techniques because ANFIS need consume much more time to learn information from each pixel in the image from Table 5.2. Also CPU consumes

much more time to remove noise with ANFIS of nonlinear passage dynamics of order 2 than nonlinear passage dynamics of order 3 because the learning parameters increase from 2 to 3 in each pixel.

ANFIS excellent capacity of cancelling noise happens to show its excellent ability of intelligent learning and the ability to obtain more knowledge from input data than conventional methods of filtering noise.

6.2 Further Work

Undoubtedly, there exists several aspects of work to explore in the future. We continue the study of this area of science in several ways.

First, beside the experiment of gray image, we also wish to remove noise and restore the RGB (red, green and blue) color image corrupted with the different type of noise at different intensity.

Secondly we can design some adaptive filters to cope with images corrupted by noise. We will derive the two dimensional counterparts of one dimensional adaptive algorithms and then use them in image processing applications to remove noise. The two dimensional versions of the least mean square (LMS) algorithm and an alternative of the recursive least square (RLS) algorithm, Euclidean direction search algorithm (EDS) will be derived. The LMS algorithm was first reported by Hadhoud and Thomas [?] who formally derived the algorithm from one dimensional and applied to two dimensional line enhancement. The results of the LMS and EDS algorithm can be compared with those of ANFIS.

Thirdly, in above research, we assume $h(x, y)$ is equal to 0 in Eq. 2.10 of image degradation model and then use ANFIS to restore the degraded images. It means the images only degraded by noise. However, usually, $h(x, y)$ is not equal to 0 in the image degradation, such as image blurring in Fig. 2.4 because of motion. We can explore what ANFIS can do in the image restoration and how it can learn and grasp knowledge when $h(x, y)$ is not equal to 0. We can discuss the effect of different parameters of ANFIS, such as the types of MFs, the training epochs, the MF number

for each input, the optimization method, the type of output MFs, and also checking if there exist overfitting in ANFIS. Then the results will compared with the conventional restoration methods. For example, We can compare the results deblurred by ANFIS with the conventional deblurring methods, such as deblurring with the Wiener filter, a regularized filter, the Lucy-Richardson algorithm, and the blind deconvolution algorithm.

References

Appendix A

A.1 Application of ANFIS to a Nonlinear Passage Dynamics of Order 3 to Restore an Image Corrupted by Gaussian Noise

In the experiment, the unknown nonlinear passage dynamics of order 3 instead of the unknown nonlinear passage dynamics of order 2 in the first experiment were assumed to be defined as

$$\begin{aligned} v_0(i) &= f(v_1(i), v_1(i-1)) \\ &= \frac{8 \sin(v_1(i)) v_1(i-1) v_1(i-2)}{1 + (v_1(i-1))^2 + (v_1(i-2))^2} \end{aligned} \quad (\text{A.1})$$

where $v_1(i)$ is a noise source and $v_1(i-1)$ and $v_1(i-2)$ is the one unit delay and two unit delay of the noise source, $v_0(i)$ is defined as the resultant of the nonlinear passage dynamics $f(\cdot)$ owing to $v_1(i)$, $v_1(i-1)$ and $v_1(i-2)$, i is from 1 to the number of the pixels in the image.

We choose that SNRs are equal to 12.2625, 2.7201 and -1.7169 for the intensity of low, medium and high Gaussian noise respectively.

Here we still use the RGB image ‘*Matirapoint*’ for comparison. All settings are the same as in a nonlinear passage dynamics of order 2 and then transfer to the gray image for experiments except the change of the ANFIS input variables from 2 to 3.

A.1.1 Application of ANFIS with Different MFs

In this section, we discuss the effect of different MFs of input for filtering the image contaminated with low, medium and high noise. The names of these MFs are bell MF, Gaussian MF, two-sided Gaussian MF, triangle MF, trapezoidal MF, product of two sigmoid MFs, difference between two sigmoidal MFs, pi-shaped MF.

The Restoration with the Bell Membership Function

We first investigate the behavior of these signals in the frequency domain before we go on to the next. Fig. A.1(a) to A.1(d) display the spectral density distributions of $s(i)$, $v_1(i)$, $v_0(i)$ and $x(i)$, respectively, from the first 256 points. Like the first experiment, the spectra of the information signal $s(i)$ and the distorted noise $v_0(i)$ overlap each other in a large frequency area. So it is impossible to apply common frequency domain filtering methods to remove $v_0(i)$ from $x(i)$.

Because the differences of the MFs in the restoration processing are almost the same to low, medium and high noise, we just show the changes of the bell MFs for high noise in Fig A.2 before and after training. Compared the nonlinear passage dynamics of order 3 with that of order 2, the only difference is the number of the MFs for the input changing from 4 to 6.

The images corrupted by the intensity of low, medium and high Gaussian noise and the results of removing noise with ANFIS are shown in Fig A.3(a) through A.3(f) respectively. The result is that the restoration effect is good compared with the original image in Fig. 4.1(b) no matter how heavy noise the image is corrupted.

The Restoration with the Triangle Membership Function

We choose another MFs, the triangle membership function to remove the noise and restore the image in the nonlinear passage dynamics of order 3. We first show the MFs to high noise before and after training in Fig A.4.

Then the images corrupted by the intensity of low, medium and high Gaussian noise and the results of removing noise with ANFIS are shown in Fig A.3(a) through

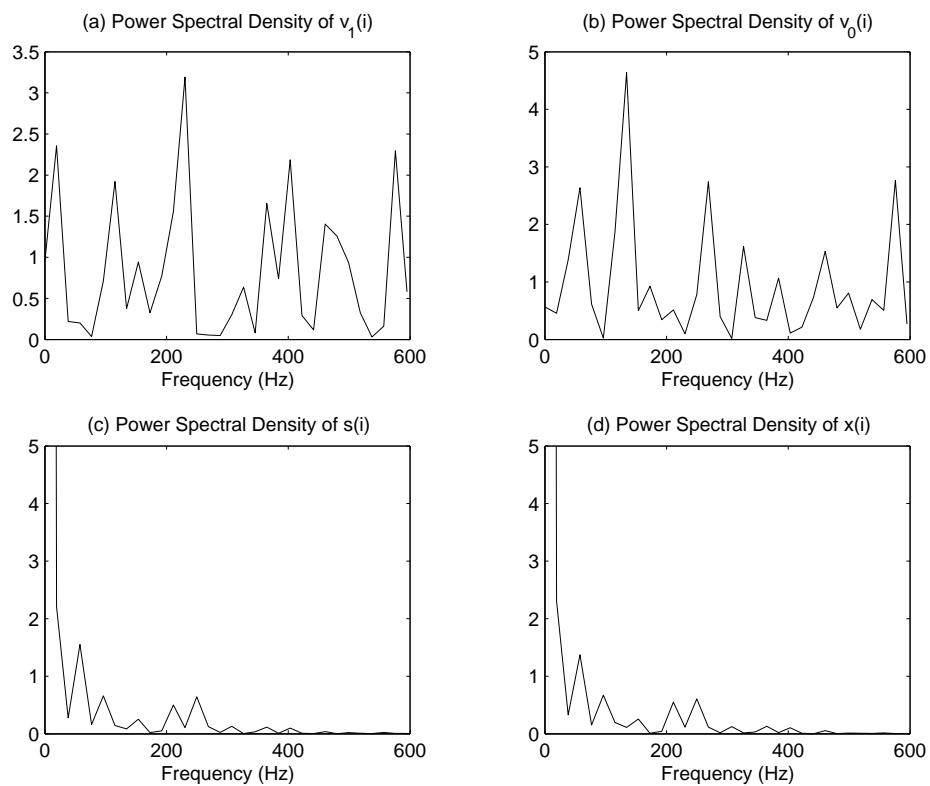


Fig. A.1: Spectral density distributions. (a) Information signal $s(i)$; (b) Source noise signal $v_1(i)$; (c) Distorted noise signal $v_0(i)$; (d) Measurable output signal $x(i)$.

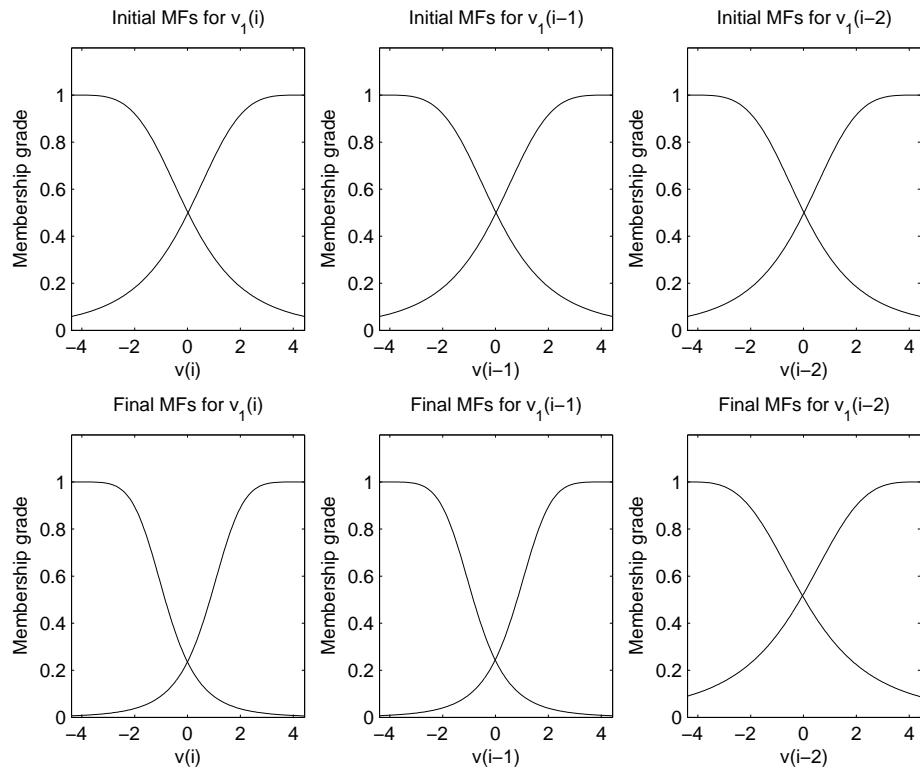


Fig. A.2: The changes of bell MFs before and after training.

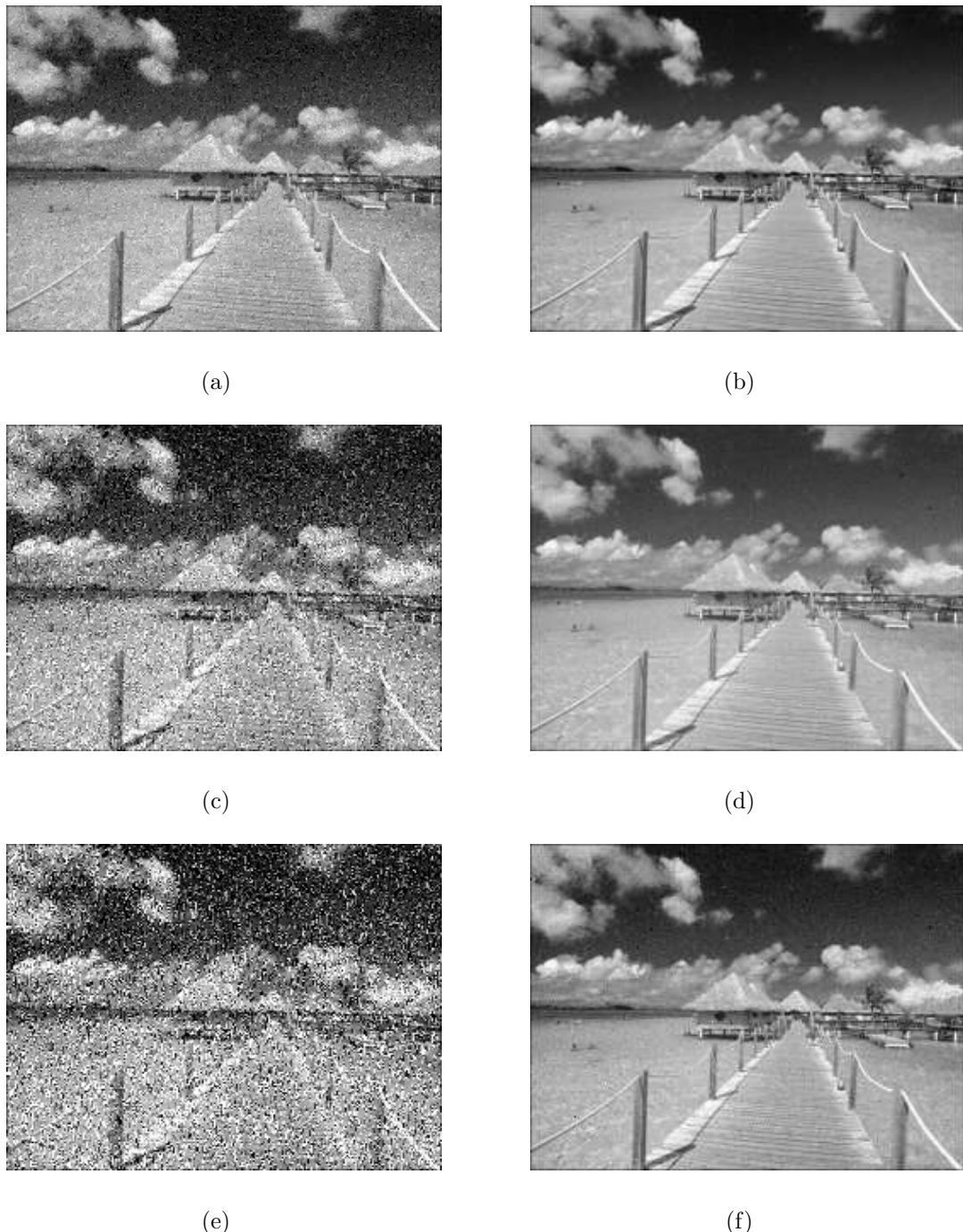


Fig. A.3: The images corrupted by the different intensity noise and the results of removing noise with ANFIS of bell MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

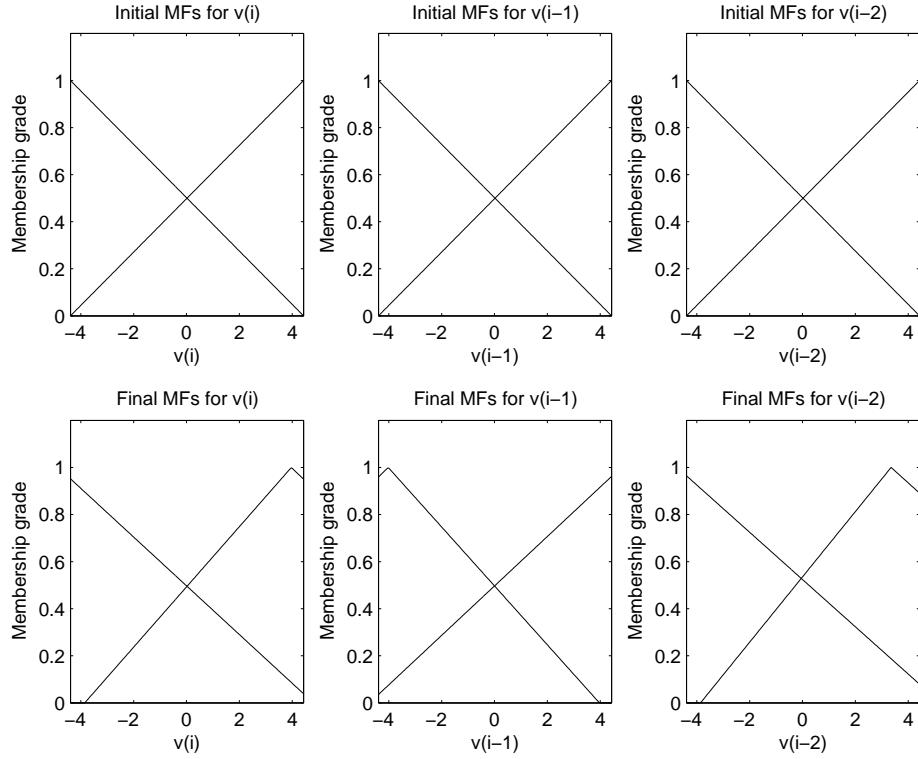


Fig. A.4: The changes of triangle MFs before and after training.

A.3(f) respectively.

The Restoration under the Gaussian Membership Function

Now we use the Gaussian membership function to remove the noise and compare the results of the image restoration in the nonlinear passage dynamics of order 3. The MFs to high noise before and after training for high noise are shown in Fig A.6.

We show the image corrupted by low, medium and high noise and the filtered results in Fig A.7(a) to A.7(f) respectively.

The Restoration with the Two-sided Gaussian Membership Function

The two-sided Gaussian membership function is used to remove the noise and the results are compare with other MFs for cancelling the noise for the nonlinear passage dynamics of order 3. Fig A.8 shows the Two-sided Gaussian MFs to high noise before and after training.

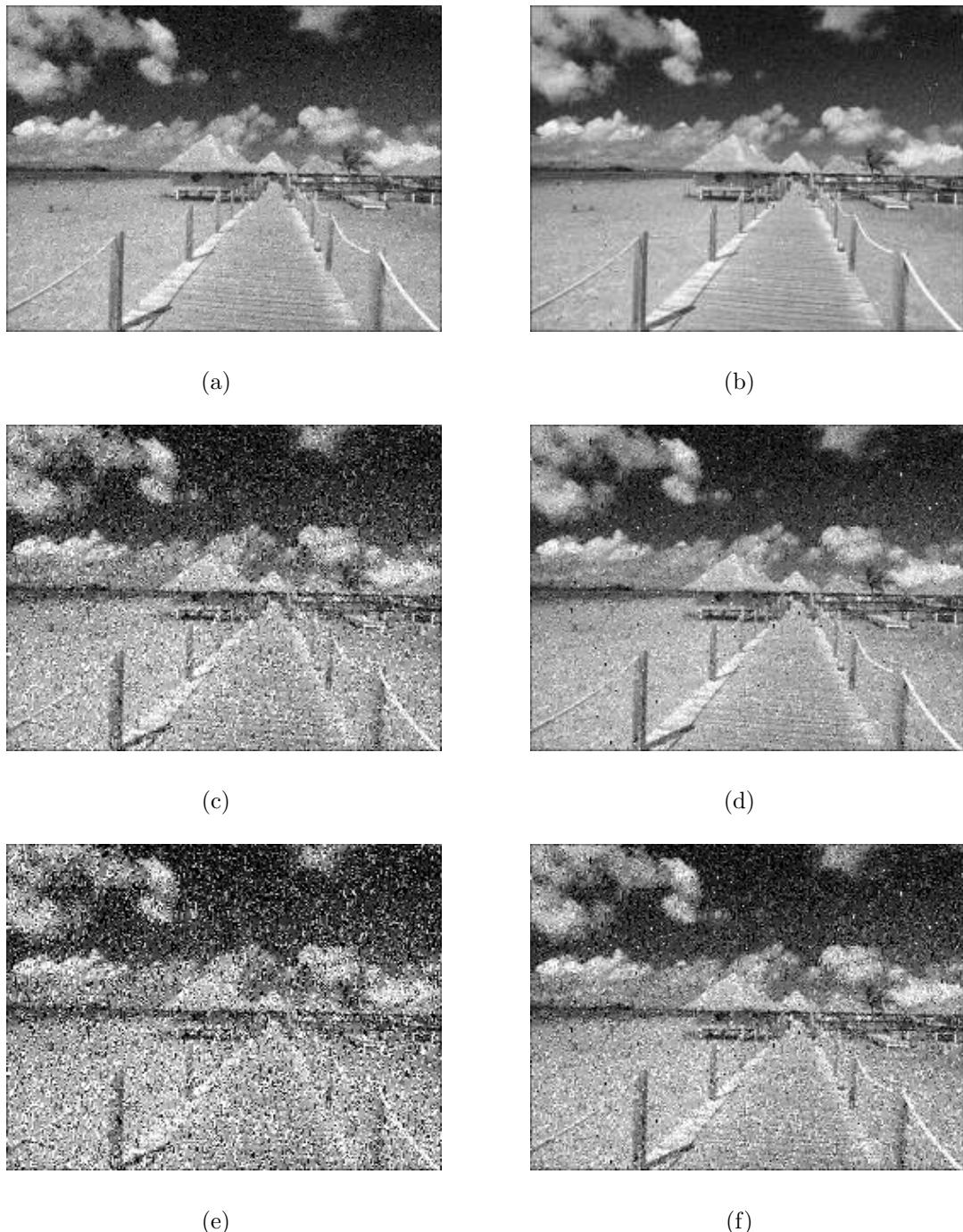


Fig. A.5: The images corrupted by the different intensity noise and the results of removing noise with ANFIS of triangle MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

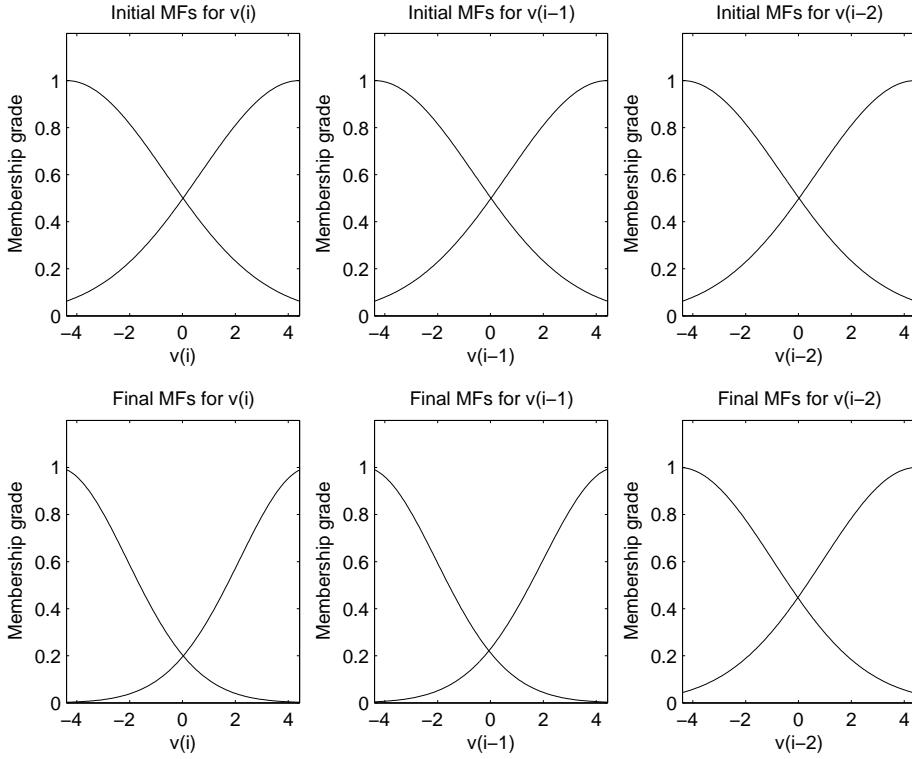


Fig. A.6: The changes of Gaussian MFs before and after training.

We show the image corrupted by low, medium and high noise and the filtered results in Fig A.9(a) to A.9(f) respectively.

The Restoration with the Trapezoidal Membership Function

The trapezoidal membership function is tested to restore the image corrupted by different intensity noise and the results are compared with other MFs for removing the noise in the nonlinear passage dynamics of order 3. Fig A.10 shows trapezoidal MFs to high noise before and after training.

The images corrupted by the intensity of low, medium and high Gaussian noise and the results of removing noise with ANFIS are shown in Fig A.11(a) through A.11(f) respectively.

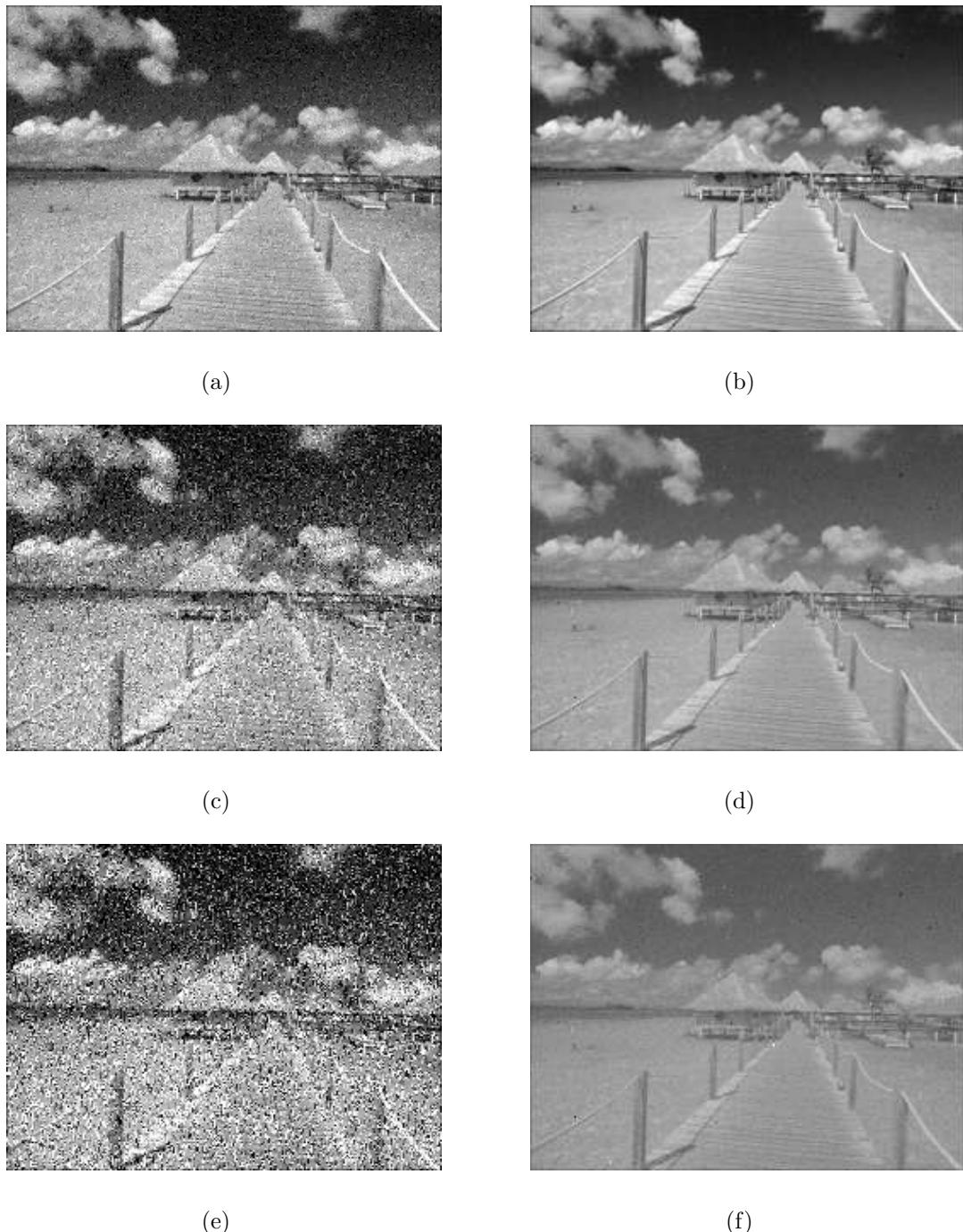


Fig. A.7: The images corrupted by the different intensity noise and the results of removing noise with ANFIS of Gaussian MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

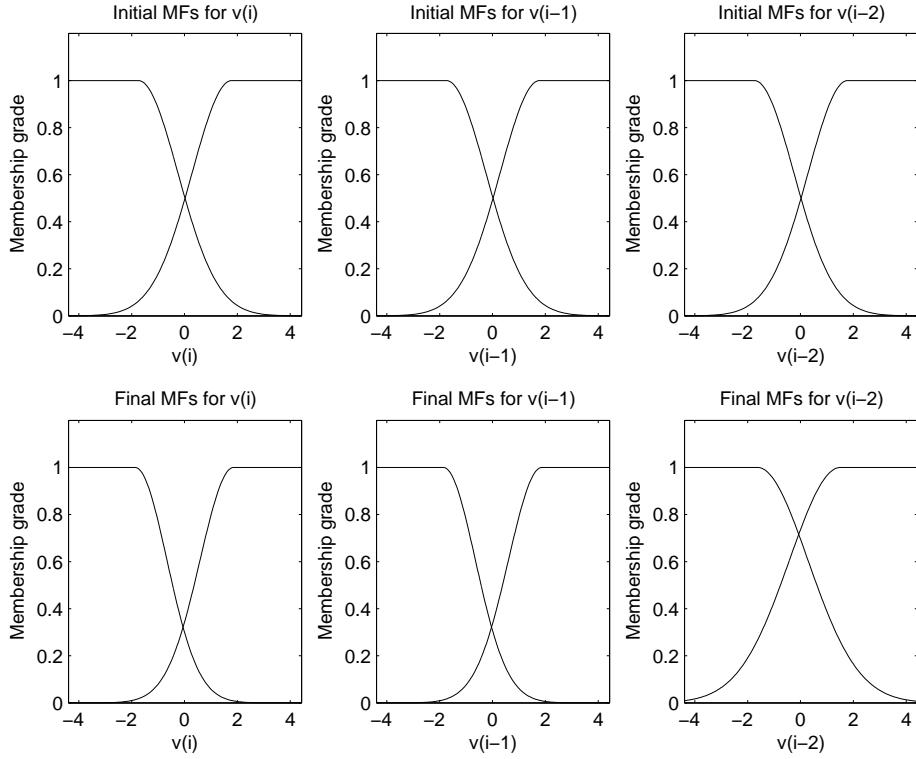


Fig. A.8: The changes of the two-sided Gaussian MFs before and after training.

The Restoration with the Product of Two Sigmoid Membership Functions

We introduce the product of two sigmoid membership functions in the nonlinear passage dynamics of order 3 to restore the image corrupted by different intensity noise and the results are compared with other MFs for removing the noise. Fig A.12 shows the product of two sigmoid MFs to high noise before and after training. The same SNRs are used for the intensity of low, medium and high noise.

The images corrupted by the intensity of low, medium and high Gaussian noise and the results of removing noise with ANFIS are shown in Fig A.13(a) through A.13(f) respectively.

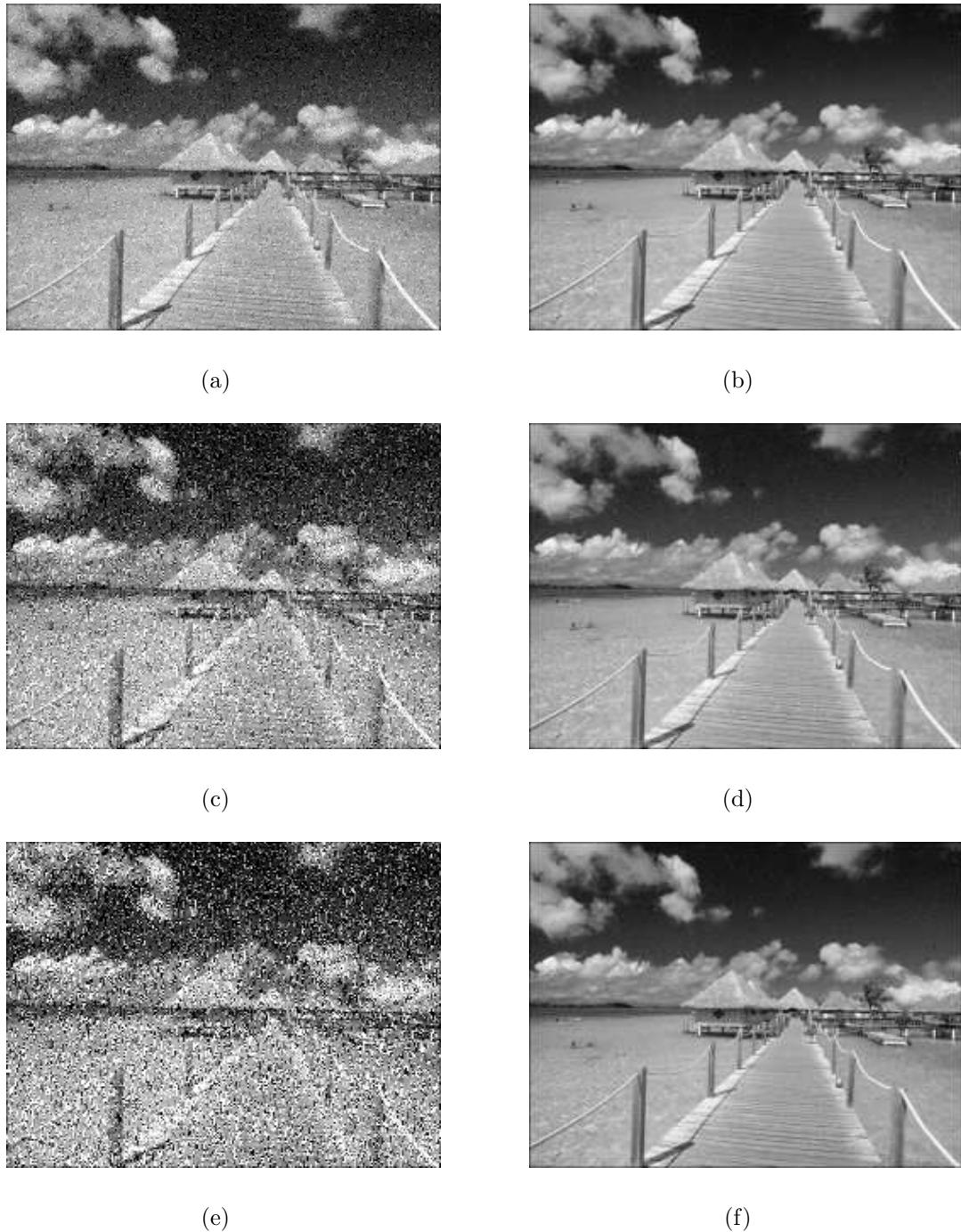


Fig. A.9: The images corrupted by the different intensity noise and the results of removing noise with ANFIS of the two sided Gaussian MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

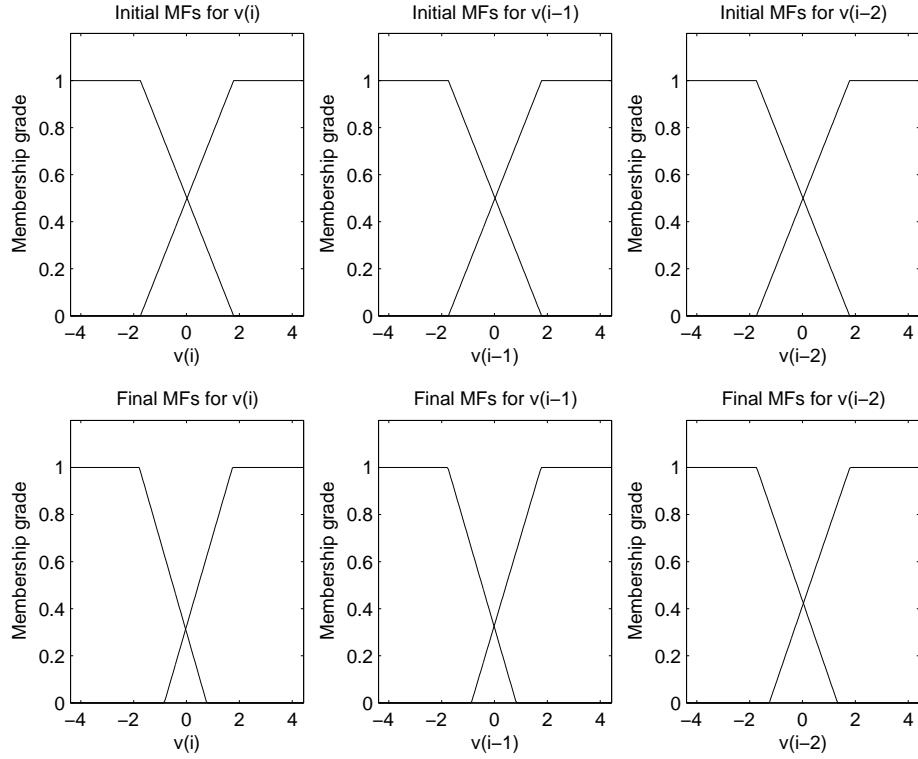


Fig. A.10: The changes of the trapezoidal MFs before and after training.

The Restoration with the Difference between Two Sigmoidal Membership Functions

We use the difference between two sigmoidal membership function in the nonlinear passage dynamics of order 3 to restore the image corrupted by different intensity noise and compare the results with other MFs for removing the noise. Fig A.14 shows the difference between two sigmoidal MFs to high noise before and after training. The SNRs are the same for the intensity of low, medium and high noise.

The images corrupted by the intensity of low, medium and high Gaussian noise and the results of removing noise with ANFIS are shown in Fig A.15(a) through A.15(f) respectively.

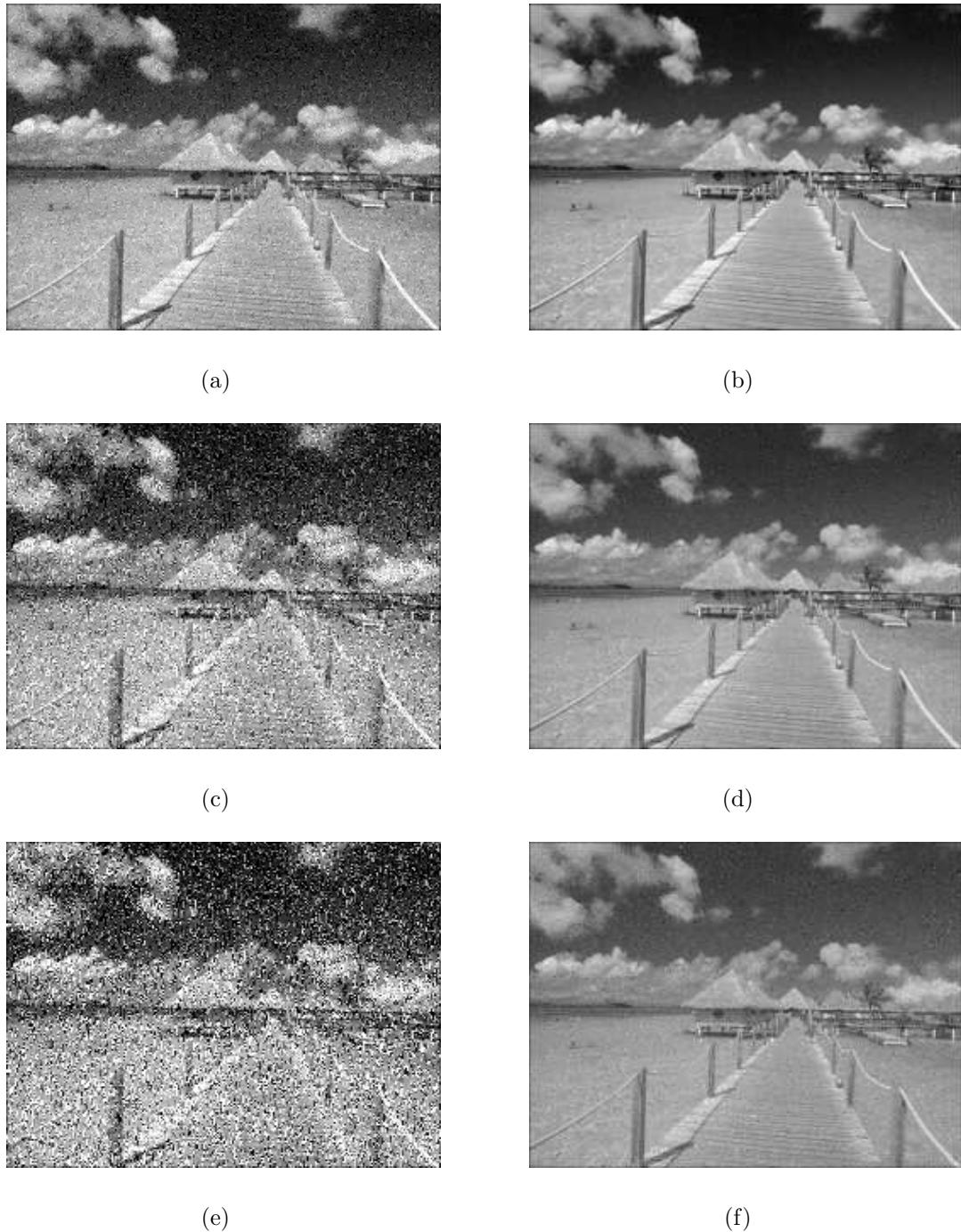


Fig. A.11: The images corrupted by the different intensity noise and the results of removing noise with ANFIS of the trapezoidal MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

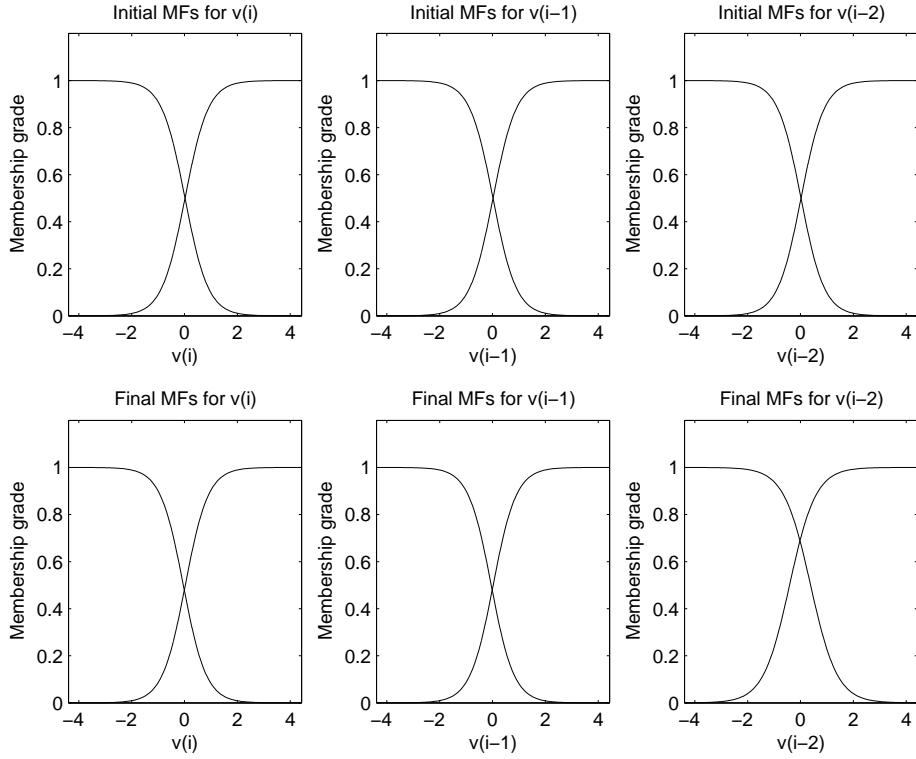


Fig. A.12: The changes of the product of two sigmoid MFs before and after training.

The Restoration with the Pi-shaped Membership Functions

We try the pi-shaped membership function in the nonlinear passage dynamics of order 3 to remove the different intensity of noise and compare the results with other MFs for cancel the noise. Fig A.16 shows the Pi-shaped MFs to high noise before and after training. The SNRs are the same for the intensity of low, medium and high noise.

The images corrupted by the intensity of low, medium and high Gaussian noise and the results of removing noise with ANFIS are shown in Fig A.17(a) through A.17(f) respectively.

A.1.2 Discussion on Parameters of ANFIS

Like the nonlinear passage dynamics of order 2, now we discuss the parameters of training epoch number, the number membership function for each input, output membership function type, the optimization method, and the training data and checking

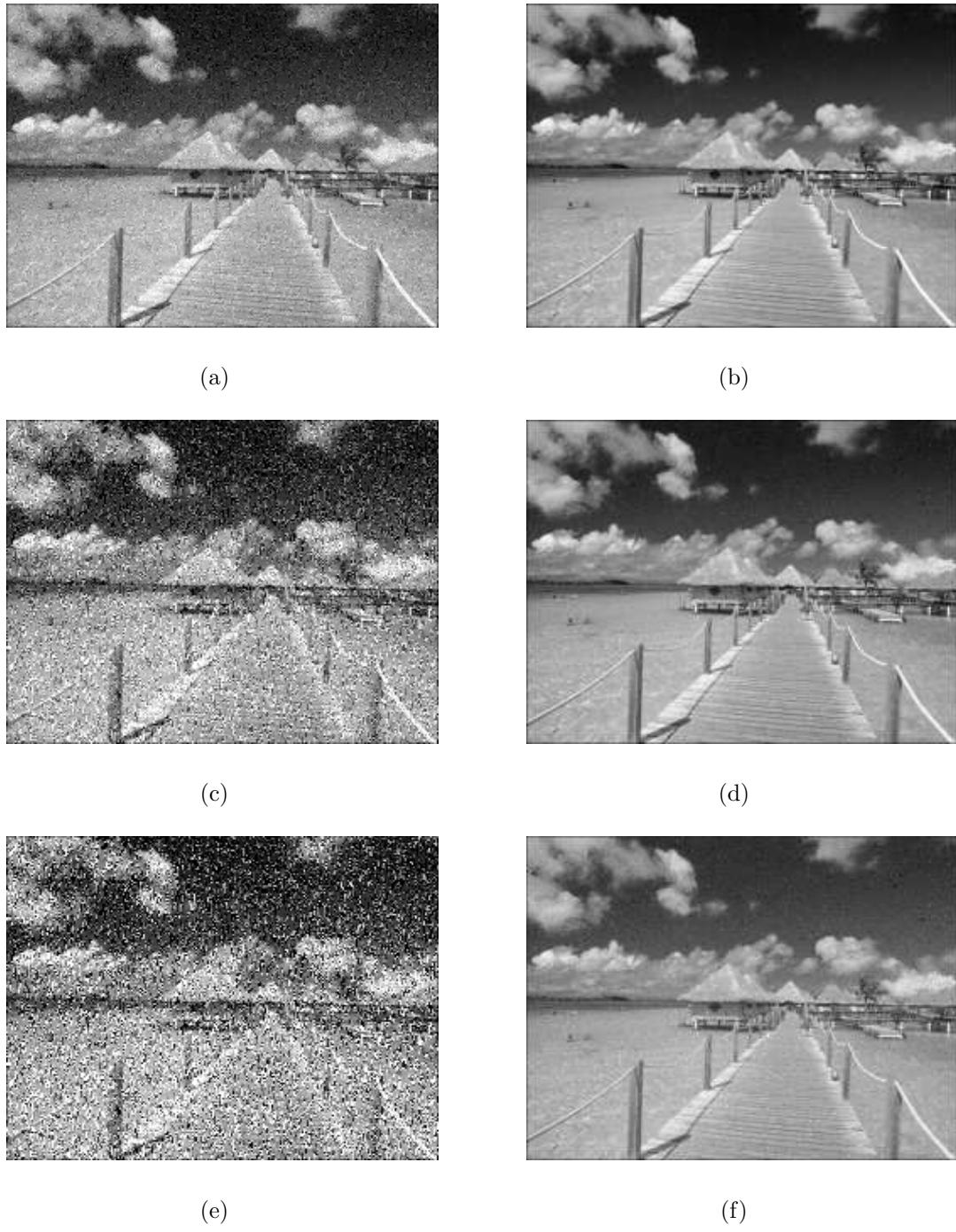


Fig. A.13: The images corrupted by the different intensity noise and the results of removing noise with ANFIS of the product of two sigmoid membership functions. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

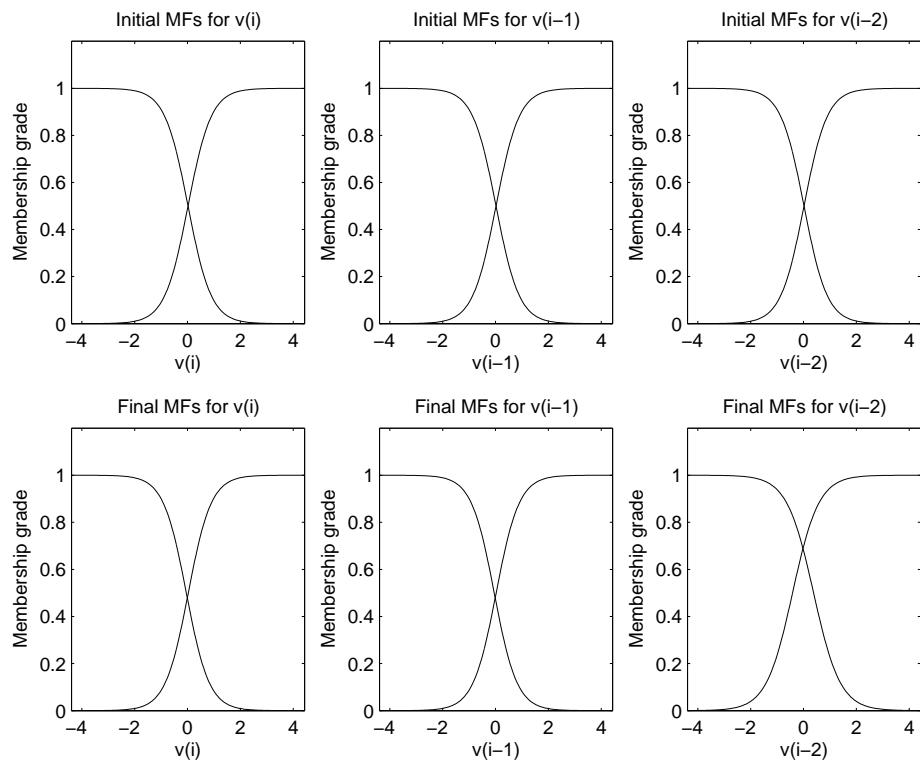


Fig. A.14: The changes of the difference between two sigmoidal MFs before and after training.

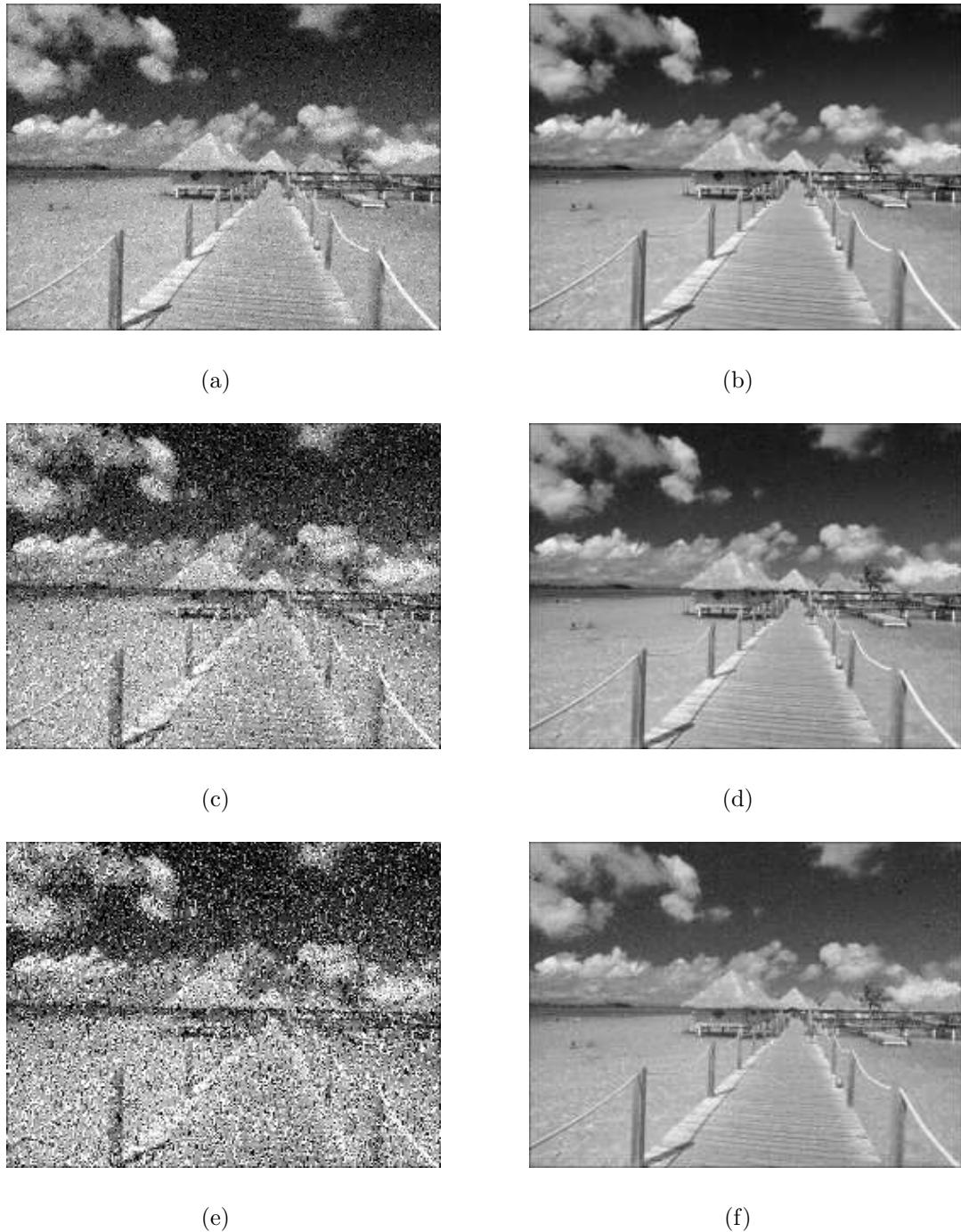


Fig. A.15: The images corrupted by the different intensity noise and the results of removing noise with ANFIS of the difference between two sigmoidal MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

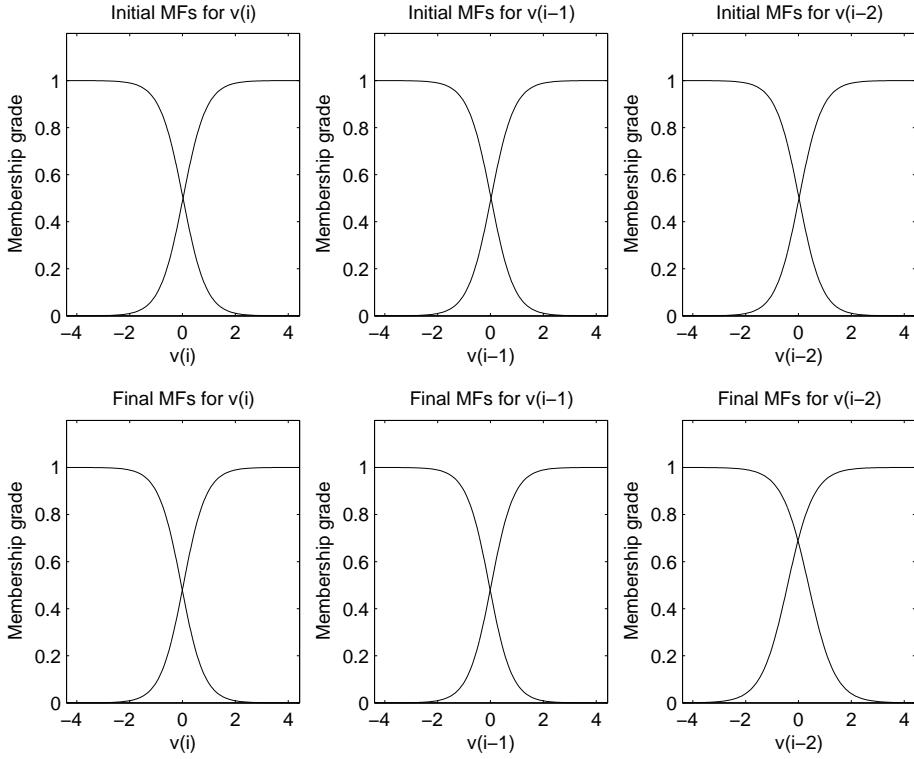


Fig. A.16: The changes of the pi-shaped MFs before and after training.

data.

The Training Epoch Number

We set the training epoch number 50, 100, 200 and 400 with the bell MFs separately. We only discuss with the intensity of high noise because the results of high noise is typical of all other intensity noise.

We can obtain the result that with the increasing training epoch number, the restoration image with ANFIS become better and better from Fig A.18(a) to A.18(f). When training epochs increase, the MSEs almost have no change and the effect of the restoration image are almost the same.

The Number of Membership Function for Each Input

Now we discuss the affection of the number of bell MFs for each input to the image restoration corrupted by high Gaussian noise. We only assume the number of MFs

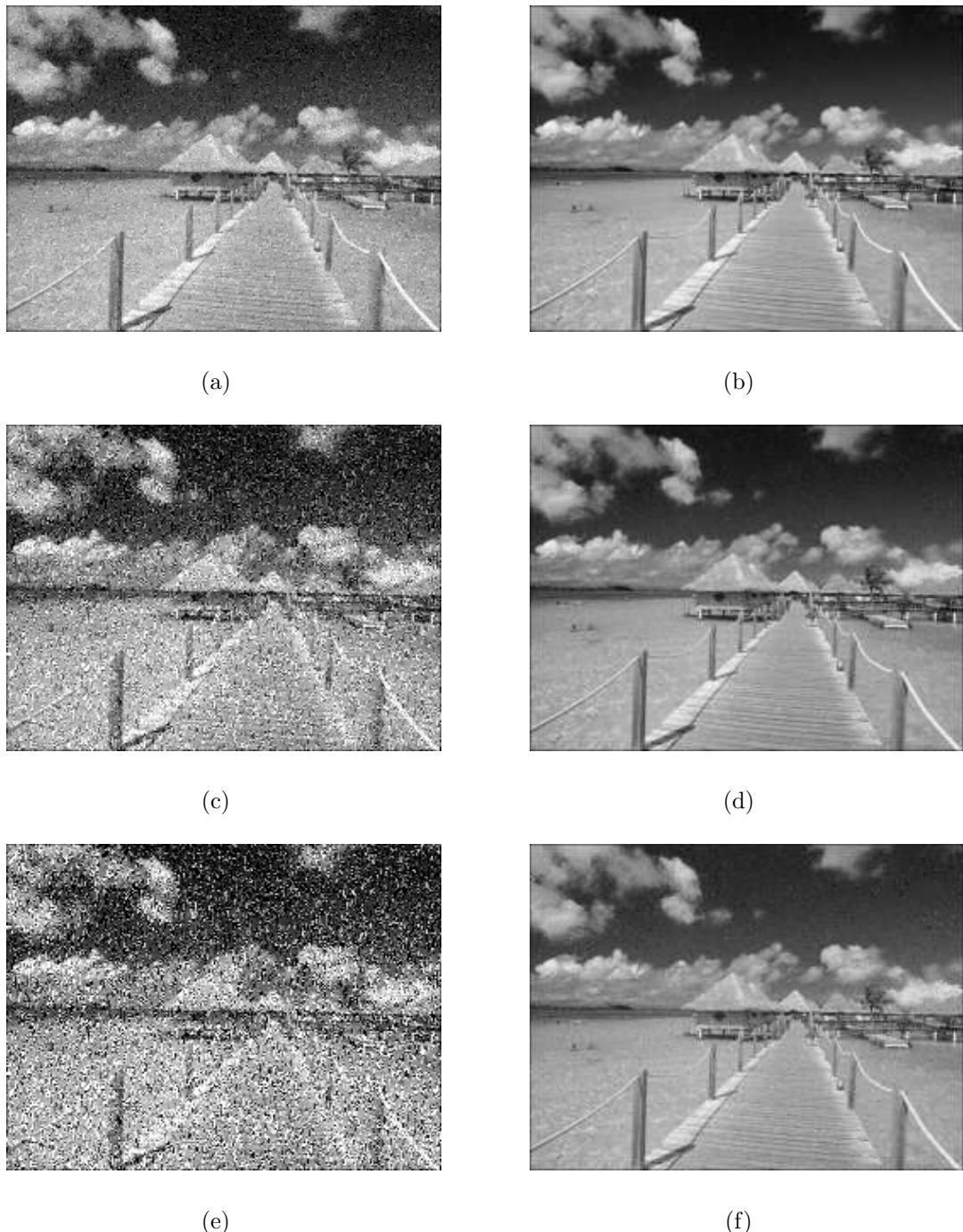


Fig. A.17: The images corrupted by the different intensity noise and the filtered images with ANFIS of the pi-shaped MFs. (a) The image with low noise; (b) The restoration from low noise; (c) The image with medium noise; (d) The restoration from medium noise; (e) The image with high noise; (f) The restoration from high noise.

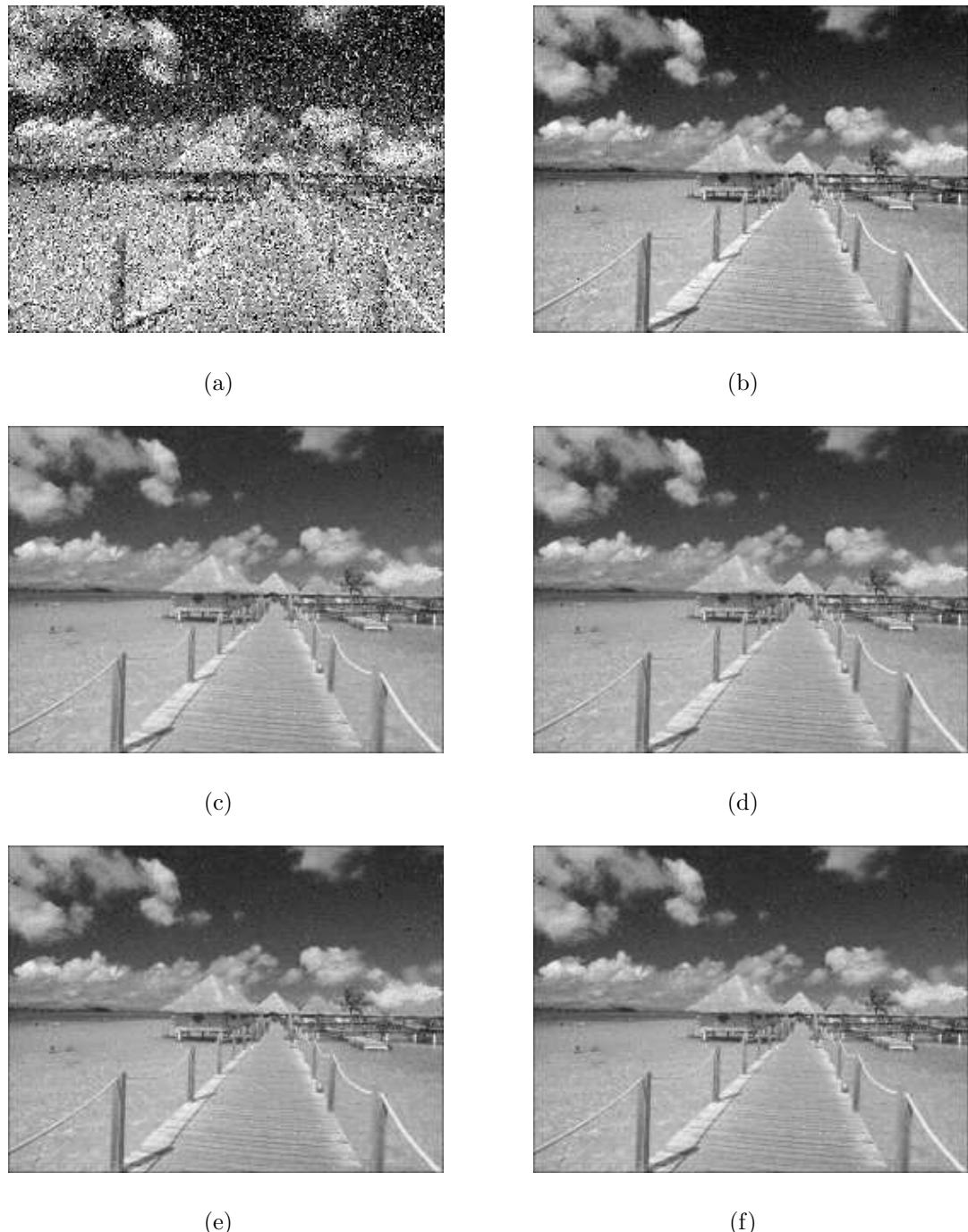


Fig. A.18: The images corrupted by the high Gaussian noise and the results of removing noise with ANFIS of the bell MFs in the different training epoch number.
(a) The image with high noise; (b) 20 epochs; (c) 50 epochs; (d) 100 epochs; (e) 200 epochs; (f) 400 epochs.

for each input is 3 and 4, bell MFs separately because the number of MFs for each input more than 4 consumes too time for running the programs. The change of 3 and 4 bell MFs for each input before and after training are shown in Fig A.19 to A.20. The image contaminated with noise and restoration images with 2, 3 and 4 MFs for each input are shown in Fig A.21(a) to A.21(d).

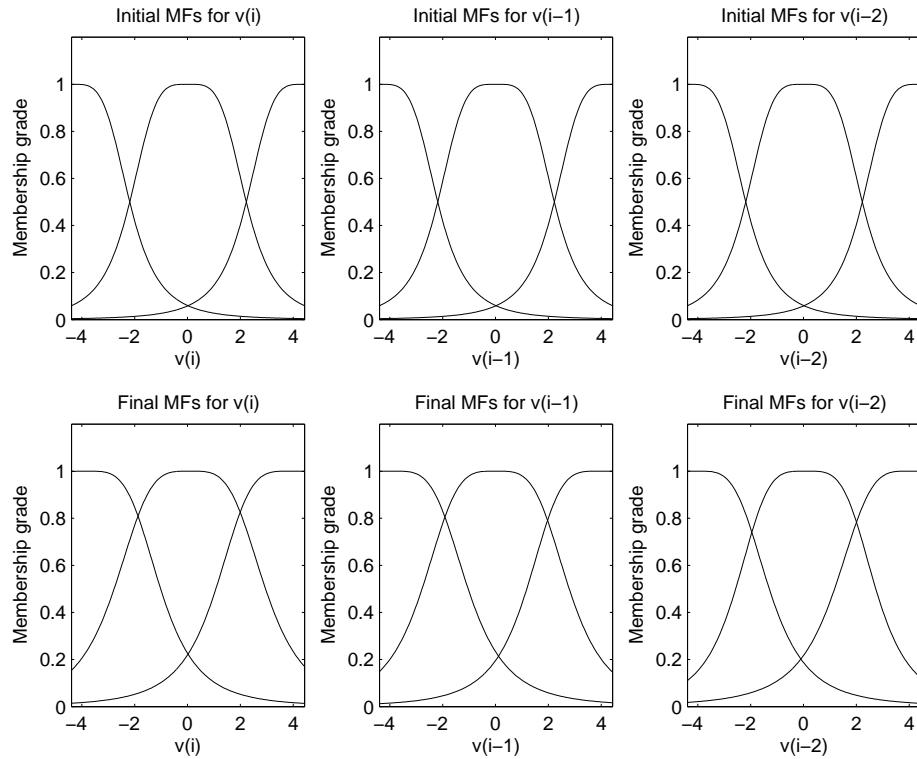


Fig. A.19: The changes of 3 MFs for each input before and after training.

The Optimization Method Used in Training

We use the backpropagation method compare with the combination of least-squares and backpropagation gradient descent methods in a nonlinear passage dynamics of order 3.

The restoration image with the backpropagation method compared with a hybrid learning algorithm are shown in Fig A.22(a) through A.22(f) respectively.

From Table 4.9 and Fig. A.22, we find that the effect of removing noise from the contaminated image by hybrid learning algorithm is much better than by the

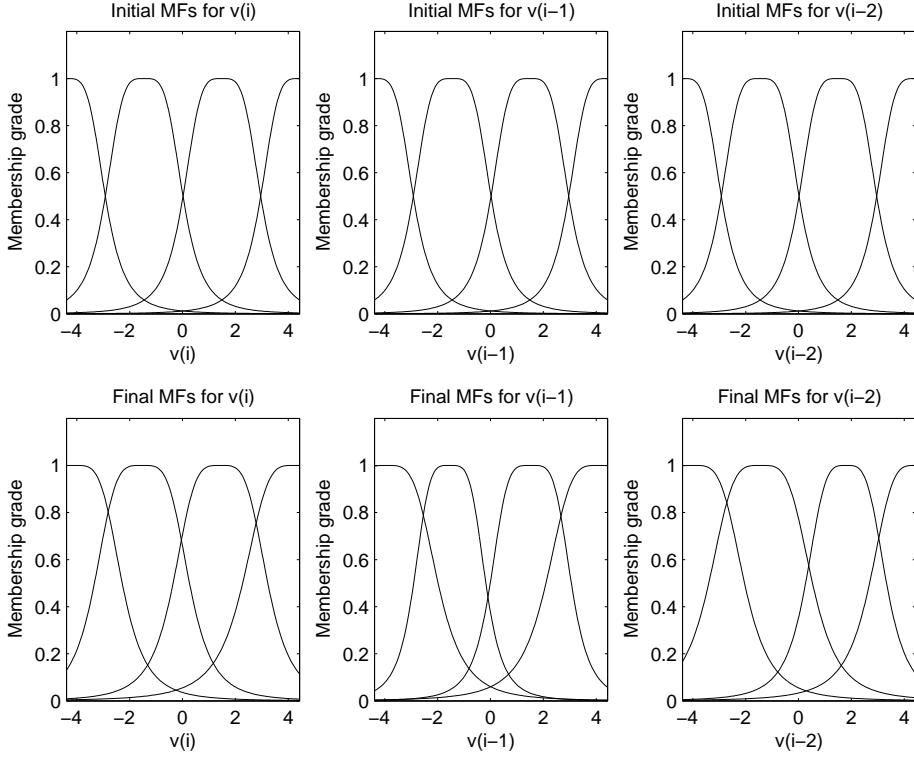


Fig. A.20: The changes of 4 MFs for each input before and after training.

backpropagation method.

The Output Membership Function Type

We compare the *constant* output MF with the *linear* output MF. The restoration image with the output MFs type as *constant* compared with as *linear* are shown in Fig A.23(a) through A.23(f) respectively.

From Table 4.10 and Fig. A.23, we find that the effect for removing noise from the contaminated image by the output MF type of *linear* is much better than by the output MF type of *constant*.

The Training Data and Checking Data

We can split the whole data into two halves. One half is called the training data and another half the checking (validation) data.

We first show the RMSE curve Fig. A.24 when the image is corrupted by high

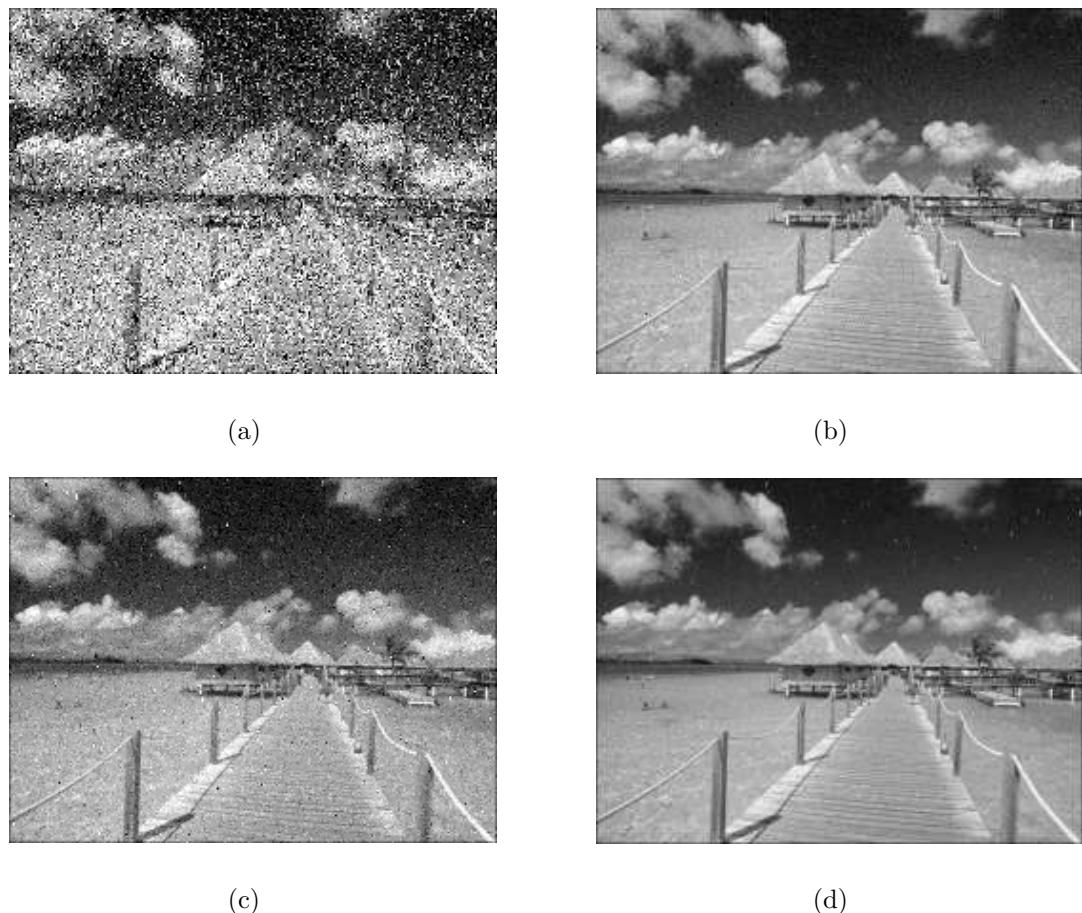


Fig. A.21: The images corrupted by the high Gaussian noise and the results of removing noise with ANFIS in the different number of the bell MFs for each input. (a) The image with high noise; (b) 2MFs; (c) 3MFs; (d) 4MFs.

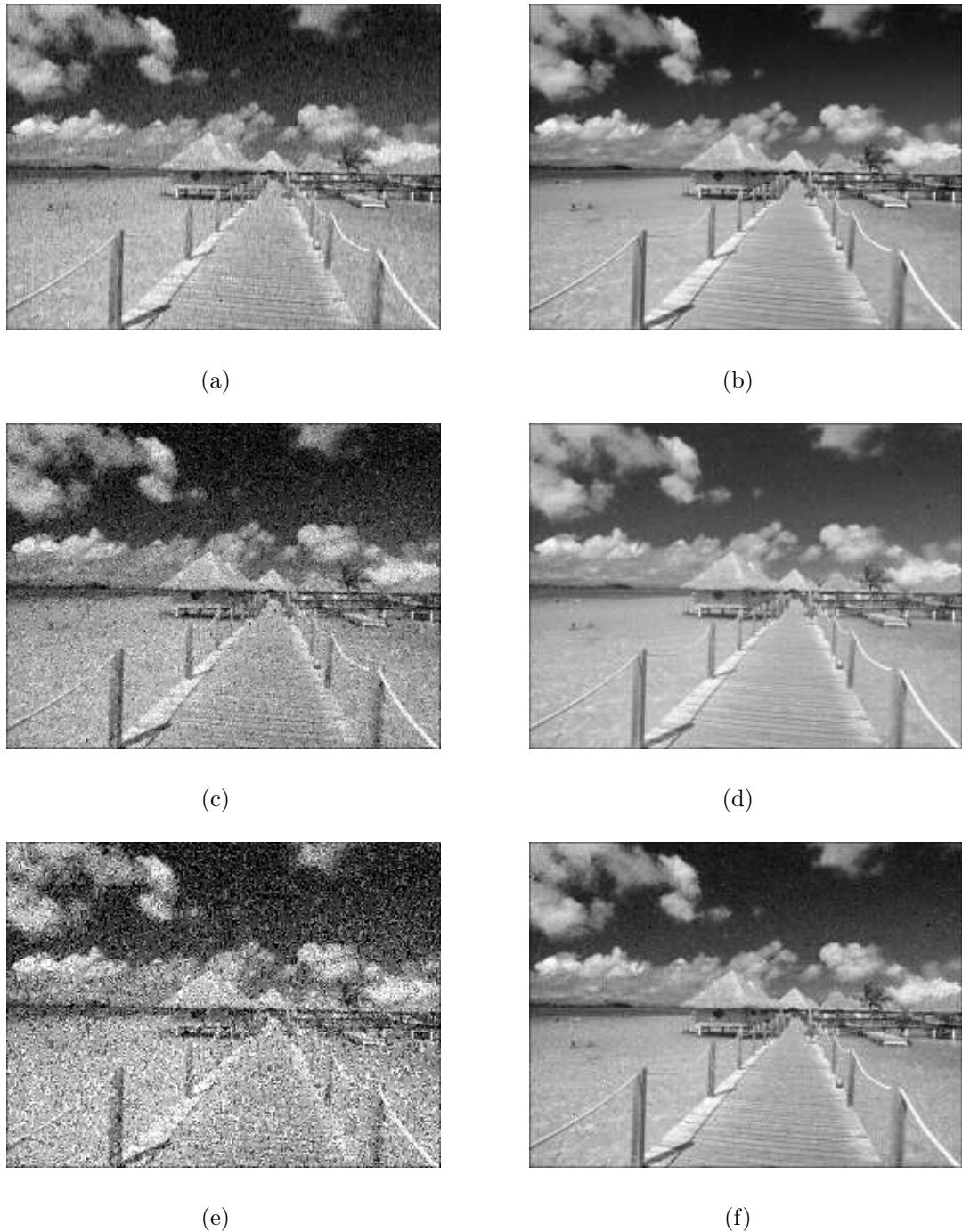


Fig. A.22: The restoration images corrupted by the different intensity noise by the backpropagation method and by a hybrid learning algorithm with ANFIS of the bell MFs. (a), (c) and (e) Backpropagation method restored from low , medium and high noise respectively; (b), (d) and (f) Hybrid learning algorithm restored from low, medium and high noise respectively.

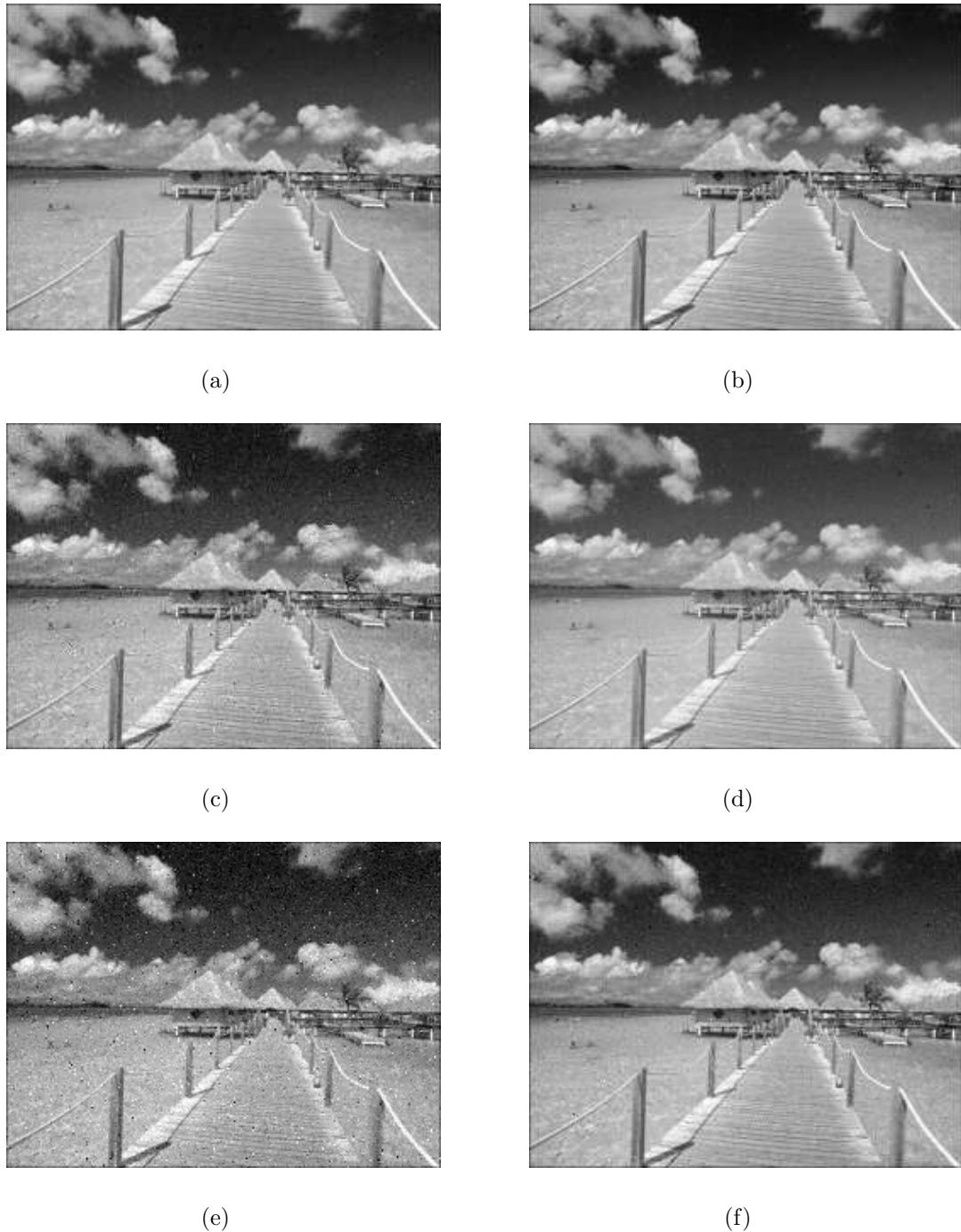


Fig. A.23: The restoration images corrupted by the different intensity noise by the output MF type of *constant* and *linear* with ANFIS of the bell MFs. (a), (c) and (e) using the output MF type of *constant* restored from low, medium and high noise respectively; (b), (d) and (f) using the output MF type of *linear* restored from low, medium and high noise respectively.

noise. Overfitting can not be detected because when the training error is slightly decreasing, the checking error also descend a little. The changes of the MFs of the training data and checking data before and after training are shown in Fig. A.25.

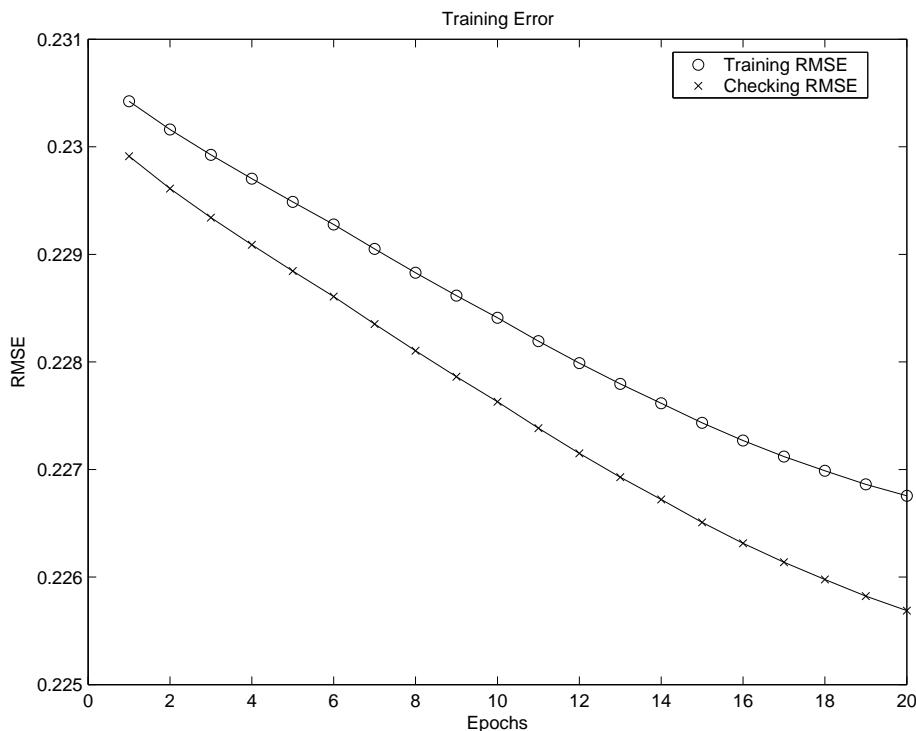


Fig. A.24: The changes of RMSEs of the training data and the checking data.

A.2 Application of ANFIS to a Nonlinear Passage Dynamics of Order 2 to Restore an Image Corrupted by Salt and Pepper Noise

Like the general rule in Chapter 4, we generally assign number of MFs per input as two, output MF type as *linear*, type of MF as *bell*, and training epoch number as 20. Usually the bell MFs are used as defaulted MFs. In general, the initial step size, step size decrease rate, and step size increase rate are 0.1, 0.9 and 1.1 respectively. A combination of least-squares and backpropagation gradient descent methods are

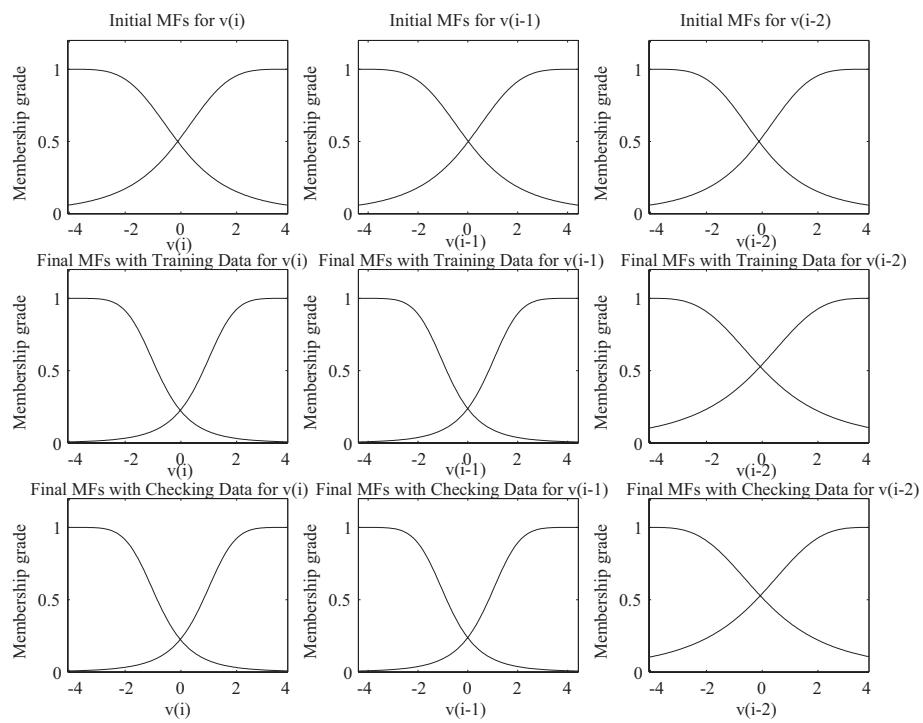


Fig. A.25: The changes of the MFs of the training data and checking data before and after training.

used for training MF parameters to model a given set of input/output data. The output membership function type is *linear*. In the experiment, the unknown nonlinear passage dynamics are assumed to be defined as in Eq.(4.1).

We just analyze the restoration of image contaminated with high salt & pepper noise which SNR is -1.4564 because the changes between the different intensity noise after restoration reach almost the same effect. In the other word, its restoration effect is much better than that of image contaminated with the same extent Gaussian noise. We will discuss the reasons later.

Fig A.26(a) shows the original RGB (red, green and blue) color image ‘*Maldives*’. We use the same method to transfer it to the gray iamge. The resulting image after processing is shown in Fig A.26(b).



Fig. A.26: The original image. (a) The original color image; (b) The original gray image.

The measurable source noise is salt & pepper noise which the probability of the distorted noise signal $v_0(i)$ is 0.3017 shown in Fig. A.27(a). The distorted noise $v_0(i)$ caused by the source noise $v_1(i)$ and produced by the nonlinear dynamics of order 2 in Eq.(4.2) is shown in Fig. A.27(b). We show the estimated distorted signal $y_0(i)$ and the estimated error in Fig A.27(c) and Fig. A.27(d) respectively. We can find Fig. A.27(d) showing black because the the estimated error between the estimated

distorted noise $y(i)$ by ANFIS and the distorted noise $v_0(i)$ is near zero matrix (less than 10^{-3}).

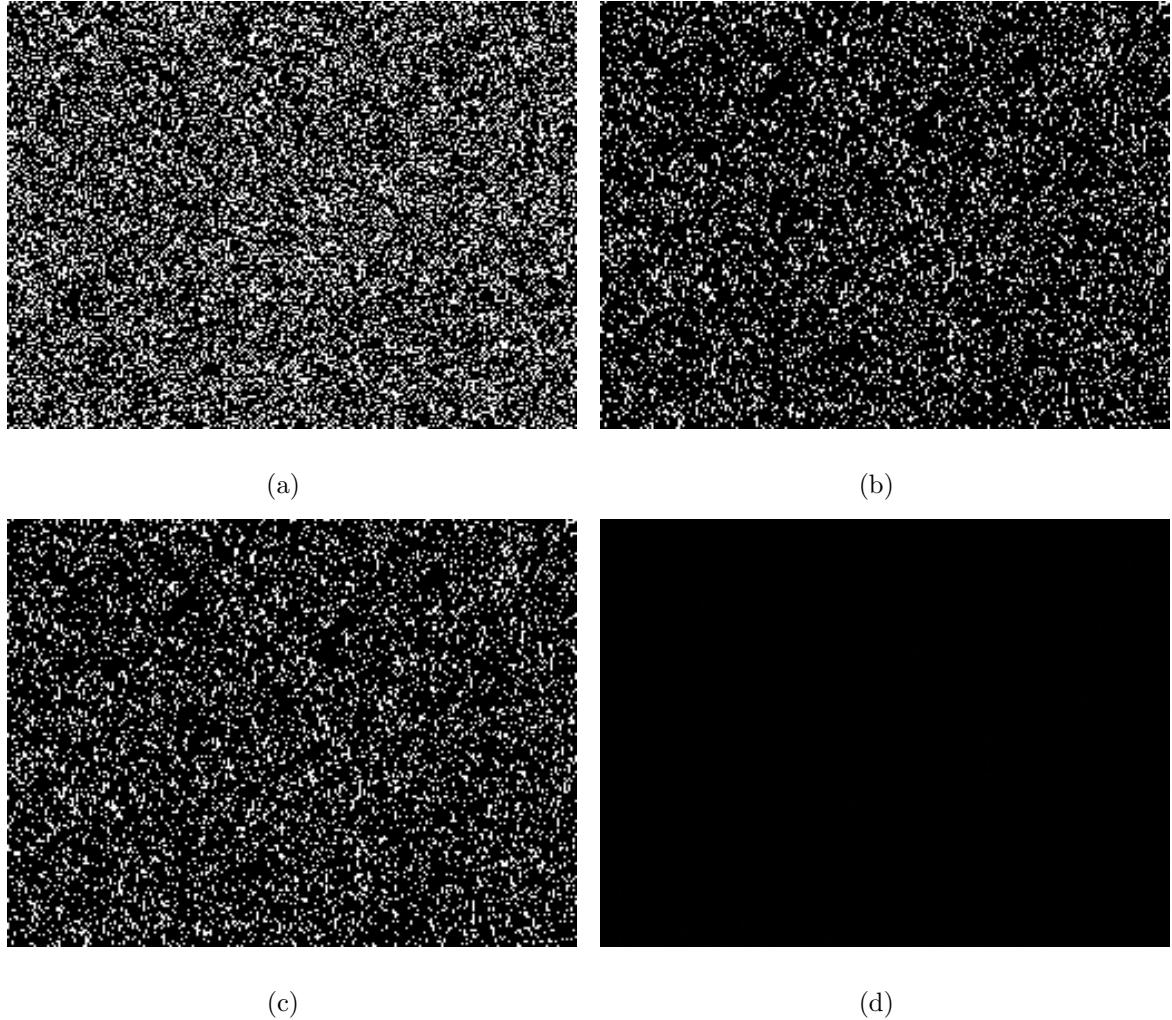


Fig. A.27: The signals being used in ANFIS. (a) The measurable source noise $v_1(i)$; (b) The distorted noise $v_0(i)$; (c) The estimated distorted noise $y(i)$ by ANFIS; (d) The error between the estimated distorted noise $y(i)$ by ANFIS and the distorted noise $v_0(i)$.

We now investigate the behavior of these signals in the frequency domain before we go on to the next. Fig. A.46(a) to A.46(d) display the spectral density distributions of $s(i)$, $v_1(i)$, $v_0(i)$ and $x(i)$, respectively, from the first 256 points. Obviously, the spectra of the information signal $s(i)$ and the distorted noise $v_0(i)$ overlap each other considerably. This makes it impossible to apply common frequency domain filtering

methods to remove $v_0(i)$ from $x(i)$. Fig A.29 is the ANFIS surface $\hat{f}(\cdot)$ after 20 epochs of batch learning.

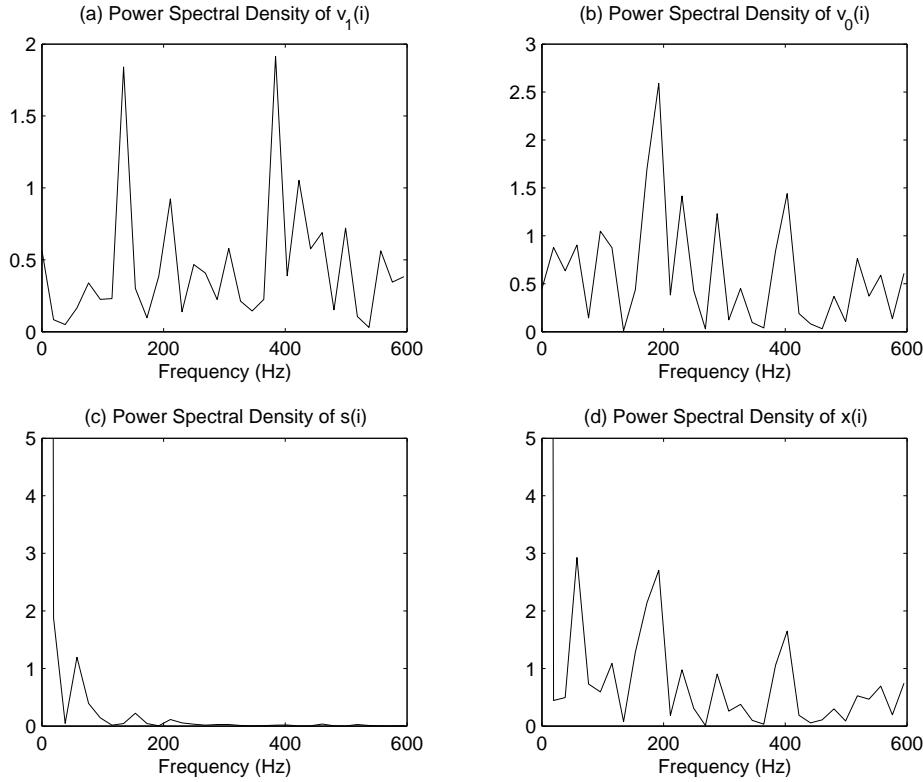


Fig. A.28: Spectral density distributions.

For visually displaying how big the error between the estimated distorted noise $y(i)$ and the distorted noise $v_0(i)$ is, we change the two dimensional pixel matrix into one dimensional vector shown in Fig. A.30. We can find the error is particularly small not beyond 10^{-2} .

A.2.1 Application of ANFIS with Different MFs

In this section, we discuss the effect of different MFs of input for filtering the image contaminated with low, medium and high noise. The names of these MFs are bell MF, Gaussian MF, two-sided Gaussian MF, triangle MF, trapezoidal MF, product of two sigmoid MFs, difference between two sigmoidal MFs, pi-shaped MF.

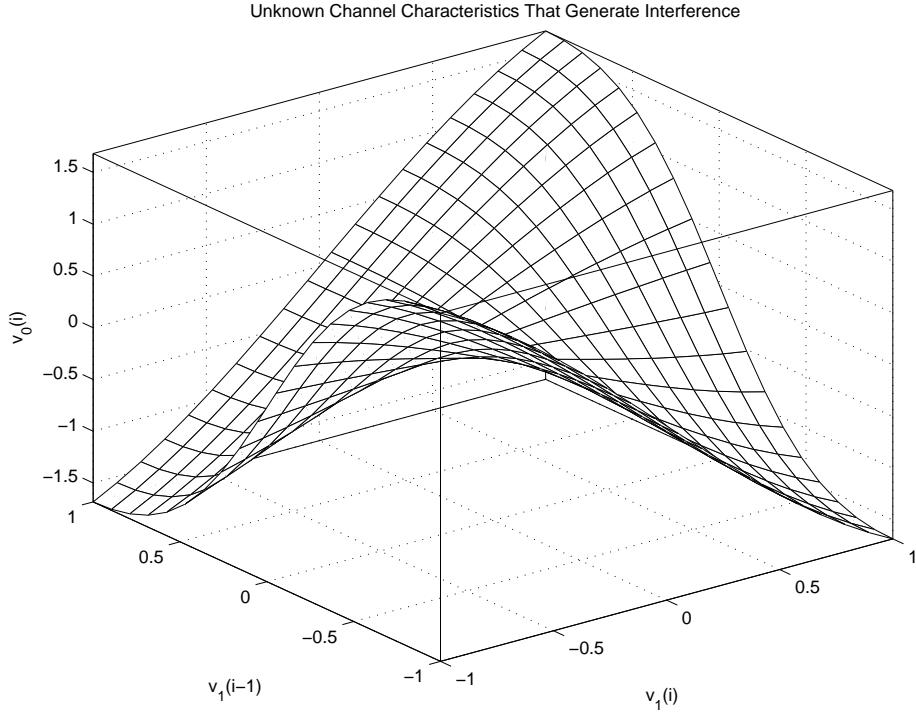


Fig. A.29: The characteristics of ANFIS function \hat{f} .

The Restoration under ANFIS with the Bell Membership Function

First we choose the bell membership function as MFs of this ANFIS. There are two inputs, $v_1(i - 1)$ and $v_0(i)$, for nonlinear passage dynamics of orders 2 and two MFs for each input. We choose probabilities are 0.1008, 0.1812 and 0.3017 correspondent to low, medium and high salt & pepper noise respectively. The images corrupted by salt & pepper noise and the results of removing noise with different PDF with ANFIS are shown in Fig A.31(a) through A.31(f) respectively.

Because the change is very small, we just compare the case when the image corrupted by the heavy noise which the probability is 0.3017 and ignore the affection of the other light noise.

The Restoration under ANFIS with the Triangle Membership Function

Now we choose the triangle membership function as MFs of this ANFIS. The images corrupted by the heavy salt & pepper noise and the results of removing noise with

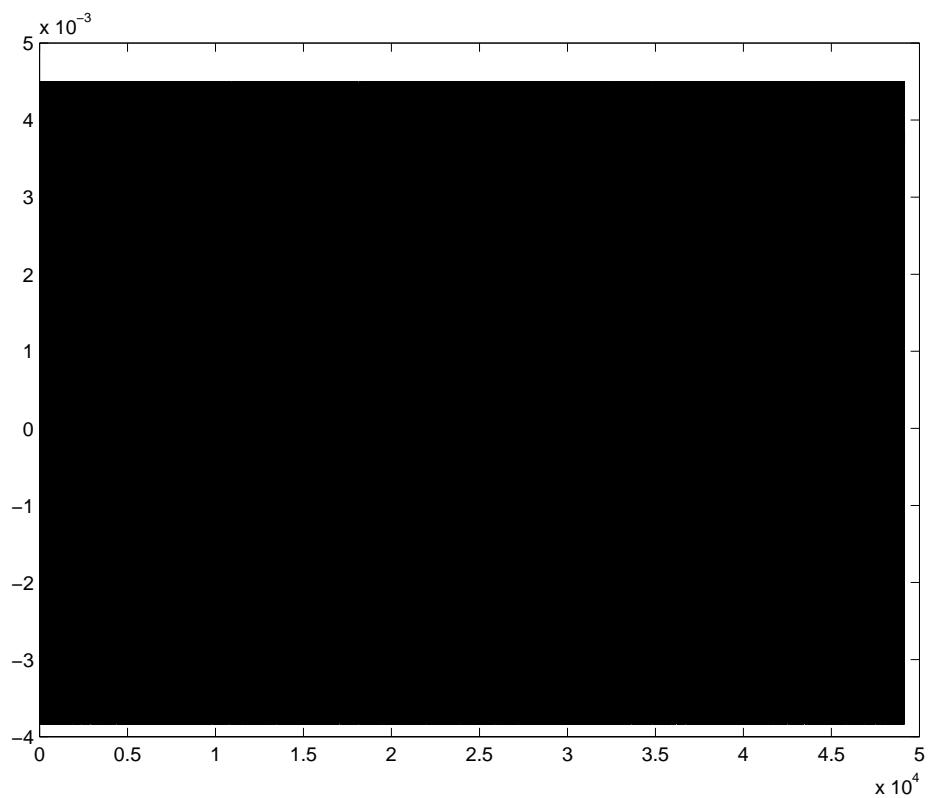


Fig. A.30: The estimated errors with ANFIS corrupted by high salt & pepper noise.



Fig. A.31: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of bell MFs. (a) The image contaminated with low salt & pepper noise ($\text{PDF}=0.1008$); (b) The restored image; (c) The image contaminated with medium salt & pepper noise ($\text{PDF}=0.1812$); (d) The restored image; (e) The image contaminated with heavy salt & pepper noise ($\text{PDF}=0.3017$); (d) The restored image.

ANFIS are shown in Fig A.32(a) through A.32(b) respectively.



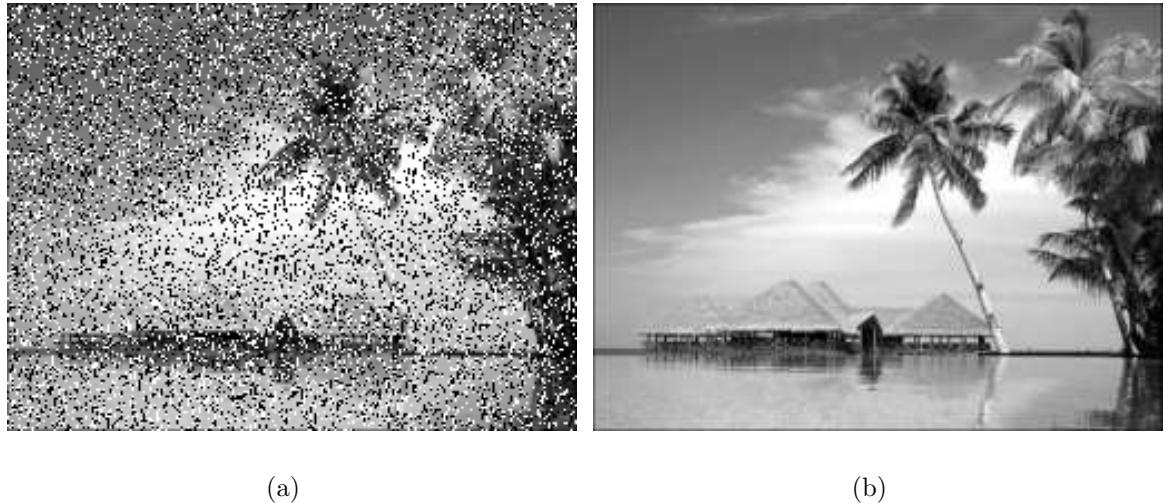
Fig. A.32: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the triangle MFs. (a) The image contaminated with salt & pepper noise; (b) The restored image.

The Restoration under ANFIS with the Gaussian Membership Function

The Gaussian membership function is now introduced as MFs of this ANFIS. The images corrupted by the heavy salt & pepper noise and the results of removing noise with ANFIS are shown in Fig A.33(a) through A.34 respectively.

The Restoration under ANFIS with the Two-sided Gaussian Membership Function

The two-sided Gaussian membership function is tested to restore the image corrupted by the heavy salt & pepper noise. The images corrupted by salt & pepper noise and the results of removing noise with ANFIS are shown in Fig A.34(a) through A.34(b) respectively.



(a)

(b)

Fig. A.33: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of Gaussian MFs. (a) The image contaminated with salt & pepper noise; (b) The restored image.



(a)

(b)

Fig. A.34: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the two-sided Gaussian MFs. (a) The image contaminated with salt & pepper noise (b) The restored image.

The Restoration under ANFIS with the Product of Two Sigmoidal Membership Function

We use the product of two sigmoidal membership function remove the noise. The images corrupted by salt & pepper noise and the results of removing noise with ANFIS are shown in Fig A.35(a) through A.35(b) respectively.



Fig. A.35: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the product of two sigmoidal MFs. (a) The image contaminated with salt & pepper noise (b) The restored image.

The Restoration under ANFIS with the Trapezoidal Membership Function

We introduce the trapezoidal membership function as MFs of this ANFIS. The images corrupted by salt & pepper noise and the results of removing noise with ANFIS are shown in Fig A.36(a) through A.36(b) respectively.

The Restoration under ANFIS with the Difference of Two Sigmoidal Membership Function

We investigate the difference of two sigmoidal membership function as MFs of this ANFIS. The images corrupted by salt & pepper noise and the results of removing noise with ANFIS are shown in Fig A.37(a) through A.37(b) respectively.

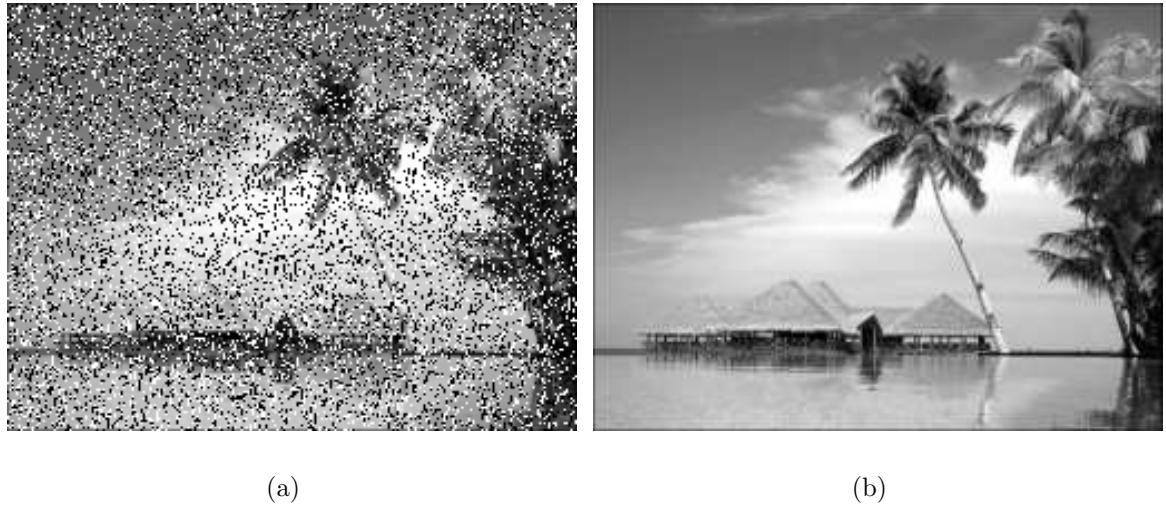


Fig. A.36: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the trapezoidal MFs. (a) The image contaminated with salt & pepper noise (b) The restored image.



Fig. A.37: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the difference of two sigmoidal MFs. (a) The image contaminated with salt & pepper noise (b) The restored image.

The Restoration under ANFIS with the Pi-shaped Membership Function

The pi-shaped membership function is tried as MFs of this ANFIS. The images corrupted by salt & pepper noise and the results of removing noise with ANFIS are shown in Fig A.38(a) through A.38(b) respectively.

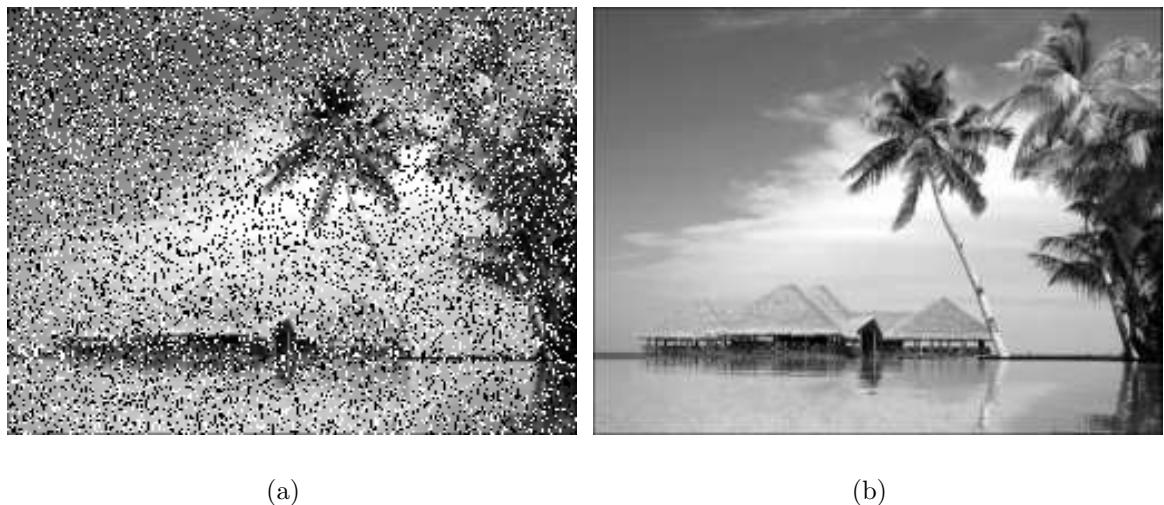


Fig. A.38: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the pi-shaped MFs. (a) The image contaminated with salt & pepper noise (b) The restored image.

A.2.2 Discussion on Parameters of ANFIS

Like removing Gaussian noise, now we discuss the parameters of training epoch number, the number membership function for each input, output membership function type, the optimization method, and the training data and checking data.

The Training Epoch Number

We set the training epoch number 50,100, 200 and 400 with the bell MFs separately to remove salt & pepper noise. The MSEs for the different epochs change particularly small in Table 4.12.

The Number of Membership Function for Each Input

Now we discuss the affection of the number of bell MFs for each input to the image restoration corrupted by salt & pepper noise. The change of 3, 4, 5, and 6 bell MFs for each input before and after training are shown in Fig A.39 to A.42.

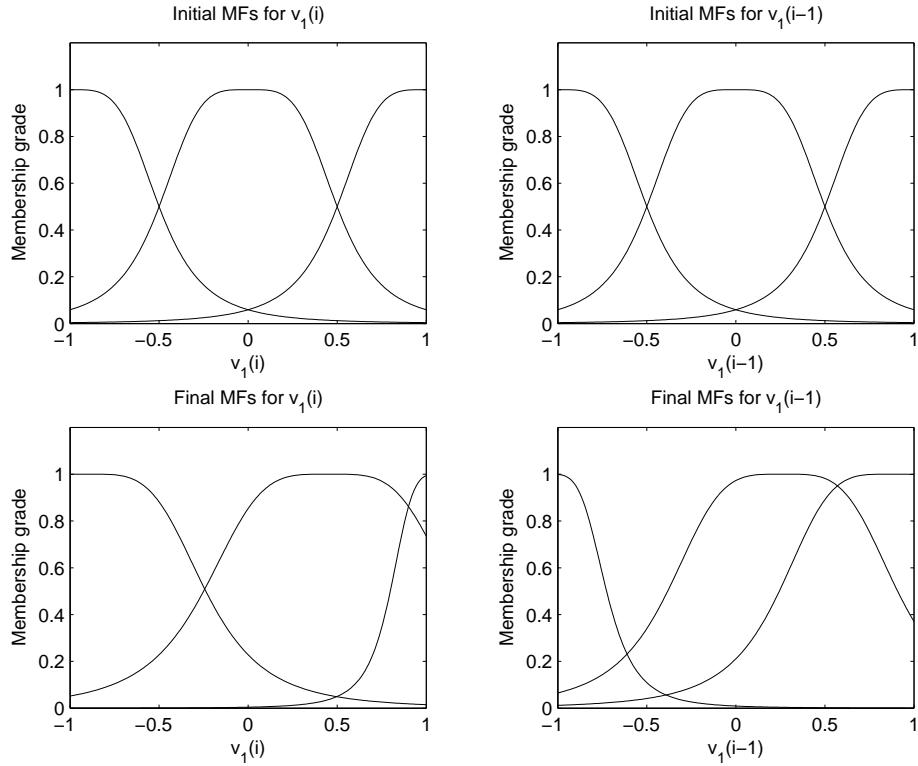


Fig. A.39: The changes of 3 MFs for each input before and after training.

The Optimization Method Used in Training

We use the backpropagation method compare with the combination of least-squares and backpropagation gradient descent methods in restoring the image corrupted with salt & pepper.

The restoration image with the backpropagation method compared with a hybrid learning algorithm are shown in Fig A.43(a) and A.43(b) respectively.

From Table 4.14 and Fig. A.43, we find that the effect of removing noise from the contaminated image by hybrid learning algorithm is much better than by the

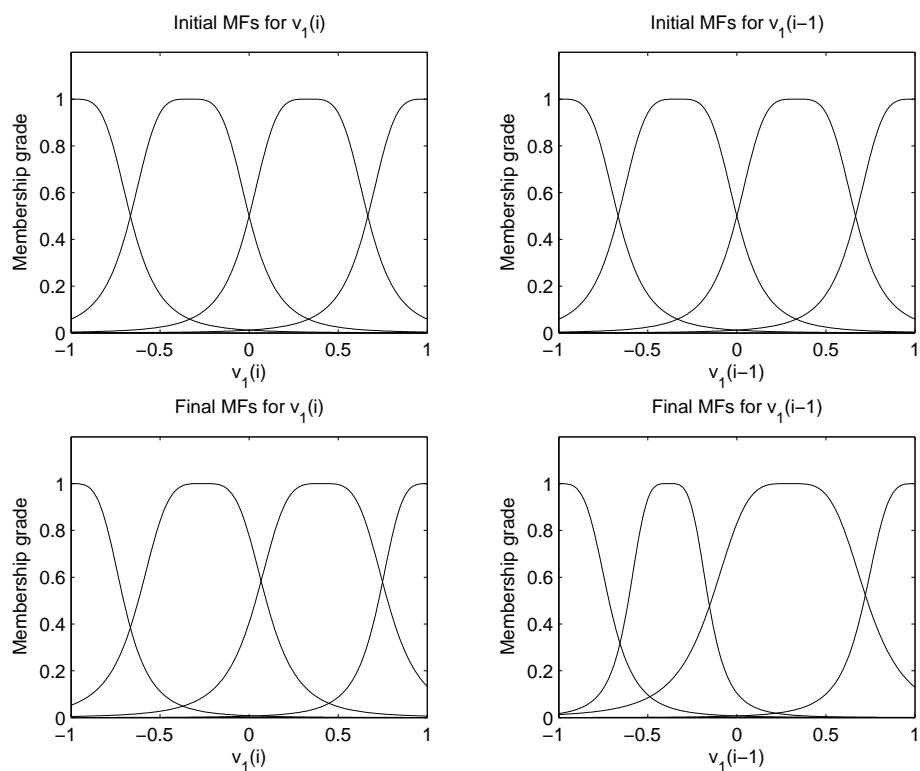


Fig. A.40: The changes of 4 MFs for each input before and after training.

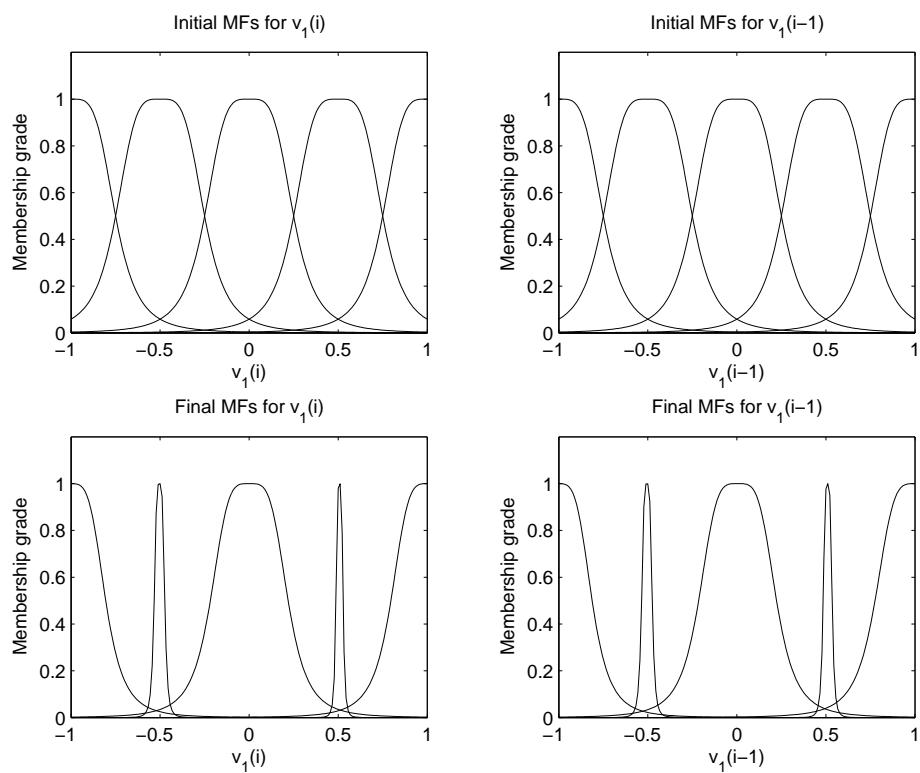


Fig. A.41: The changes of 3 MFs for each input before and after training.

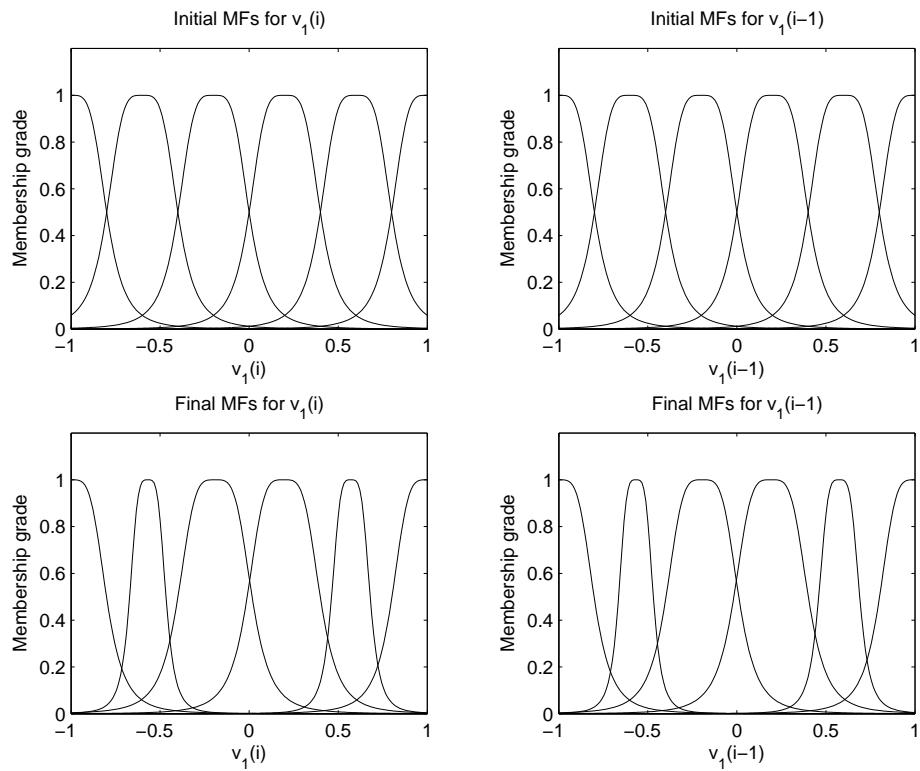


Fig. A.42: The changes of 4 MFs for each input before and after training.



Fig. A.43: The restoration images corrupted by salt & pepper noise by the backpropagation method and by a hybrid learning algorithm with ANFIS of the bell MFs. (a) Backpropagation method; (b) Hybrid learning algorithm.

backpropagation method.

The Output Membership Function Type

We compare the *constant* output MF with the *linear* output MF. From Table 4.15, we find that the effect for removing noise from the contaminated image by the output MF type of *constant* is better than by the output MF type of *linear*.

A.2.3 The Training Data and Checking Data

We can split the whole data into two halves. One half is called the training data and another half the checking (validation) data. Overfitting can be detected when the checking error starts increasing while the training error is still decreasing.

We first show the RMSE curve Fig. A.44 when the image is corrupted by high noise. Overfitting can not be detected because when the training error and the checking error all oscillate. The changes of the MFs of the training data and checking data before and after training are shown in Fig. A.45.

A.3 Application of ANFIS to a Nonlinear Passage

Dynamics of Order 3 to Restore an Image

Corrupted by Gaussian Noise

In the experiment, we use the same general rule in Chapter 4. The unknown nonlinear passage dynamics are assumed to be defined as in Eq.(A.1).

We just analyze the restoration of image contaminated with high salt & pepper noise which SNR is -1.4564 because the changes between the different intensity noise after restoration reach almost the same effect. In the other word, it restoration effect is much better than that of image contaminated with the same extent Gaussian noise. We will discuss the reasons later.

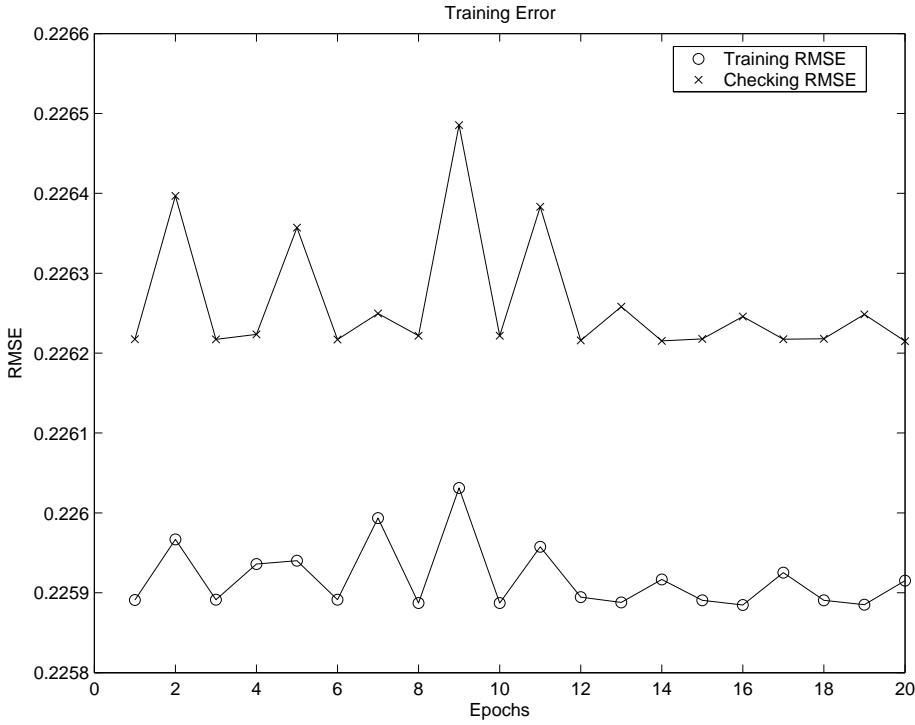


Fig. A.44: The changes of RMSEs of the training data and the checking data.

Fig A.26(a) shows the original RGB (red, green and blue) color image ‘*Maldives*’. We use the same method to transfer it to the gray iamge. The resulting image after processing is shown in Fig A.26(b).

Fig. A.46(a) to A.46(d) display the spectral density distributions of $s(i)$, $v_1(i)$, $v_0(i)$ and $x(i)$, respectively, from the first 256 points. Obviously, the spectra of the information signal $s(i)$ and the distorted noise $v_0(i)$ overlap each other considerably. This makes it impossible to apply common frequency domain filtering methods to remove $v_0(i)$ from $x(i)$.

The Restoration under ANFIS with the Bell Membership Function

First we choose the bell membership function as MFs of this ANFIS. There are three inputs, $v_1(i - 2)$, $v_1(i - 1)$ and $v_0(i)$, for nonlinear passage dynamics of orders 3 and two MFs for each input. We choose probabilities are 0.2793 separately. The images corrupted by salt & pepper noise and the results of removing noise with different

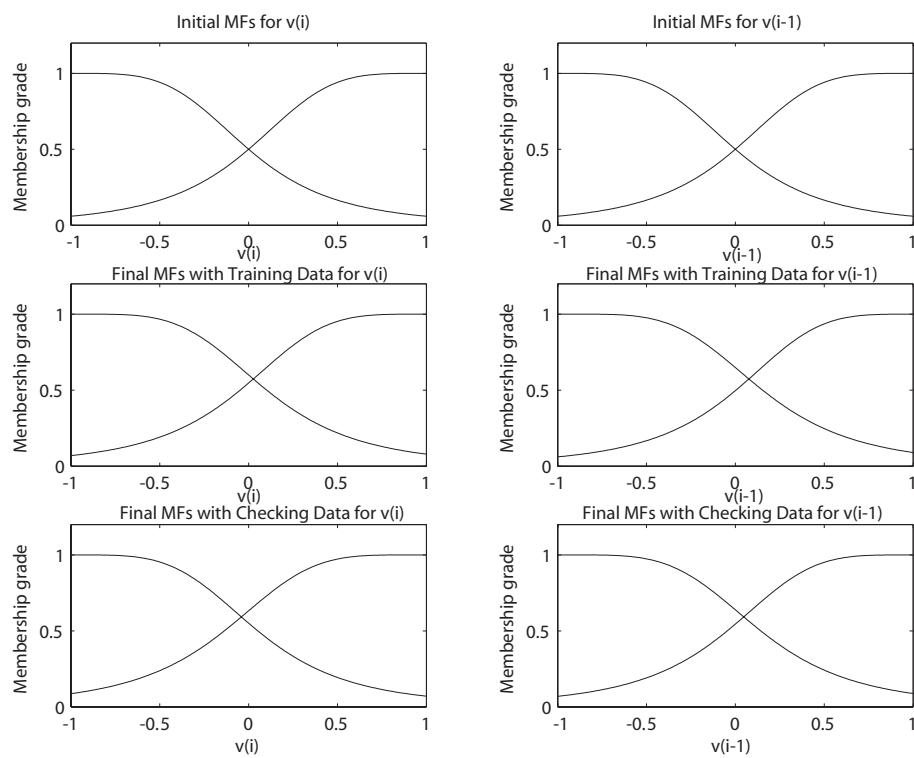


Fig. A.45: The changes of the MFs of the training data and checking data before and after training.

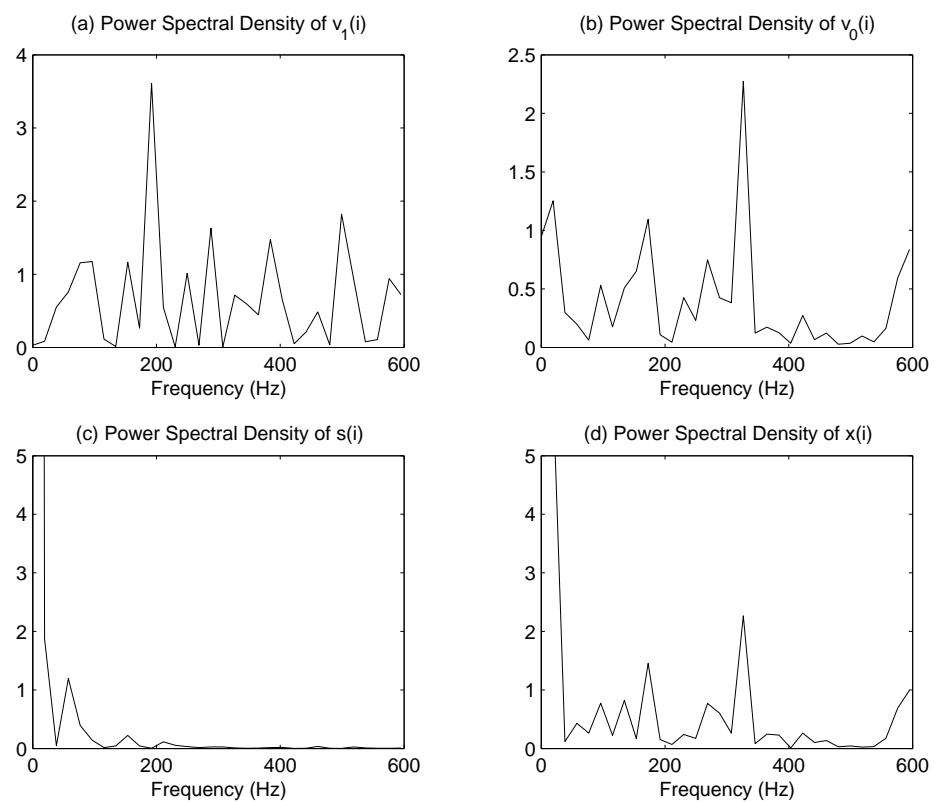


Fig. A.46: Spectral density distributions.

PDF with ANFIS are shown in Fig A.47(a) through A.47(b) respectively.

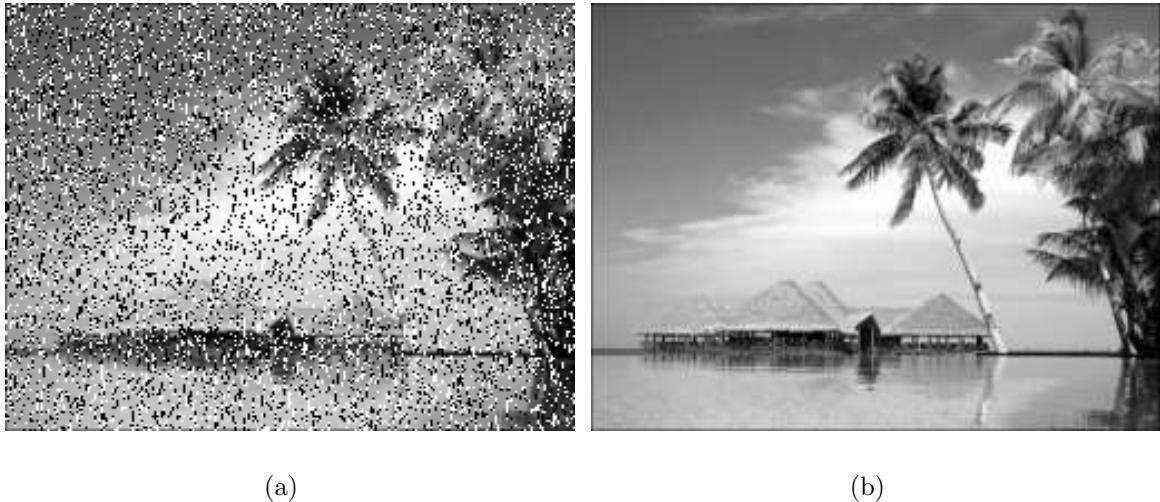


Fig. A.47: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of bell MFs. (a) The image contaminated with salt & pepper noise ($\text{PDF}=0.2793$); (b) The restored image.

The Restoration under ANFIS with the Gaussian Membership Function

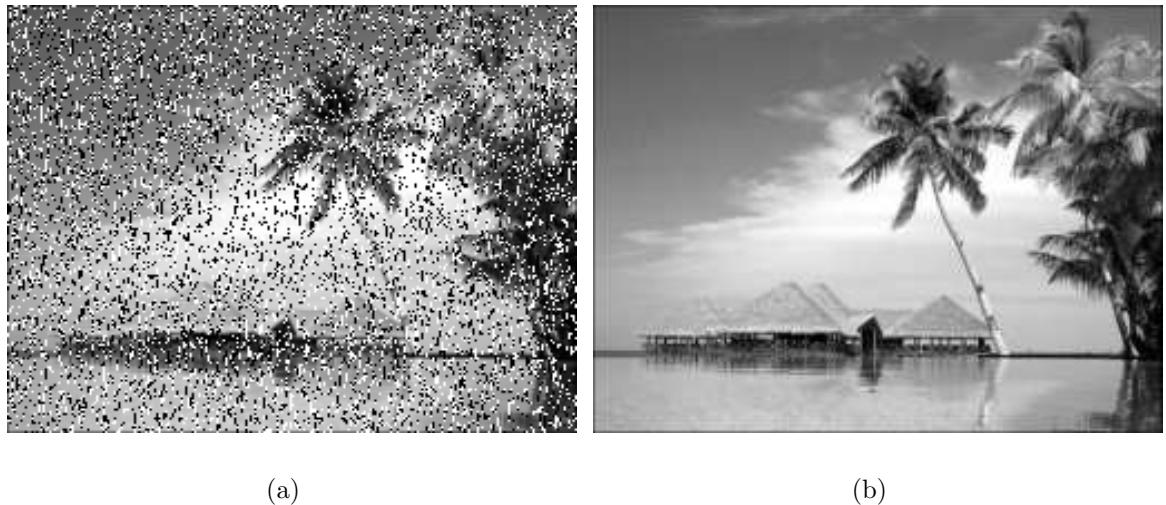
The images corrupted by the heavy salt & pepper noise and the results of removing noise with Gaussian MFs are shown in Fig A.48(a) through A.48(b) respectively.

The Restoration under ANFIS with the Triangle Membership Function

We show the images corrupted by the heavy salt & pepper noise and the results of removing noise with triangle MFs in Fig A.49(a) through A.49(b) respectively.

The Restoration under ANFIS with the Two-sided Gaussian Membership Function

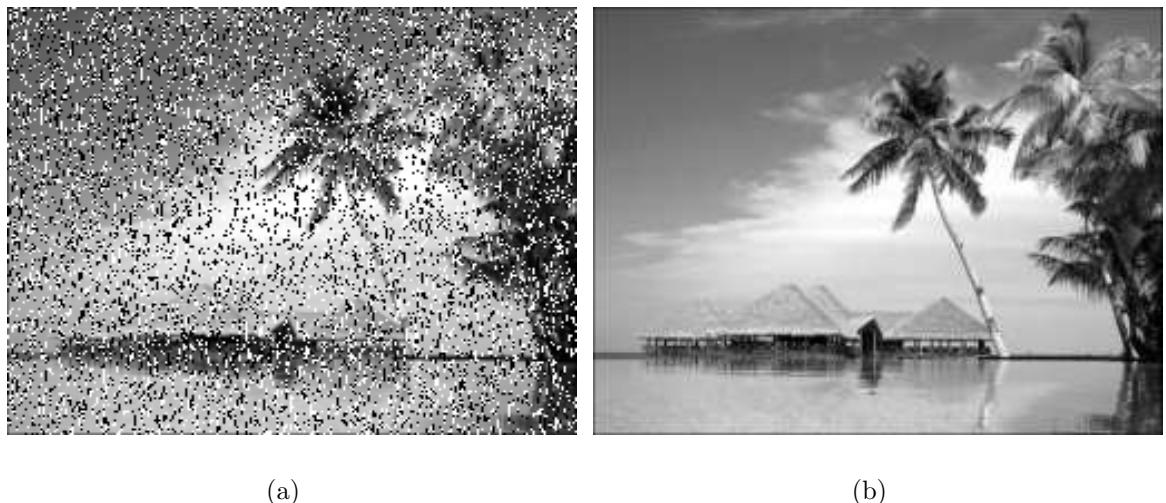
The images corrupted by the heavy salt & pepper noise and the results of removing noise with two-sided Gaussian MFs are displayed in Fig A.50(a) through A.50(b) respectively.



(a)

(b)

Fig. A.48: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of Gaussian MFs. (a) The image contaminated with salt & pepper noise ($\text{PDF}=0.2793$); (b) The restored image.



(a)

(b)

Fig. A.49: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of triangle MFs. (a) The image contaminated with salt & pepper noise ($\text{PDF}=0.2793$); (b) The restored image.

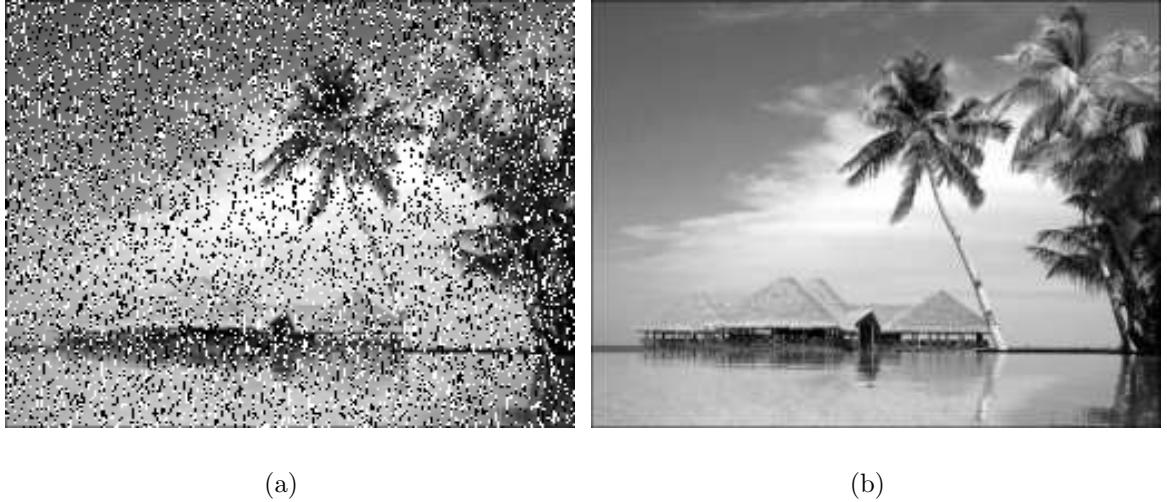


Fig. A.50: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of two-sided Gaussian MFs. (a) The image contaminated with salt & pepper noise ($\text{PDF}=0.2793$); (b) The restored image.

The Restoration under ANFIS with the Trapezoidal Membership Function

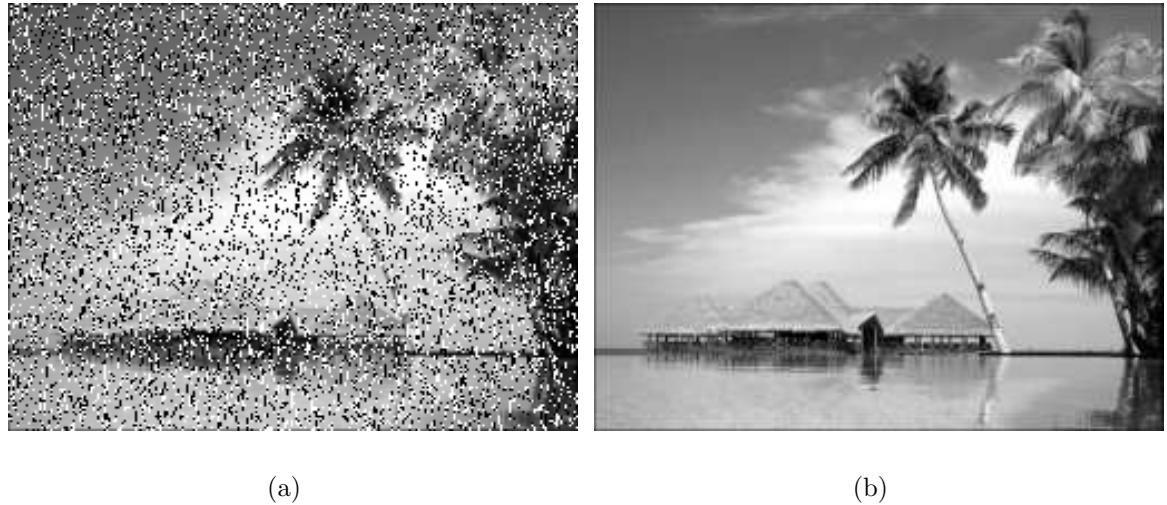
The images contaminated by the heavy salt & pepper noise and the results of removing noise with trapezoidal MFs are shown in Fig A.51(a) through A.51(b) respectively.

The Restoration under ANFIS with the Difference between Two Sigmoidal Membership Function

We show the images corrupted by the heavy salt & pepper noise and the results of removing noise with the difference between two sigmoidal MFs in Fig A.52(a) through A.52(b) respectively.

The Restoration under ANFIS with the Product of Two Sigmoidal Membership Function

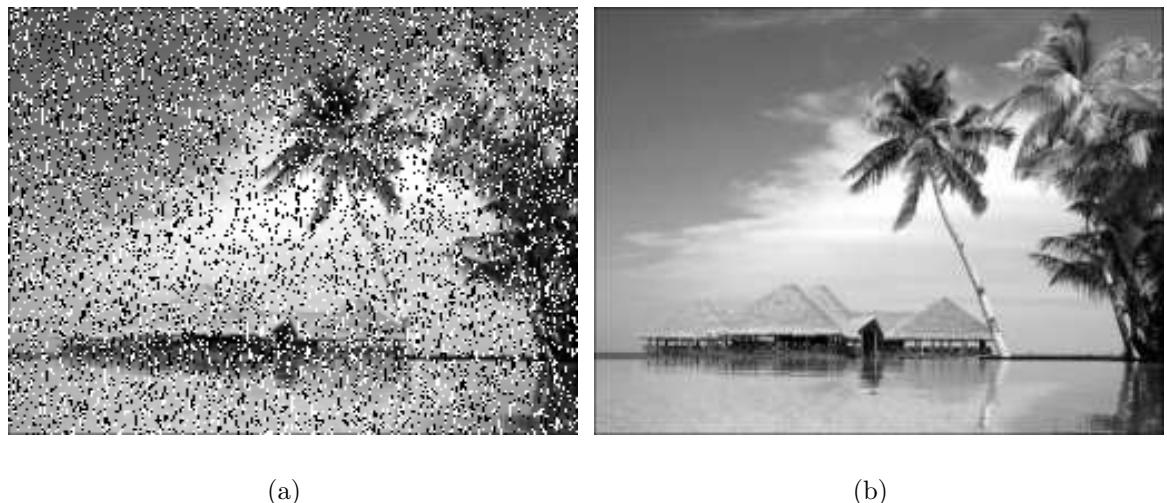
The images contaminated by the heavy salt & pepper noise and the results of removing noise with the product of two sigmoidal MFs are shown in Fig A.53(a) through A.53(b) respectively.



(a)

(b)

Fig. A.51: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of trapezoidal MFs. (a) The image contaminated with salt & pepper noise ($\text{PDF}=0.2793$); (b) The restored image.



(a)

(b)

Fig. A.52: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the difference between two sigmoidal MFs. (a) The image contaminated with salt & pepper noise ($\text{PDF}=0.2793$); (b) The restored image.



Fig. A.53: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the product of two sigmoidal MFs. (a) The image contaminated with salt & pepper noise ($\text{PDF}=0.2793$); (b) The restored image.

The Restoration under ANFIS with the Pi-shaped Membership Function

Finally the images corrupted by the heavy salt & pepper noise and the results of removing noise with the pi-shaped MFs are shown in Fig A.54(a) through A.54(b) respectively.

A.3.1 Discussion on Parameters of ANFIS

Like removing Gaussian noise, now we discuss the parameters of training epoch number in the nonlinear passive dynamics of order 3, the number membership function for each input, output membership function type, the optimization method, and the training data and checking data.

The Training Epoch Number

We set the training epoch number 50,100, 200 and 400 with the bell MFs separately to remove salt & pepper noise. The MSEs for the different epochs change particularly small.

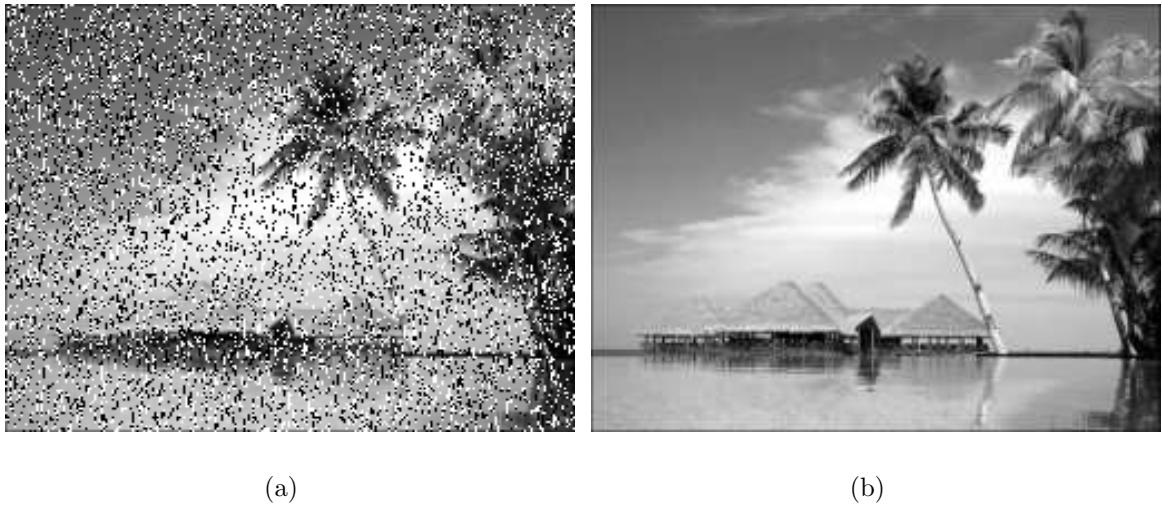


Fig. A.54: The images corrupted by salt & pepper noise and the results of removing noise with ANFIS of the pi-shaped MFs. (a) The image contaminated with salt & pepper noise ($\text{PDF}=0.2793$); (b) The restored image.

The Number of Membership Function for Each Input

Now we discuss the affection of the number of bell MFs for each input to the image restoration corrupted by salt & pepper noise. When the number of bell MFs for each input is 4, the MSE is the least.

The Optimization Method Used in Training

We use the backpropagation method compare with the combination of least-squares and backpropagation gradient descent methods in restoring the image corrupted with salt & pepper.

The restoration image with the backpropagation method compared with a hybrid learning algorithm are shown in Fig A.55(a) and A.55(b) respectively.

From Table 4.19 and Fig. A.55, we find that the effect of removing noise from the contaminated image by hybrid learning algorithm is much better than by the backpropagation method.



Fig. A.55: The restoration images corrupted by salt & pepper noise by the backpropagation method and by a hybrid learning algorithm with ANFIS of the bell MFs. (a) Backpropagation method; (b) Hybrid learning algorithm.

The Output Membership Function Type

We compare the *constant* output MF with the *linear* output MF. From Table 4.20, we find that the effect for removing noise from the contaminated image by the output MF type of *constant* is better than by the output MF type of *linear*.

A.3.2 The Training Data and Checking Data

We can split the whole data into two halves. One half is called the training data and another half the checking (validation) data. Overfitting can be detected when the checking error starts increasing while the training error is still decreasing.

We first show the RMSE curve Fig. A.56 when the image is corrupted by high noise. Overfitting can not be detected because when the training error and the checking error all oscillate. The changes of the MFs of the training data and checking data before and after training are shown in Fig. A.57.

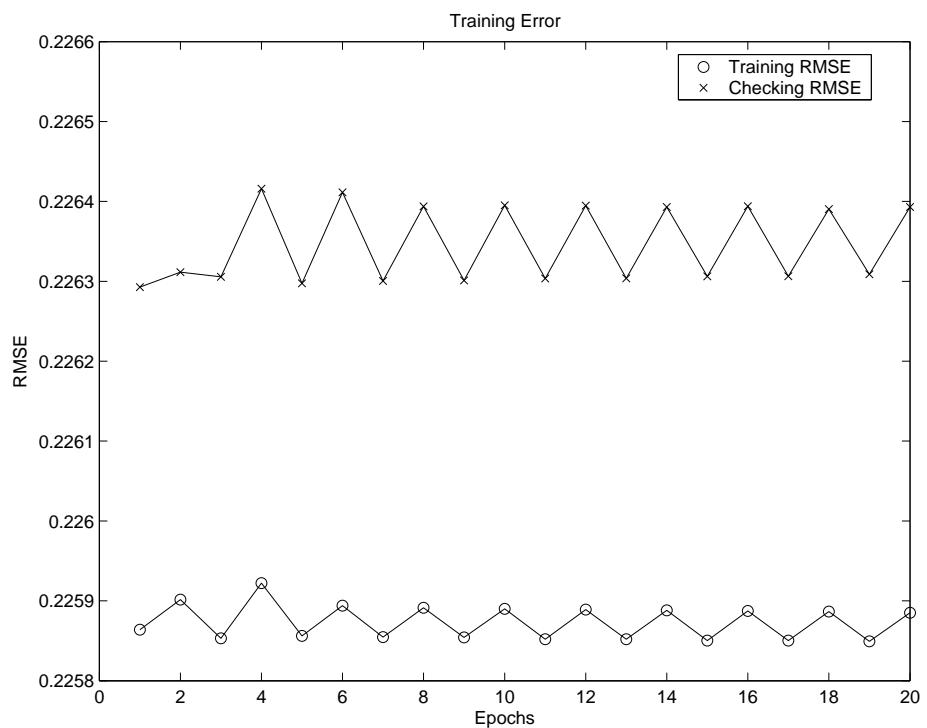


Fig. A.56: The changes of RMSEs of the training data and the checking data.

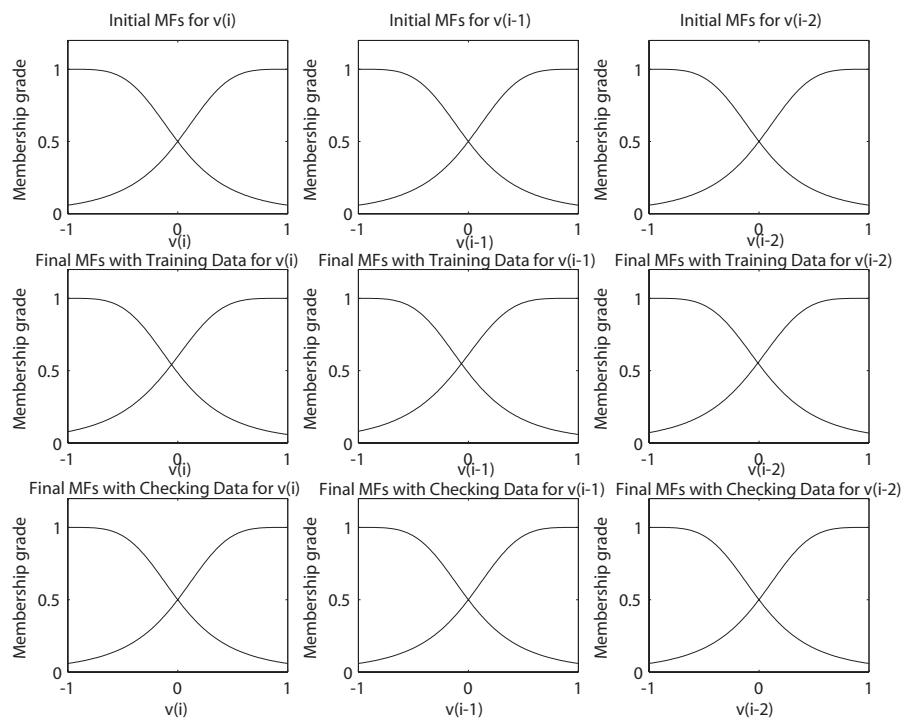


Fig. A.57: The changes of the MFs of the training data and checking data before and after training.

Appendix B—Index

- Alpha-trimmed filter, 29
- ANFIS, Adaptive network-based fuzzy inference system, 59
- ANFIS, Adaptive neuro-fuzzy inference system, 60
- Approximation, 44
- Arithmetic mean filter, 26
- BLPF, Butterworth lowpass filter, 33
- BP, Back-propagation, 62
- Contraharmonic mean filter, 27
- CWT, Continuous wavelet transform, 43
- Defuzzifier, 56
- Detail, 44
- DFT, Discrete fourier transform, 35
- Dirac Delta function, 17
- ECG, electrocardiogram, 69
- EDS, Euclidean direction search, 164
- FIR, Finite impulse response, 35
- FIS, Fuzzy inference system, 53

Fourier transform, 32
Frequency domain filtering, 32
FSM, Frequency sampling method, 32
FTM=, Frequency transformation method, 32
Fuzzy generator, 55
Fuzzy reasoning, 52
Fuzzy reasoning mechanism, 55
Fuzzy rule database, 55

Geometric mean filter, 27

Harmonic mean filter, 27

Ideal filter, 33
IDWT, Inverse discrete wavelet transform, 47
If-then rule, 52
ILPF, Ideal lowpass filter, 33
Image, 10
Image degradation, 14
Image restoration, 14

LMS, Least mean square, 164
Lowpass filter, 33
LSE, Least square estimator, 73
LSM, Least square method, 20, 62

Mamdani fuzzy inference system, 54
Max filter, 28
Mean filtering, 26
Median filter, 28

MF, Membership function, 52

Midpoint filter, 29

Min filter, 29

NN, Neural network, 59

Order-statistics filter, 28

PDF, Probability density function, 29

PID, Proportional, integral, derivative, 60

Point spread function, 18

Pure fuzzy inference system, 53

RGB, Red, Green, Blue, 78, 199, 215

RLS, Recursive least square, 164

RMSE, Root mean square error, 82

ROM, Rank order mean filter, 31

SC, Soft computing, 8

SD-ROM, Signal dependent rank order mean, 30

SNR, Signal noise ratio, 79

Spatial filtering, 26

STFT, Short time Fourier transform, 41

Sugeno FIS, 56

Wavelet analysis, 41

Wavelet decomposition tree, 46

Wavelet packet analysis, 47

Wiener filter, 38

Windowing method, 33

Index

- Alpha-trimmed filter, 29
ANFIS, Adaptive network-based fuzzy inference system, 59
ANFIS, Adaptive neuro-fuzzy inference system, 59
Approximation, 43
Arithmetic mean filter, 26
BLPF, Butterworth lowpass filter, 33
BP, Back-propagation, 62
Butterworth filter, 32
Contraharmonic mean filter, 27
CWT, Continuous wavelet transform, 42
Defuzzifier, 55
Detail, 43
DFT, Discrete fourier transform, 35
Dirac Delta function, 16
DWT, Discrete wavelet transform, 43
ECG, electrocardiogram, 70
EDS, Euclidean direction search, 179
FIR, Finite impulse response, 35
FIS, Fuzzy inference system, 52
Fourier transform, 31
Frequency domain filtering, 31
Frequency sampling method, 32
Frequency transformation method, 32
Fuzzy generator, 55
Fuzzy reasoning, 50
Fuzzy reasoning mechanism, 55
Fuzzy rule database, 54
Gaussian filter, 32
Geometric mean filter, 27
Harmonic mean filter, 27
Ideal filter, 32
IDWT, Inverse discrete wavelet transform, 45
If-then rule, 52
ILPF, Ideal lowpass filter, 32
Image, 10
Image degradation, 15
Image restoration, 15
LMS, Least mean square, 179
Lowpass filter, 32
LSE, Least square estimator, 73
LSM, Least square method, 20, 62
Mamdani fuzzy inference system, 53
Max filter, 28

Mean filtering, 26
Median filter, 28
MF, Membership function, 52
Midpoint filter, 28
Min filter, 28
NN, Neural network, 59
Order-statistics filter, 27
PDF, Probability density function, 29
PID, Proportional, integral, derivative, 60
Point spread function, 18
Pure fuzzy inference system, 52
RGB, Red, Green, Blue, 79, 135, 142, 209,
225
RLS, Recursive least square, 179
RMSE, Root mean square error, 83
ROM, Rank order mean filter, 30
SC, Soft computing, 8
SD-ROM, Signal dependent rank order mean,
30
SNR, Signal noise ratio, 80
Spatial filtering, 25
STFT, Short time Fourier transform, 41
Sugeno FIS, 55
Wavelet analysis, 41
Wavelet decomposition tree, 45
Wavelet packet analysis, 49
Wiener filter, 37
Windowing method, 32, 36

Vita

Mr. Hao Qin received his B. Eng in Electrical Engineering Department at Huazhong University of Technology in 1987.

After graduation, he worked an electrical engineering to design electrical test and control equipment in Guangzhou Apparatus Research Institute. His designing products have asynchronous dynamometer, hysteresis dynamometer, and magnetic powder dynamometer, controller and display.

Now he works on his master thesis “Image Denoising with Adaptive Neuro Fuzzy Inference Systems”.

His current interests include image processing, pattern recognition, neural networks, fuzzy, genetic algorithm and the combination of hard and soft computing in test and control system.

His mailing address is School of Engineering, University of Guelph, Albert A. Thornbrough Building, Guelph, Ontario N1G 2W1, Canada. His Email address is hqin@uoguelph.ca.