THE UNIVERSITY OF CALGARY

Adaptive Wavelet-Based Noise Filtering Techniques

by

Farshad Faghih

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF

ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA

November, 1998

Abstract

The purpose of this thesis is to investigate new and existing methods for filtering noise without blurring images. Several existing techniques are investigated and a number of new approaches introduced.

The LLMMSE (Locally Linear Minimum Mean-Squared Error) filters use the local statistics of the image in a neighbourhood around each pixel to estimate the original pixel value. Some modifications are proposed to improve the accuracy of estimating the local image statistics.

We propose a new class of wavelet-based noise filters which process the image in both spatial and scale-space domains. As part of this approach, a new probabilistic method for detecting the edges in the presence of noise is also proposed. The approaches are theoretically and experimentally compared to a number of existing wavelet-based filtering algorithms.

Finally we present a systematic approach for compensating the noise estimation bias during noise filtering, which is normally overlooked in the literature.

Acknowledgements

I am greatly indebted to my supervisor, Dr. Michael R. Smith, whose guidance was invaluable. I would like to thank him for his patience, direction and encouragement during this research. I am deeply grateful to Dr. R. M. Rangayyan for his direction and mentoring during my work on the spatial domain and adaptive neighbourhood methods. I would like to thank my fellow graduate students for their assistance and camaraderie. Most importantly, I thank my parents for their continuous and unending support.

Table of Contents

Approval Page ii

Abstract iii

Acknowledgement iv

Table of Contents v

List of Tables viii

List of Figures ix

Chapter 1: Introduction 1
1.1 Introduction and Purpose of the Thesis 1
1.2 A Possible Problem with the Existing Evaluation of Noise Filter 5
1.3 Overview of Thesis 6

Chapter 2: Spatial Domain Noise Filtering Techniques 8
2.1 Introduction 8
2.2 Evaluation of the Methods 8
2.3 Locally Linear, Minimum Mean Square Error Filters 11
2.3.1 Definition 11
2.3.2 Physical Interpretation of the LLMMSE Estimator 12
2.4 Fixed Neighbourhood LLMMSE Filters 15
2.4.1 Lee's Method 15
Implementation and Results 15
2.4.2 Kuan's Method 20
Implementation and Results 21
2.5 Advanced Neighbourhood LLMMSE Filters 21
2.5.1 Lee's Refined Filtering Method 22
Implementation and Results 24
2.5.2 Adaptive Neighbourhood (AN) Filters 24
Implementation and Results 27
2.5.3 Adjusting the AN Algorithm for Low-Contrast and Low-SNR Images 28
Improving the Accuracy of the Initial Ensemble Mean Estimation 30
Implementation and Results 32
2.6 Conclusion 36

Chapter 3: Wavelet Analysis 38
3.1 Introduction 38
3.2 Fourier Analysis 39
3.3 Short-Time Fourier Analysis 40

3.4 Wavelet Analysis 43
3.4.1 Continuous Wavelet Analysis 44
3.4.2 Discretization of the Time-Scale Parameters 45
3.4.3 Calculating the Wavelet Coefficients 47
3.5 An alternative Viewpoint: Subband Coding 48
3.5.1 Approximation and Details of a Signal 48
3.5.2 Multiple Level Decomposition 52
3.5.3 Sub-band Coding and Discrete Wavelet Analysis 52
3.6 The Two-Dimensional Fast Wavelet Transform 54
3.7 Specialised Wavelets for Image Processing 60
3.7.1 Biorthogonal Wavelet Transform 60
3.7.2 The Redundant Wavelet Transform 60
3.8 Using the Wavelets 61
3.9 Summary 62

Chapter 4: Wavelet-Based Denoising Filters **64**
4.1 Introduction 64
4.2 Basic Thresholding Method 65
4.2.1 Introduction 65
4.2.2 Hard and Soft Thresholding Methods 67
4.2.3 Implementation and Results 69
4.3 Spatially Selective Wavelet Domain Filters 76
4.3.1 Introduction 76
4.3.2 Mallat's Method 77
The Regularity Measures 78
Wavelet Transform Modulus Maxima 79
One-Dimensional Signal Denoising 80
Two-Dimensional Signal (Image) Denoising 82
4.3.3 Xu's Method 83
Implementation and Results 84
4.4 Summary 90

Chapter 5: A New Noise Filtering Technique **93**
5.1 Introduction 93
5.2 Detecting the Edges in the Spatial Domain 95
5.2.1 Derivative Operators 95
5.2.2 Extraction of Edges 97
Measure 1- Similarity in the Amplitude of the Gradient 102
Measure 2- Similarity in the Phase of the Gradient 104
Measure 3- Connectivity of the Direction Field 106
Combining the Amplitude, Phase and Direction Similarity Measures 109
5.3 Detecting the Edges in the Wavelet Domain 113
5.3.1 Introduction 113
5.3.2 Involving More Scales in the Direct Correlation Process 114
5.3.3 Introducing *a priori* Geometrical Knowledge 117

5.4 Combining the Results 119
5.5 Filtering the Noise 120
5.6 Results 123
5.7 Discussion and Conclusions 128

Chapter 6: Practical Considerations **130**
6.1 Introduction 130
6.2 Demonstrating the Presence of Noise Bias in the Performance of Wavelet Algorithms 131
6.3 How is the Noise Bias Introduced 134
6.4 Systematic Correction of Noise Bias 137
6.5 Filtering Noise from Low-Intensity Portions of the Image in the Presence of the Noise Bias 138
6.6 The Effect of Grey-Level Quantization 145
6.7 Choosing the Size and Position of the Signal-Free Region 148
6.8 Conclusion 150

Chapter 7: Conclusions and Future Considerations **152**
7.1 Conclusions 152
7.2 Future Considerations 157

Bibliography **159**

List of Tables

2.1 Quantitative measures for the results obtained by applying the LLMMSE filtering methods to a noisy synthetic image 19

2.2 Quantitative measures for the results obtained by applying the LLMMSE filtering methods to the noisy natural image 20

2.3 Quantitative measures for the results obtained by applying the LLMMSE filtering methods to a significantly degraded natural image 35

4.1 Quantitative measures for the results obtained by applying the basic thresholding method to a noisy synthetic image 73

4.2 Quantitative measures for the results obtained by applying the basic thresholding method to a noisy natural image 74

4.3 Quantitative measures for the results obtained by applying Xu's noise filtering method to a noisy synthetic image 89

4.4 Quantitative measures for the results obtained by applying Xu's noise filtering method to a noisy natural image 90

5.1 Quantitative measures for the results obtained by applying different noise filtering techniques to a noisy synthetic image 126

5.2 Quantitative measures for the results obtained by applying different noise filtering techniques to a noisy natural image 128

6.1 The effect of grey-level quantisation in removing the noise from noisy test images 148

List of Figures

2.1 Applying the LLMMSE-based filtering methods to a synthetic test image ... 16
2.2 Applying the LLMMSE-based filtering methods to a natural test image 17
2.3 Comparing LLMMSE-based noise filtering methods (magnified version of Fig. 2.2) 18
2.4 Possible refined neighbourhoods associated with a 7×7 window 23
2.5 Removing noise from a low-SNR natural test image 33
2.6 Removing noise from a low-SNR natural image (magnified version of Fig. 2.5) 34

3.1 Frequency domain signal analysis 41
3.2 Wavelet domain signal analysis 46
3.3 Filtering process required to generate the approximation and detail coefficients of a signal 49
3.4 Reducing the number of output coefficients to half by down-sampling 50
3.5 Reconstructing the original signal from its approximation and detail coefficients 51
3.6 Iterating the decomposition process for the approximation coefficients 52
3.7 Two-dimensional decomposition process at a general level 55
3.8 Two-dimensional discrete wavelet transform. The approximation and Detail images at the first decomposition level 57
3.9 The two-dimensional discrete wavelet transform. The approximation and detail images at the second decomposition level 58
3.10 Two dimensional Discrete Fourier Transform 59
3.11 Two-dimensional reconstruction process at a general level 59

4.1 Applying the basic thresholding techniques to a noisy synthetic image 70
4.2 Applying the basic thresholding techniques to a noisy natural image 75
4.3 Applying Xu's noise filtering method to the noisy synthetic image 88
4.4 Applying Xu's noise filtering method to the noisy natural image 91

5.1 The Block diagram of the proposed noise filtering technique 94
5.2 Sobel gradient masks 96
5.3 Detecting the edges of a noisy synthetic image 98
5.4 Detecting the edges of a noisy natural image 99
5.5 The fuzzy classifier used to assign a similarity measure to each candidate pixel based on the similarity in the gradient amplitude 103
5.6 The fuzzy classifier used to assign a similarity measure to each candidate pixel based on the similarity in the phase of the gradient 105
5.7 The masks defined for assigning a direction to each pixel 106
5.8 Smooth change in the phase of the gradient and direction of pixels along edges 108
5.9 The fuzzy classifier used to assign a similarity measure to each candidate pixel based on the similarity in the direction 109

5.10 Enhancing the edges in the gradient image of the synthetic test image 111
5.11 Enhancing the edges in the gradient image of the natural test image 112
5.12 The direct spatial correlation across the scales for the noisy synthetic image 115
5.13 The direct spatial correlation across the scales for the noisy natural image 118
5.14 Detecting the edges of the noisy synthetic image 121
5.15 Detecting the edges of the noisy natural image 122
5.16 Using the Edge information across the scales to preserve the edges during noise filtering 124
5.17 Comparing the performance of some noise filtering methods in removing noise from a synthetic test image 125
5.18 Comparing the performance of some noise filtering methods in removing noise from a natural test image 127

6.1 Estimating the noise parameters from uniform regions with different intensities 132
6.2 Apparent standard deviation of noise as a function of apparent image intensity 135
6.3 Apparent standard deviation of the noise as a function of apparent SNR 136
6.4 The Noise correction factor as a function of apparent SNR 137
6.5 Compensating the noise bias in Xu's and the new method 139
6.6 Filtering noise from "real" and magnitude images 142
6.7 Filtering noise from "real" and magnitude images 143
6.8 The effect of quantization error on high- and low-SNR test images 147

Chapter 1

Introduction

1.1 Introduction and Purpose of the Thesis

Noise filtering is an important step in many image processing applications. The existence of noise in an image can be the consequence of poor input sampling or interference from other sources [Bracewell95]. Although in some applications the signal-to-noise ratio of the scanned images can be improved by using more advanced equipment, this does not always hold. In aerial imaging for instance, noise in the visible channel due to clouds and fog can only be suppressed by post-processing techniques [Malfait97].

The key function of noise filtering techniques is to improve the image in ways that increase the chances for success of subsequent processing steps [Gonzalez92]. By filtering the noise, we can facilitate extracting information from the image. A good noise filtering technique also allows us to spend less on the imager hardware and still have images with acceptable quality.

The purpose of this thesis is to examine ways to enhance the quality of noisy images. Noise smoothing and edge enhancement are inherently conflicting processes. Smoothing a region might destroy edges, which carry much of the image information. However, sharpening edges might lead to increased noise. In this thesis we will examine different methods for preserving edges while suppressing the noise.

Noise filtering techniques can process the image either in the spatial or other domains. The spatial domain refers to the image plane itself, and approaches in this category are based on direct manipulation of the pixels in the image. The other techniques are based on modifying the coefficients in a transform of the image.

One of the simplest and yet effective methods for removing the noise is image averaging [Papoulis65, Gonzalez92]. In many applications, noise is often removed by averaging over many identical image acquisitions. For example in magnetic resonance imaging (MRI), averaging the images can remove breathing artefacts [Wood86]. The feasibility of the averaging method, which is considered a spatial domain technique, depends on the image acquisition time and cost.

Early techniques in noise filtering concentrated mostly on procedures that were carried out computationally in the frequency domain. In these approaches, the images are assumed to be stationary, which means all pixels in an image have similar statistics. Almost all practical images violate the basic assumption of data stationary required by these techniques. Images are typically only quasi-stationary. Moreover, the intensity of a pixel is somewhat correlated with the value of pixels in a small neighbourhood around it [Chan85, Paranjape94a]. Methods that do not take this characteristic into account cannot effectively adjust themselves to the local features of an image. As a result, such methods generally smooth the edges and other structured features of the image [Mahesh90].

The least mean squares (Wiener) filter, as a frequency domain technique, assumes that good statistical models of the image and noise are available [Gonzalez92]. The noise spectrum can often be estimated from a signal free region of the image or in a high-frequency band. However, there is often a severe difficulty in obtaining a good estimate of

the image power spectrum from the degraded data. In particular, the high-frequency components of the spectrum are often lost or inaccurately estimated. Moreover, the noise spectrum obtained from a signal free region of the image may be distorted [Henkleman86, McGibney93].

At very low signal-to-noise ratios (SNR), the Wiener filter tends to reject almost all high-frequency components including those from important edge features, and hence the restored image is fairly smooth. While this smoothness leads to good improvement in objective metrics, such as improvement in the mean squared error (MSE), it does not necessarily produce the best restoration from a perceptual viewpoint.

Early examples of spatial domain techniques are the median filter and its variants, averaging filter, sigma filter, and box filter [Yung96]. These filters are mostly designed for removing a specific type of noise distribution. For instance, the median filter is designed to remove impulsive noise, while the sigma filter is designed to suppress Gaussian noise. A common characteristic of these methods is that they process all pixels in a similar manner, ignoring all local characteristics [Yung96]. Although this characteristic has been proven effective in removing additive random noise, it can blur the image. Such distortion may be unacceptable as it can reduce the sharpness of lines and edges. Obviously, to keep the image sharp, we need to distinguish noisy pixels from edge pixels and process each group differently.

Due to this feature-preserving requirement, another class of filter algorithms emerges, with the aim of preserving the image features while possessing an effective noise removal capability. These filters are based on some techniques to identify and preserve the edge pixels during the noise smoothening process.

In this thesis, two different classes of spatially-selective filters are discussed. The filters are sensitive to the local features of an image: they can detect edges and pass the associated high-frequency data. The first group of filters discussed in this thesis is the class of adaptive neighbourhood filters [Lee81, Paranjape94, Das97, Rangayyan98b]. In these techniques, the local behaviour of the signal and noise in a neighbourhood of each pixel is used to estimate the original pixel value. In adaptive neighbourhood techniques, unlike fixed neighbourhood methods, the shape (and size) of the selected neighbourhood is adapted to the local features of the image. We will discuss an extension to Das's work to include a more realistic definition of neighbourhoods [Rangayyan98b]. New work extending these concepts to low SNR images is also presented.

The second group of filters discussed in this thesis is wavelet-based filters. Wavelet transforms provide multi-resolution representation of signals and images: they decompose signals and images into multi-scale details. The basis functions used in wavelet transforms are locally supported: they are non-zero only over a part of the domain represented. Sharp transitions in images are preserved and depicted extremely well in wavelet expansions [Mallat89a]. This special treatment of edges by wavelet transforms is very attractive in image filtering.

A number of wavelet-based denoising techniques have been developed during recent years [Weaver91, Mallat92, Xu94, Malfait97]. A typical noise-filtering algorithm is composed of three steps: computing the wavelet transform of the signal, manipulating the wavelet coefficients, and finally reconstructing the filtered signal from the modified coefficients. The difference between various denoising methods is mainly in the second

step, in the way they modify the wavelet coefficients. Several wavelet-based filtering techniques are discussed in this thesis.

We shall discuss a new wavelet-based algorithm to suppress the noise without blurring edges. The proposed method uses a matched filter determined from edge information obtained by a combination of spatial and wavelet domain information. The wavelet domain analysis uses a new compression technique to highlight weak edges.

1.2 A Possible Problem with the Existing Evaluation of Noise Filters

In the following chapters, some commonly-used noise-filtering techniques are discussed and re-implemented. The results are compared to those of the proposed noise-filtering algorithm.

A key feature of all algorithms in the study is the determination of the noise level. The approach to determining the noise level is not well defined in the literature. Explanations are vague in term of: "In MR images there are almost always lines at the edge of the image that contain no signal, only noise. We use these lines to characterise the noise in the image." [Weaver91] or "In digital image processing, one can often use the background noise at the dark (signal-free) regions near the boundaries of an image as the reference noise" [Xu94].

From the work of Henkelman [Henkelman86] and McGibney *et al.* [McGibney93] it is clear that determination of the noise level from the signal-free regions of the image is not straightforward which indicates a possible bias in the results reported by other researchers. In the first part of this thesis we will use a theoretical approach to avoid the bias in

estimating the noise. This approach allows us to concentrate on the algorithms. Later, we will consider the problems associated with estimating the noise in more detail.

1.3 Overview of the Thesis

The organisation of this thesis is as follows. In Chapter 2, we review spatial domain adaptive noise-filtering techniques. This group of filters adapts to local changes in image statistics. A new adaptive approach (published as [Rangayyan98b]) will be explained. In the last section of Chapter 2, a further extension of this work to filter noise from low-SNR images is proposed.

In Chapter 3, we briefly review some aspects of wavelet theory that is necessary to understand the methods discussed in the rest of this thesis. A brief explanation of the wavelet transform along with a standard filter-bank implementation of the transform are given.

In Chapter 4, some commonly-used wavelet-based noise filtering techniques are discussed. We used both quantitative and qualitative measures to compare the performances of the methods. In Chapter 5, a new method for suppressing noise from images is proposed where both spatial and wavelet domain techniques are exploited. In the last part of this chapter, the performance of the new method is compared with that of other methods discussed in the thesis.

In Chapter 6, we discuss some practical applications, which are:

- Appropriate techniques to determine the noise level.
- Effect of image amplitude quantization on algorithm stability.

- The special case of MR imaging where processing can also be performed in the complex image domain with characteristics different from those of images normally processed.

The conclusions drawn from the research carried out in this thesis as well as suggestions for future work are presented in Chapter 7.

Chapter 2

Spatial Domain Noise Filtering Techniques

2.1 Introduction

In this chapter, we will review a class of spatial domain filters that make use of local statistics of the image to adapt to the non-stationary characteristics of the image. All the noise-smoothing algorithms discussed in this chapter are based on locally linear, minimum mean-squared error (LLMMSE) estimation of signals [Kuan82]. The LLMMSE estimator has the property that noise in flat regions of the image are smoothed while observations are left unchanged in the vicinity of edges [Jiang86].

To remove the noise more effectively from edge areas, several modifications to the basic LLMMSE method have been proposed by other authors [Lee80, Lee81, Rajala81, Kuan85, Jiang86, Paranjape94, Das97, Rangayyan98b]. We will review and simulate some of these techniques in this chapter. We will then propose a new filtering algorithm that is a modification of a recently published work by Rangayyan, Ciuc, and Faghih [Rangayyan98b]. The procedure used to evaluate the performance of the different noise filtering methods will also be explained.

2.2 Evaluation of the Methods

To evaluate the performance of filtering methods, we apply them to noisy images and examine the results. One approach to assessing the quality of the filtered images is to

compare them with the original, clean images. Using naturally-degraded images to test a method has the drawback that the original, clean images are not accessible. Therefore, to generate a noisy image, we usually start with a clean test image and add synthetic Gaussian white noise to it.

As we will see in Chapter 6, the measured noise level of a magnitude image is a function of the background intensity level [Henkelman86, McGibney93]. This bias in estimating the noise appears to be either ignored or to be compensated for in an *ad hoc* manner in the literature [Lee80, Lee81, Rajala81, Kuan85, Jiang86, Weaver91, Xu94]. The bias can be removed by estimating the noise from a medium-intensity, uniform region artificially added to a corner of the image, thereby allowing us to concentrate on the problems associated with the algorithms themselves. In Chapter 6, we shall explain how to properly estimate the noise level in practical situations. The sensitivity of the algorithm to inaccurate estimation of the noise will also be examined at that time.

In Chapter 6, we also consider the effect of amplitude quantization on the stability of our algorithm. However, for the initial chapters of the thesis, we ignore problems associated with a limited number of image grey levels.

Both quantitative measures and qualitative comparisons are used to evaluate the quality of the restored images. The visual effect associated with sharpness of edges and the amount of noise remaining in the output image are used to compare the quality of the images.

As quantitative global measures of objective improvement of the results, we refer to both the mean squared error (MSE) given by:

$$MSE = \frac{1}{N^2}\sum_{m,n}[\hat{f}(m,n) - f(m,n)]^2 ,$$

and the improvement in the SNR which is defined by Banham *et al.* [Banham96] and Malfait *et al.* [Malfait97] as:

$$ISNR = SNR_{output} - SNR_{input} = 10 \ \log_{10} \frac{P_{input-noise}}{P_{output-noise}} .$$

or:

$$ISNR = 10 \ \log_{10}\left[\frac{\sum_{m,n}[g(m,n) - f(m,n)]^2}{\sum_{m,n}[\hat{f}(m,n) - f(m,n)]^2}\right],$$

where $f(m,n)$ is the intensity of the original undegraded image at location (m,n), and $g(m,n)$ and $\hat{f}(m,n)$ are the intensities of the degraded and filtered images at (m,n), respectively.

In practice, we generate several versions of each noisy image to improve the accuracy of the quantitative measures. For this purpose, we run our noise generator with different seed values. The result will be test images with the same noise power but different noise patterns. After filtering the noise from each test image, the quantitative measures are calculated for all the resulting images. The average and the standard deviation of the quantitative measures are presented as more accurate measures for evaluating the performance of different methods.

The quantitative measures reflect the global effects of the filter. However they cannot take into account all aspects of image quality for which the human eye is sensitive. Therefore, quantitative measures should always be used in conjunction with qualitative measures.

2.3 Locally Linear, Minimum Mean Squared Error (LLMMSE) Filters

2.3.1 Definition

A noisy image can be represented by:

$$g(i,j) = f(i,j) + n(i,j) \qquad i = 0, 1, \ldots, N\text{-}1 \quad j = 0, 1, \ldots, M\text{-}1,$$

where $f(i,j)$ is the original, clean image (unknown) and $g(i,j)$ is the known noisy image. $n(i,j)$ is the additive noise component which is assumed to be a zero-mean, Gaussian function of variance σ_n^2. Moreover, the noise is assumed to be uncorrelated with the original pixel value, $f(i,j)$. To restore the image, we need to find an estimate of $f(i,j)$ based on the given noisy image and some statistical parameters of the noise. The LLMMSE estimate, $\hat{f}_{LLMMSE}(i,j)$, of the undegraded image, $f(i,j)$, is defined as [Kuan85]:

$$\hat{f}_{LLMMSE}(i,j) = E[f(i,j)] + \frac{\sigma_f^2(i,j)}{\sigma_f^2(i,j) + \sigma_n^2(i,j)} \{g(i,j) - E[g(i,j)]\}, \qquad (2.1)$$

where $E[f(i,j)]$, $E[g(i,j)]$, $\sigma_f^2(i,j)$, and $\sigma_g^2(i,j)$ are the ensemble mean and ensemble variance of $f(i,j)$ and $g(i,j)$, respectively. Usually, the ensemble statistics associated with the original image, $f(i,j)$, are not available *a priori* and must be estimated from the noisy image. For additive zero-mean Gaussian noise, we can write [Lee80]:

$$E[f(i,j)] = E[g(i,j)] \qquad (2.2a)$$

$$\sigma_f^2(i,j) = \sigma_g^2(i,j) - \sigma_n^2(i,j). \qquad (2.2b)$$

To calculate the ensemble statistics of the degraded image, we need several images of the same scene (an ensemble) which is not usually available. In most practical cases, we

can find a small neighbourhood around each pixel in which pixels have the same statistical parameters. Therefore we can estimate the ensemble mean and ensemble variance of a pixel from the local mean and local variance of all pixels in a proper neighbourhood around it. Using local statistics instead of ensemble statistics, we can rewrite Eqns. 2.2(a) and 2.2(b) as:

$$m_f(i,j) = m_g(i,j) \qquad (2.3a)$$

$$v_f(i,j) = v_g(i,j) - \sigma_n^2(i,j), \qquad (2.3b)$$

where $m_f(i,j)$, $m_g(i,j)$, $v_f(i,j)$, and $v_g(i,j)$ are the local mean and local variance of the original and noisy images. If we substitute the ensemble statistics by local statistics of the original clean image in Eqn. 2.1, the result is:

$$\hat{f}_{LLMMSE}(i,j) = m_f(i,j) + \frac{v_f(i,j)}{v_f(i,j) + \sigma_n^2(i,j)}(g(i,j) - m_g(i,j)).$$

After substituting the unknown original image statistics with those known from the degraded image we obtain the final form of the LLMMSE estimator [Kuan85]:

$$\hat{f}_{LLMMSE}(i,j) = m_g(i,j) + \frac{v_g(i,j) - \sigma_n^2(i,j)}{v_g(i,j)}(g(i,j) - m_g(i,j)). \qquad (2.4)$$

2.3.2 Physical Interpretation of the LLMMSE Estimator

Human vision is generally assumed to be more sensitive to noise in flat areas than in edge areas [Lee81]. Therefore, removing the noise from flat areas and leaving the edge areas unchanged will improve the subjective quality of the image without blurring the edges.

We can show that this noise removal property is associated with the LLMMSE estimator. For this purpose, we rewrite the estimated value for each pixel (Eqn. 2.4) as a sum which weights the contribution from the corrupted pixel value, $g(i,j)$, and the average pixel values in a neighbourhood around it, $m_g(i,j)$:

$$\hat{f}_{LLMMSE}(i,j) = W(i,j)g(i,j) + (1 - W(i,j))m_g(i,j), \qquad (2.5)$$

where

$$W(i,j) = \frac{v_g(i,j) - \sigma_n^2(i,j)}{v_g(i,j)} = \frac{v_f(i,j)}{v_f(i,j) + \sigma_n^2(i,j)}. \qquad (2.6)$$

In flat image regions, the local variance of the original image will be close to zero ($v_f(i,j) \cong 0$) so that the contribution from a single noisy pixel is reduced, i.e., $W(i,j) \cong 0$. As a result the LLMMSE estimator (Eqn. 2.5) replaces the noisy pixel with the average of pixels in its neighbourhood:

$$\hat{f}_{LLMMSE}(i,j) \cong m_g(i,j).$$

In the vicinity of edges, the local variance of the original image is much larger than the variance of the noise ($v_f(i,j) >> \sigma_n^2(i,j)$) and the weighting factor becomes closer to unity, $W(i,j) \cong 1$. As a result, the LLMMSE estimator leaves the pixels unchanged:

$$\hat{f}_{LLMMSE}(i,j) \cong g(i,j).$$

In other parts of the image, the filter's behaviour is between these two extreme cases.

In the LLMMSE method, the quality of the restored image strongly depends on the shape and size of the neighbourhood defined around each pixel. For the LLMMSE method

to be accurate, local statistics should be a good approximation to ensemble statistics. Later in this chapter we shall discuss methods to choose the neighbourhood appropriately.

Several methods have been proposed in the literature to implement the LLMMSE estimator. The techniques can be classified in two main groups:

The fixed neighbourhood filter implementations include:

- Lee's method in which the local parameters are calculated within a fixed rectangular window around each pixel [Lee80].
- Kuan's method in which the local rectangular window is not uniformly weighted; pixels that are closer to the pixel under consideration have heavier weightings [Kuan85].

The advanced neighbourhood implementations include:

- Lee's refined method in which the most suitable neighbourhood is selected among a finite set of (fixed-size) pre-defined neighbourhoods [Lee81].
- The adaptive neighbourhood method in which both the shape and the size of the neighbourhood are adapted to the local features of the image [Paranjape94, Das97, Rangayyan98b].

In the rest of this chapter, we will review the above-listed techniques in more detail and discuss our modification to the adaptive neighbourhood method of Rangayyan *et al.* [Rangayyan98b].

2.4 Fixed Neighbourhood LLMMSE Filters

2.4.1 Lee's Method

Lee used a fixed-size rectangular window to calculate the local statistics of the image [Lee80]. For a window of size $(2n+1)\times(2m+1)$, centred at (i,j), the local mean is:

$$m_g(i,j) = \frac{1}{(2n+1)(2m+1)} \sum_{k=i-n}^{i+n} \sum_{l=j-m}^{j+m} g(k,l) \qquad (2.7)$$

with the local variance given by:

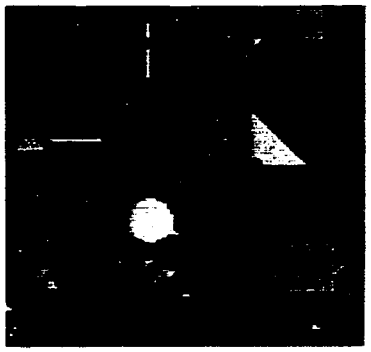
$$v_g(i,j) = \frac{1}{(2n+1)(2m+1)} \sum_{k=i-n}^{i+n} \sum_{l=j-m}^{j+m} \left[g(k,l) - m_g(i,j)\right]^2 . \qquad (2.8)$$

After calculating the local mean and variance, Lee used the LLMMSE estimator (Eqn. 2.4) to restore the noisy image. The quality of the restored image strongly depends on the size of the fixed rectangular window. Using small windows will reduce the noise smoothing effect of the filter, while using large windows will blur edges. Lee [Lee80] used 7×7 windows (i.e. $m = n = 3$) to calculate the local statistics.

Implementation and Results

To compare the performance of Lee's filter with the other techniques discussed in this thesis, we implemented the algorithm using 7×7 fixed-size windows to calculate the local statistics.

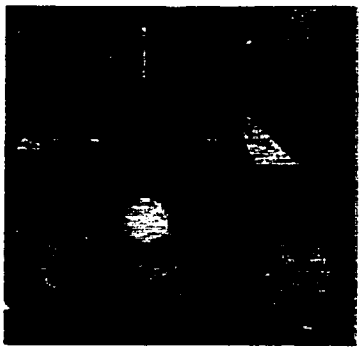
As discussed earlier, we avoid the bias in estimating the noise by using a uniform region added to the image prior to introduction of the noise. The results obtained from this simulation of Lee's work are shown in Figs. 2.1 and 2.2 (or equivalently in Fig. 2.3, which gives a magnified version of the images shown in Fig. 2.2). Fig. 2.1(a) shows the original



(a)- Original

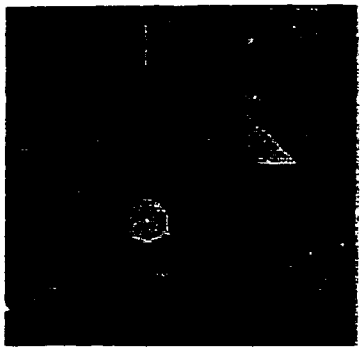
(b)- Noisy

(c)- Lee's Method



(d)- Kuan's Method

(e)- Lee's Refined Method



(f)- Adaptive Neighbourhood

Figure 2.1 *Applying the LLMMSE-based filtering methods to a synthetic test image. (a) Original, synthetic 128×128 image with maximum grey level of 255. (b) After adding non-quantized Gaussian noise with variance 400 (grey level)^2. Results of: (c) Lee's method. (d) Kuan's method. (e) Lee's refined filtering method. (f) Adaptive neighbourhood method.*

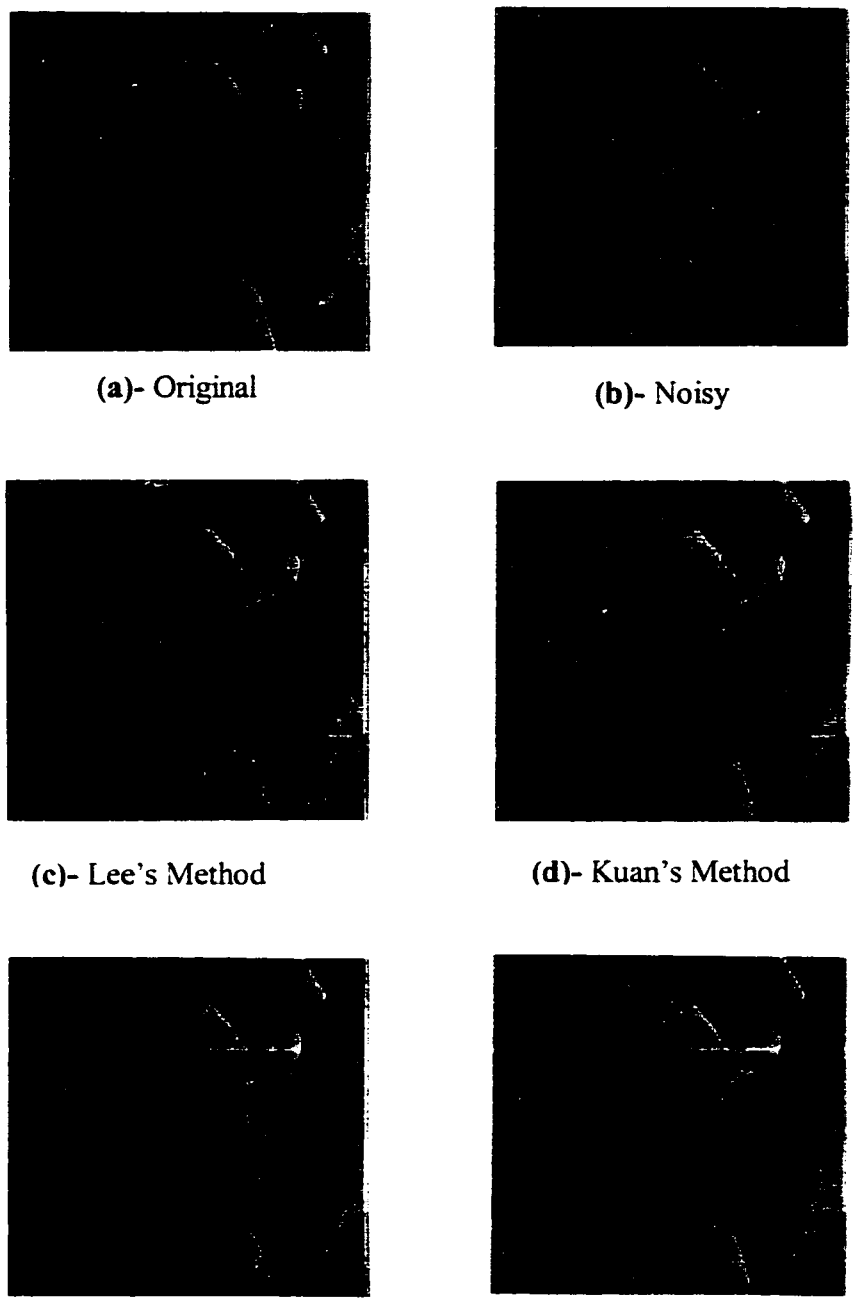


Figure 2.2 *Applying the LLMMSE-based filtering methods to a natural test image. (a) Original 256×256 image with maximum grey level of 255. (b) After adding non-quantized Gaussian noise with variance 400 (grey level)^2. Results of: (c) Lee's method. (d) Kuan's method. (e) Lee's refined filtering method. (f) Adaptive neighbourhood method.*

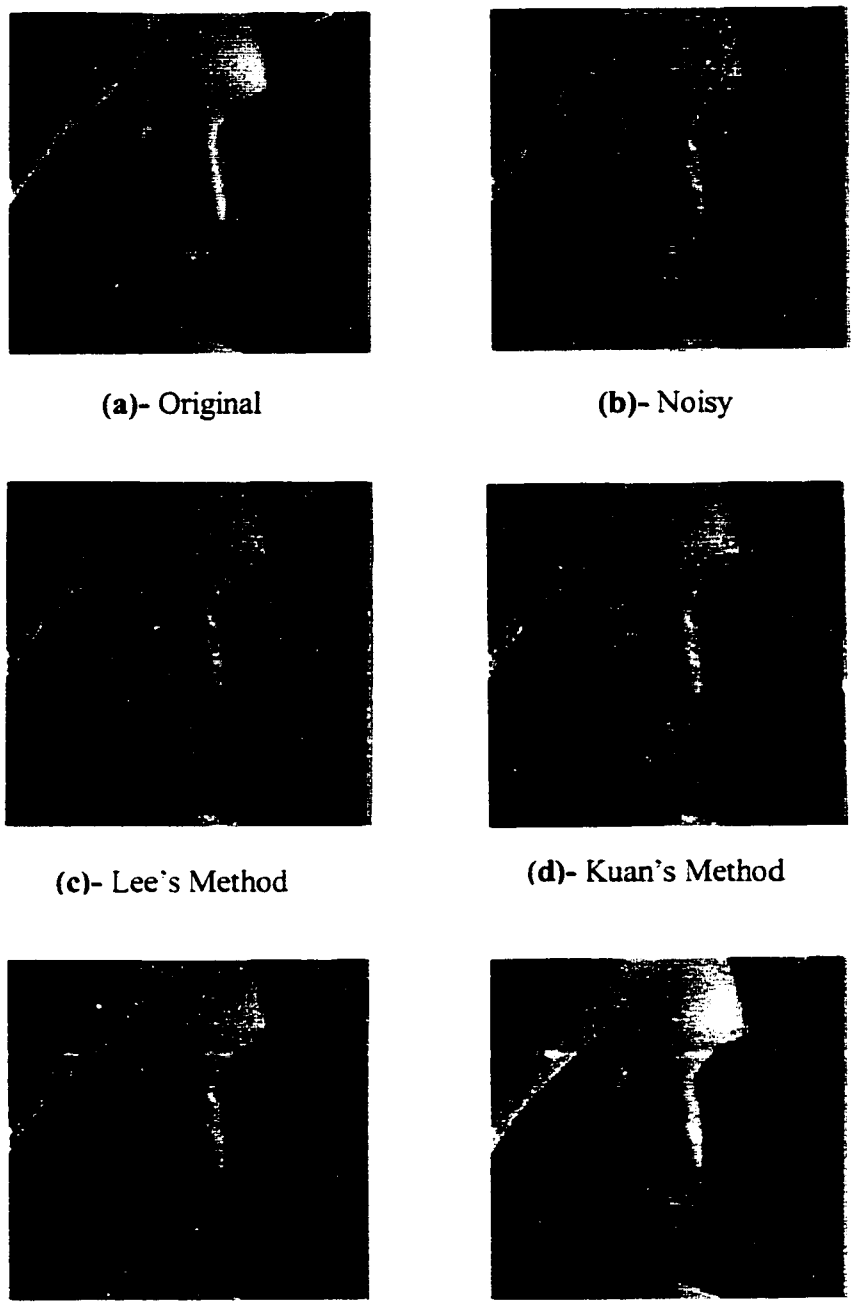


Figure 2.3 *(Magnified Version of Fig. 2.2) Comparing LLMMSE-based noise filtering methods. (a) Original image. (b) After adding Gaussian noise. Results of: (c) Lee's method. (d) Kuan's method. (e) Lee's refined filtering method. (f) Adaptive neighbourhood method.*

test image, which is a 128×128 synthetic image. Fig. 2.1(b) shows the result of adding a significant amount of Gaussian noise with variance 400 (grey level)^2 to the test image. The RMS (Root Mean Squared) difference between the original image and the noisy image is 19.82.

The result of applying Lee's method to the test image is shown in Fig. 2.1(c). Noise is removed effectively from the smooth areas, but the image remains noisy in the vicinity of edges. To evaluate the restored image quantitatively, the MSE between the original and the restored image and ISNR were calculated. We compare these quantitative measures from Lee's method with those of the other approaches in Table 2.1.

Figure	*Description*	*MSE (Grey level)^2*	*ISNR (dB)*
2.1(b)	Additive noise, σ^2 = 400 (grey level)^2	393.0 ± 1.7	0.00 ± 0.00
2.1(c)	Lee's method	105.2 ± 0.6	5.73 ± 0.03
2.1(d)	Kuan's method	107.0 ± 0.5	5.66 ± 0.03
2.1(e)	Lee's refined method	77.3 ± 0.7	7.06 ± 0.04
2.1(f)	Adaptive neighbourhood method	74.7 ± 0.4	7.21 ± 0.02

Table 2.1 *Quantitative measures (MSE and ISNR) for the results obtained by applying the LLMMSE filtering methods to the noisy synthetic image (Fig. 2.1(b)).*

In the example in Fig. 2.2 (or equivalently Fig. 2.3), we use a natural test image (Fig. 2.2(a)) to evaluate Lee's method. Fig. 2.2(b) shows the result of adding Gaussian noise with variance 400 (grey level)^2 to the test image. The RMS difference between the original image and the noisy image is 19.60. The result of applying Lee's method to the image is shown in Fig. 2.2(c). Again, the presence of noise in the vicinity of edges is noticeable. Table 2.2 compares the quantitative measures from Lee's method with those of the other approaches.

Figure	*Description*	*MSE (Grey level)²*	*ISNR (dB)*
2.2(b)	Additive noise, σ^2 = 400 (grey level)^2	384.0 ± 1.7	0.00 ± 0.00
2.2(c)	Lee's method	112.2 ± 0.5	5.34 ± 0.02
2.2(d)	Kuan's method	105.3 ± 0.4	5.62 ± 0.02
2.2(e)	Lee's refined method	99.6 ± 0.4	5.86 ± 0.02
2.2(f)	Adaptive neighbourhood method	79.4 ± 0.3	6.85 ± 0.03

Table 2.2 *Quantitative measures (MSE and ISNR) for the results obtained by applying the LLMMSE filtering methods to the noisy natural image (Fig. 2.2(b)).*

2.4.2 Kuan's Method

Kuan *et al.* also used a fixed-sized rectangular window to compute local statistics [Kuan85]. The local mean is calculated using Lee's approach. Kuan's definition for local variance is different as a multiplicative factor is used to put more weight on the pixels that are closer to the pixel under consideration:

$$v_g(i,j) = \frac{1}{(2m+1)(2n+1)} \sum_{k=i-m}^{i+m} \sum_{l=j-n}^{j+n} c(i-k, j-l)[g(k,l) - m_g(k,l)]^2 , \qquad (2.9)$$

where $c(i,j)$ is a weighting function which has its highest value at the centre of the window (i.e. at $(k,l) = (0,0)$). In addition to introducing the weighting function, Kuan *et al.* allowed the local mean to vary within the window for calculation of the local variance. The above definition of local variance is claimed [Kuan82] to be more robust and less sensitive to noise than Lee's definition.

Implementation and Results

In our implementation of Kuan's work, we used 7×7 fixed-size windows to calculate the local statistics. The weighting function, $c(i,j)$, was given the Gaussian shape proposed by Kuan *et al.*.

Fig. 2.1(d) shows the result obtained by applying the Kuan filter to the synthetic noisy image of Fig. 2.1(b). By comparison to the result of Lee's method (Fig. 2.1(c)), Kuan's method has removed the noise more effectively from the vicinity of edges. However, the presence of noise in the neighbourhood of edges is still noticeable, as is some edge blurring. Table 2.1 compares the quantitative measures from Kuan's method with those of the other approaches.

Fig. 2.2(d) shows the result obtained by applying Kuan's method to the natural test image in Fig. 2.2(a). A quantitative comparison of the results with other methods is given in Table 2.2. The results obtained using Kuan's method are slightly better than those obtained using Lee's method (both quantitatively and qualitatively). In particular, the noise is removed more effectively from the neighbourhood of edges. However, it is still desirable to reduce noise in edge areas without blurring the image.

2.5 Advanced Neighbourhood LLMMSE Filters

The fixed-neighbourhood methods discussed so far have a common drawback. They do not remove the noise from the vicinity of edges. Although the human vision is more sensitive to the noise in the flat areas, it is still desirable to remove the noise from the edges; however, this must be achieved without blurring the edges.

As we saw in Section 2.3.2, the LLMMSE estimation is a weighted average of the local mean and the image itself. If the local variance of the pixels in a neighbourhood is small, the LLMMSE algorithm puts more weight on the average of the pixels and hence smoothes the noise. In other words, the LLMMSE filter is effective only if the local neighbourhood is homogeneous. Near the edges, a fixed rectangular neighbourhood, such as those used in Lee's or Kuan's methods, can easily include pixels from different regions (i.e., regions on the two sides of an edge) and hence cause edge blurring. By using more sophisticated neighbourhoods, we can limit the neighbourhood to more homogeneous regions.

2.5.1 Lee's Refined Filtering Method

The algorithm discussed in this section is an extension made by Lee [Lee81] to his former method [Lee80]. This approach uses Lee's original definition of neighbourhoods in flat regions. However, neighbourhood definitions are adapted in the edge areas to take into account edge orientation.

To detect the edge areas, a threshold is set up for the local variance and only those pixels with local variance exceeding this threshold are processed with the refined algorithm. The local mean and variance are first calculated in a 7×7 neighbourhood. If the local variance is greater than the established threshold, the pixel is assumed to be near an edge. Since Lee did not state [Lee81] how to calculate the threshold we assumed that the threshold of the local image variance, T, was given by:

$$T = k\sigma_u^2,$$

where σ_n^2 is the unbiased variance of noise determined from a known flat region. When the local image variance was below this level, we applied Lee's original approach and replaced the pixel by the standard LLMMSE estimates.

The refined method is applied to the pixels recognised as belonging to an edge. The algorithm uses directional gradient operators to implicitly generate two areas in the 7×7 neighbourhood: A homogeneous area whose amplitude is similar to the pixel under consideration and another area (the "other side of the edge") whose amplitude is dissimilar to the considered pixel.

An attempt is made to generate a reduced neighbourhood associated with the homogeneous area with amplitude similar to the considered pixel. Fig. 2.4 shows the eight possible reduced neighbourhoods (subsets) associated with a 7×7 window centred about the considered pixel.

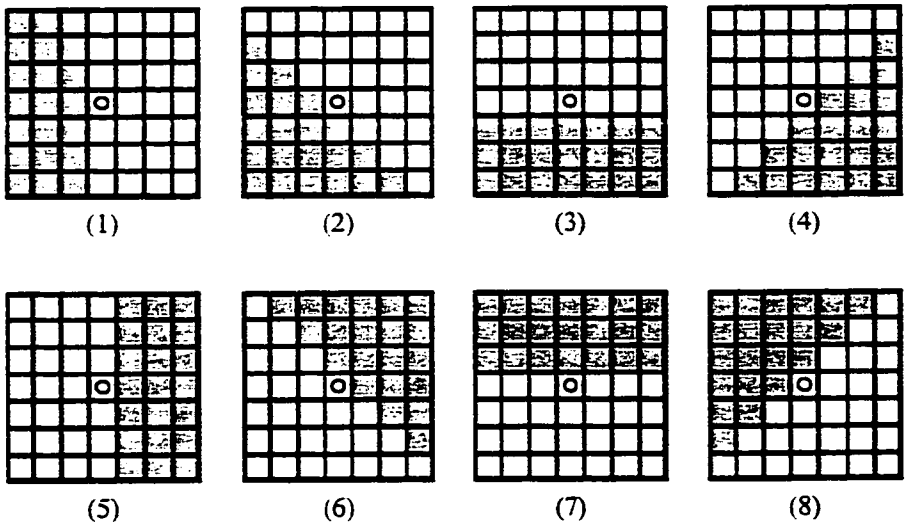


Figure 2.4. *Possible refined neighbourhoods associated with a 7×7 window (white areas).*

Since a refined neighbourhood consists of pixels from the same side of the prospective edge the calculated local statistics should be better estimates of the ensemble statistics. This should lead to a better restoration of the intensity of the pixel under consideration.

Implementation and Results

Figs. 2.1(e) and 2.2(e) show the results of applying the refined filtering method to the noisy synthetic image (Fig. 2.1(b)) and natural image (Fig. 2.2(b)) respectively. In comparison to the fixed neighbourhood methods, noise is removed effectively from the vicinity of most edges. However, in some regions (for example Lenna's face in Fig. 2.2(e) or equivalently in Fig. 2.3(e)) the noise is completely left unchanged. These are areas in which none of Lee's refined neighbourhoods (Fig. 2.4) would be recognised as a valid homogeneous region. Moreover, Lee's refined method may lead to false contours. The quantitative measures (Table 2.1 and 2.2) also show a considerable improvement over the fixed-neighbourhood methods.

2.5.2 Adaptive Neighbourhood (AN) Filters

The original LLMMSE estimator assumes that the ensemble mean and variance of a pixel are equal to the local mean and variance of all pixels in a neighbourhood around that pixel. This assumption is valid only if the pixels used to calculate the local statistics belong to the same ensemble (i.e., region) as the pixel under consideration. The fixed neighbourhood methods are based on the *a priori* assumption that all pixels in the applied window belong to the same region.

Although the orientation of edges is taken into account in Lee's refined neighbourhood method, the fixed window size is still the main criterion for selecting the neighbourhood. Selection of a refined neighbourhood remains limited to only a few possible choices (Fig. 2.4). In most practical cases, changing the neighbourhood to one of the predefined subsets cannot completely exclude pixels of other regions from the neighbourhood. This is especially true for the pixels that belong to small objects or textured regions. By contrast, the adaptive neighbourhood method will take similarity of pixels into account without an *a priori* assumption about neighbourhood sizes or shapes.

The adaptive neighbourhood (AN) filtering method was first introduced by Paranjape *et al.* [Paranjape94a, Paranjape94b] to remove signal-independent noise. The AN method was extended by Das and Rangayyan [Das97] and Rangayyan and Das [Rangayyan98a] to remove multiplicative noise. Recently Rangayyan, Ciuc, and Faghih [Rangayyan98b] improved the criteria for selecting the neighbourhood and applied the method to other types of signal-dependent noise (Poisson, film-grain, and speckle).

In the adaptive neighbourhood method, a pixel aggregation algorithm is used to grow a neighbourhood around each pixel. The pixel aggregation algorithm starts by checking the grey levels of pixels that are connected to the pixel under consideration (seed-pixel). The criterion used to include a pixel in the neighbourhood is that the absolute difference between the grey level of that pixel, $f(m,n)$, and the grey level of the seed, $f(i,j)$, is less than a threshold T, i.e.:

$$|f(m,n)-f(i,j)| \leq T, \qquad (2.10)$$

where the pixel at location (m,n) is directly connected (8-connected [Gonzalez92]) to the seed.

For every pixel which is included in the initial neighbourhood, the algorithm continues by checking its 8-connected pixels for possible inclusion in the region. The algorithm continues until the region stops growing because all the pixels with properties similar to the seed are included in the region. Alternatively, a limit on neighbourhood size can be adopted.

In a noisy image, the original intensity of a seed pixel could be completely corrupted by the noise. Using a highly corrupted seed will lead to including a biased set of pixels in the neighbourhood grown. As a result, the local mean and variance of the pixels in the neighbourhood would not be a good estimate of the ensemble mean and variance of the original seed value.

To solve this problem, Rangayyan *et al.* [Rangayyan98b] used a modified version of the similarity measure (Eqn. 2 10) to grow the neighbourhood region:

$$\left| f(i,j) - \tilde{f}(i,j) \right| \leq T \qquad (2.11)$$

where $\tilde{f}(i,j)$ is an initial estimate of the undegraded seed value. An estimate of the seed intensity was obtained using the α-trimmed mean value, $f_{\alpha TM}(i,j)$, calculated within a 3×3 window [Pitas92, Rangayyan98b]; this estimate was used to grow the neighbourhood.

The shape and size of the neighbourhood grown and hence the performance of the filter strongly depends on the threshold value T. A small threshold would lead to very small regions over which the noise statistics cannot be reliably estimated. A large threshold would lead to regions that may grow over edges with the result that homogeneity in such region is no longer guaranteed.

The threshold T must be a function of the noise parameters. For images with higher levels of noise we should increase the threshold value; otherwise the regions will not grow in the degraded image. The threshold value used by Rangayyan *et al.* [Rangayyan98b] is the standard deviation of the noise:

$$T = \sigma_n . \qquad (2.12)$$

Therefore, in order to grow the regions, we first need to estimate the noise parameters. In most practical images, a uniform region can typically be found near the boundaries of the image. We can use such uniform regions to estimate the noise parameters. However, as will be shown in Chapter 6, it is easy to introduce biases into the noise measurement.

Implementation and Results

Figs. 2.1(f) and 2.2(f) show the result of applying the AN filtering method to the noisy synthetic and natural images of Fig. 2.1(b) and 2.2(b) respectively. When compared to the methods discussed earlier, the AN method has been more successful in filtering the noise from the test images. Lee's refined method is almost as successful as the adaptive neighbourhood method in filtering the noise from the synthetic test image (Fig. 2.1(b)). This is mainly because Lee's refined neighbourhoods (Fig. 2.4) are in close match with the structure of the synthetic image. However, in the natural test image (Fig. 2.2(b)) there are many locations where none of Lee's reduced windows (Fig. 2.4) result in a homogeneous neighbourhood. As a result, Lee's refined method could not remove the noise from such locations (for example, compare Lenna's face in Figs. 2.2(e) and (f) or equivalently in Figs. 2.3(e) and (f)). Table 2.1 and 2.2 give a comparison of the quantitative measures for the filtered images using the various methods.

2.5.3 Adjusting the AN Algorithm for Low-Contrast and Low-SNR Images

The adaptive neighbourhood algorithm relies on the assumption that a grown region does not cross the edges and grows only in the area that contains the seed. However, this assumption is not valid if the difference between the intensities of adjacent regions is small (low-contrast images) or when the chosen threshold, T in Eqn. 2.11, is too big, or equivalently (Eqn. 2.12) when the image is very noisy (low-SNR images).

To adjust the adaptive neighbourhood method to low-contrast or low-SNR images, we need to decrease the value of the threshold. However, as we saw in Section 2.5.2, small threshold values usually lead to large biases in accepting the pixels into the region and hence large errors in estimating the pixel ensemble parameters.

In this section, we will propose a new method, which is a modification of the method that was explained in the previous section [Rangayyan98b]. These modifications will allow us to use smaller threshold values but still retain reasonably small estimation errors.

In the original adaptive neighbourhood method, after degrading an image with zero mean Gaussian noise, the intensity of each pixel would treated as a random variable whose density function has a Gaussian form. The ensemble mean of this random variable is equal to the undegraded pixel value (because the noise is zero-mean) and its variance is equal to the variance of noise. If we represent the ensemble mean of a noisy pixel by M_0, our initial rough estimation of this average value using the α-trimmed method [Pitas92, Rangayyan98b]) would be $M_0 + \varepsilon$, where ε is our estimation error. Now if we use the criterion given in Eqn. 2.11 to grow the region, only pixels with intensities between $M_0 + \varepsilon - T$ and $M_0 + \varepsilon + T$ would be allowed in the grown region.

As we mentioned before, the density function associated with the grey level of each pixel has a Gaussian shape. As a result, for $\varepsilon > 0$, the number of pixels in the grown region whose grey levels are between $M_0 + \varepsilon - T$ and $M_0 + \varepsilon$ would be larger than the number of pixels with intensities between $M_0 + \varepsilon$ and $M_0 + \varepsilon + T$. Therefore, the difference between the average grey level of the pixels in the grown region and ensemble mean, M_0, would be less than ε. This implies an improvement in the estimation of M_0.

Similarly, for $\varepsilon < 0$ we can improve our estimation of the ensemble mean by calculating the local average of the pixels in the region. By considering the Gaussian distribution of grey levels, it can be seen that to improve poor initial estimates of the ensemble mean (i.e., big estimation errors, ε) we need larger threshold values, whereas to compensate small estimation errors we can use smaller threshold values. Therefore, if we want to use a small threshold value to ensure not crossing the borders between low-contrast regions, we must ensure that we do not have poor initial estimation of the mean.

In the former method (Section 2.5.2), we used the α-trimmed method [Pitas92, Rangayyan98b] to find an initial estimation for each pixel. Here we propose a more complicated method that leads to better initial estimations. For high-contrast or high-SNR images, the results obtained from both the modified and the original methods are almost the same, but the computation time is slightly longer for the modified method. For low-contrast or very-low-SNR images, the extra processing time required by the new method is justified by its superior performance.

Improving the Accuracy of the Initial Ensemble Mean Estimation

We use an improved version of Lee's refined algorithm (Section 2.4.1 and [Lee81]) to make a quick estimation of the ensemble mean of each pixel.

As we saw in Section 2.4.1, Lee's refined filtering algorithm is not successful in removing the noise from all regions. There are many locations in an image where none of Lee's refined windows (Fig. 2.4) could result in an appropriate (homogeneous) neighbourhood. In fact, in Lee's refined neighbourhood method, there is an implicit *a priori* assumption about the neighbourhood of the considered pixel, which is not always valid. First, Lee assumes the pixel is located on a smooth region comparable in size with a 7×7 rectangular window. Moreover, if there are some edges or sharp features inside the 7×7 neighbourhood of the seed pixel, they all must be located on one side of the seed pixel. Therefore, the method is not successful if the pixel under consideration belongs to a small region (compared to the 7×7 window) or if the sharp features are not concentrated on one side of the seed pixel. The new method suggested here does not make such *a priori* assumptions about the pixels and hence can successfully estimate the ensemble mean for a much broader set of pixels.

The new method starts with the same procedure as Lee's refined filtering algorithm. First we look at a 7×7 neighbourhood around the pixel and try to figure out if there is any edge or sharp feature in the neighbourhood. For this purpose, we compare the local variance of the pixels in this neighbourhood with the variance of the noise. The difference between the noise variance and calculated local variance is due to the local variance of the original image in the neighbourhood. If this difference is small (compared to a threshold

chosen as a function of the noise variance) we assume that there is no edge or sharp feature in the neighbourhood. In this case, we can use the local mean of the pixels in the 7×7 window as the initial estimate of the ensemble mean. If the difference between the local variance and variance of noise is large, we should use a refined window as the neighbourhood (see Fig. 2.4).

As in Lee's refined algorithm, we assume that a high local variance calculated in a 7×7 window is due to the presence of a major edge in one side of the neighbourhood. But unlike Lee's method, we check the result to see if our assumption is valid using the following approach.

We initially assume that the high variance in the 7×7 window is due to the presence of a major edge in the window. If this assumption is true, after excluding the edge from the neighbourhood using one of the new neighbourhoods (see Fig. 2.4), the local variance should be dramatically decreased; in fact, the local variance must be very close to the noise variance. We can compare the local standard deviation with a threshold value, T, which is a function of the noise standard deviation ($T = 1.2\sigma_n$). Having a local standard deviation larger than the threshold T means that our model for the neighbourhood is too simple, and we need to use a more complicated model.

In the next step we change the neighbourhood to a 5×5 window and assume that the sharp features are concentrated mostly on one side of this neighbourhood. Based on this assumption, we repeat all the steps we took for the 7×7 neighbourhood. Finally, we modify the 5×5 rectangular neighbourhood to a refined 5×5 region (similar to the refined 7×7 regions in Fig. 2.4).

After modifying the neighbourhood, we again have to test the validity of our initial assumption about the presence of edges on only one side of the 5×5 neighbourhood. Again, we compare the local standard deviation with a threshold ($T = 1.2\sigma_n$). A large standard deviation means that the distribution of sharp features in the neighbourhood is more complicated than what we initially assumed. To make this distribution easier to model, we refine the neighbourhood to a smaller window (3×3). To estimate the ensemble mean of the seed pixel we use a weighted average of the all pixels in this small neighbourhood:

$$m(i,j) = \sum_{u=-1}^{1} \cdot \sum_{v=-1}^{1} D(u,v) f(i+u, j+v) \qquad (2.13)$$

where the D's are non-negative weights that sum to 1 and indicate the degree to which each neighbour of the seed-pixel should be allowed to contribute to the estimation of the seed's grey level. We use the weight matrix method suggested by Lev *et al.* [Lev77] to compute the weighting coefficients.

Implementation and Results

For high-contrast or high-SNR images, the performance of the new method is essentially the same as that of the former adaptive neighbourhood method. In this section we will compare the performances of these methods in removing the noise from a low-SNR image.

The image in Fig. 2.5(b) (or equivalently Fig. 2.6(b)) was derived from that in Fig. 2.5(a) by adding a non-quantized Gaussian zero-mean noise with variance 2025 (5 times more noise power than in the earlier images). Fig. 2.5(c) gives the result obtained by

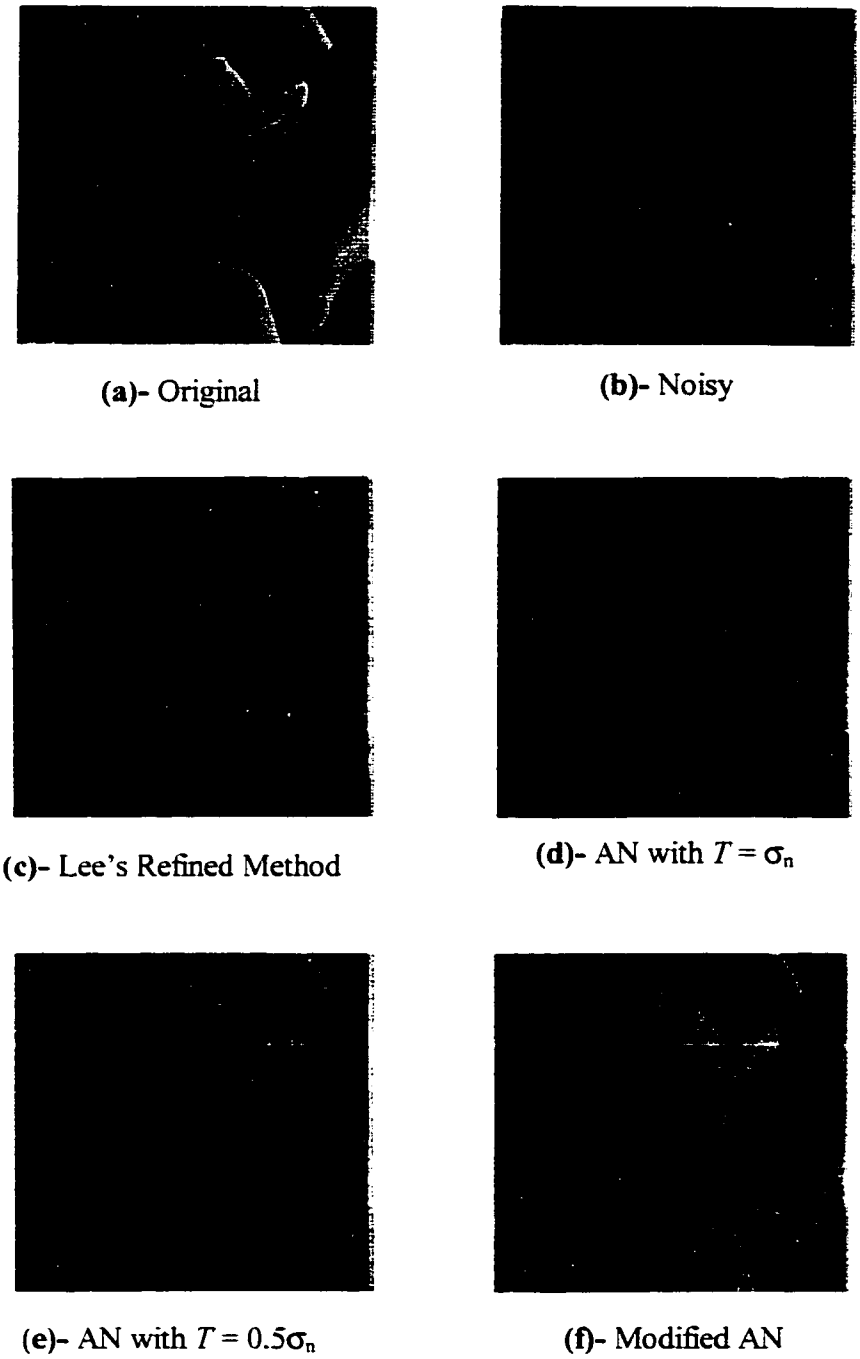


Figure 2.5 *Removing noise from a low-SNR natural test image.*
(a) Original 256×256 image with maximum grey level of 255.
(b) After adding non-quantized Gaussian noise with variance 2025 (grey level)^2.
Results of: (c) Lee's refined filtering method.
(d) Original adaptive neighbourhood (AN) method with $T = 1.0\sigma_n$.
(e) Original adaptive neighbourhood method with $T = 0.5\sigma_n$.
(f) Modified adaptive neighbourhood method.

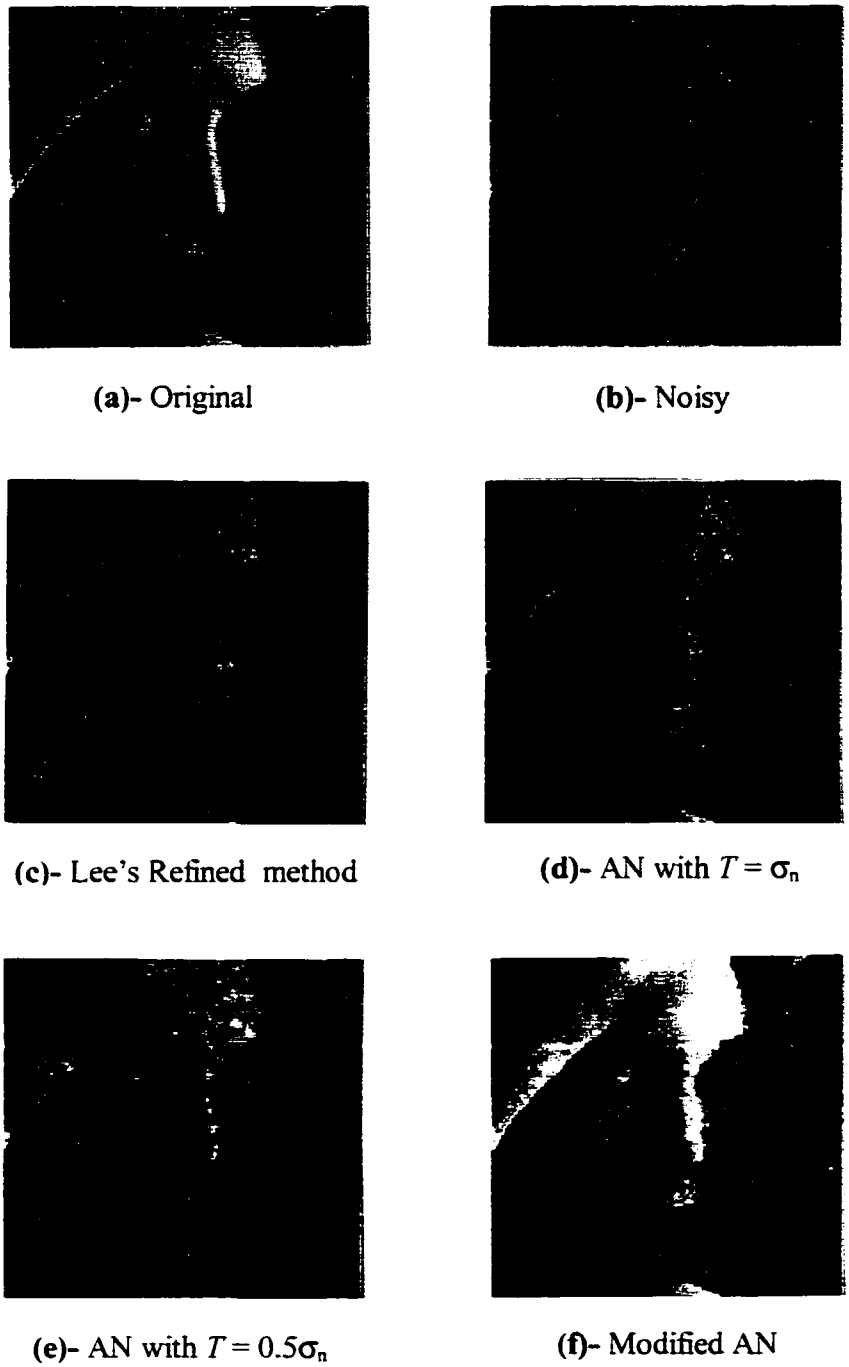


Figure 2.6 *(Magnified Version of Fig. 2.5) Removing noise from a low-SNR natural image. (a) Original image. (b) After adding Gaussian white noise. Results of: (c) Lee's refined filtering method. (d) Adaptive neighbourhood (AN) method with $T = 1.0\sigma_n$. (e) AN method with $T = 0.5\sigma_n$. (f) Modified AN method.*

using the refined Lee method. Fig. 2.5(d) shows the result obtained by applying the original adaptive neighbourhood method to the noisy image of Fig. 2.5(b). The threshold used to grow the neighbourhoods (T in Eqn. 2.11) is the same as the standard deviation of the noise estimated from the flat regions of the image. Most edges and other sharp features of the image are blurred in the filtered image. This is mainly because the regions grown have passed through edges and extended into other areas.

To limit the regions grown to homogeneous areas, we should decrease the value of T in Eqn. 2.11. Fig. 2.5(e) shows the result obtained for $T = 0.5\sigma_n$. The noise has still not been removed effectively in this image. This is because the seed values (our first estimations of the ensemble means) were not accurate and the regions grown are not large enough to correct the estimates through averaging.

Fig. 2.5(f) gives the result obtained using the new method. In comparison to the images in Figs. 2.5(d) and 2.5(e), the noise is removed more effectively in this image. Table 2.3 provides the quantitative measures for the filtered images.

Figure	*Description*	*MSE* *(Grey level)*^{2}	*ISNR* *(dB)*
2.4(b)	Additive noise , $\sigma^2 = 2025$	1997.4 ± 8.4	0.00 ± 0.00
2.4(c)	Lee's refined method	283.6 ± 1.4	8.48 ± 0.03
2.4(d)	AN method, $T = \sigma_n$	281.4 ± 2.2	8.51 ± 0.04
2.4(e)	AN method, $T = 0.5\sigma_n$	307.2 ± 2.4	8.13 ± 0.04
2.4(f)	Modified AN method.	248.2 ± 1.0	9.06 ± 0.02

Table 2.3 *Quantitative measures (MSE and SNR Improvement) for the results obtained by applying the LLMMSE filtering methods to the significantly degraded natural image (Fig. 2.5(b)).*

2.6 Conclusion

In this chapter, several spatial domain noise-filtering techniques have been discussed. All the methods discussed in this chapter are based on the LLMMSE estimator. As we saw in this chapter the LLMMSE estimator has the property that it smoothes noise in flat regions of the image while leaving the observation unchanged in the vicinity of edges.

To remove the noise more effectively from edge areas, several modifications to the basic LLMMSE method have been proposed in recent years. We reviewed some of these techniques and compared their performances in this chapter.

The LLMMSE filters can be classified in two main groups: fixed and advanced neighbourhood methods. The fixed neighbourhood techniques such as Lee's and Kuan's methods are not very successful in removing the noise from edge areas. As examples of the second group (advanced neighbourhood methods), Lee's refined neighbourhood method [Lee81] as well as the adaptive neighbourhood method [Rangayyan98b] were discussed in this chapter. In comparison to fixed neighbourhood methods, the advanced neighbourhood methods were more successful in removing the noise from the vicinity of edges.

Finally, we have presented a new method for filtering the noise from low-contrast or low-SNR images. The proposed method is a modification of the adaptive neighbourhood method [Rangayyan98b]. We have compared our approach with the other methods discussed in this Chapter, by applying them to some low-SNR images. The results of the new method have proven to be better in terms of visual quality and MSE.

In the rest of this thesis, we will concentrate on wavelet-based denoising techniques. As with the methods discussed in this chapter, the wavelet domain methods can adapt themselves to the local features of the given image.

Chapter 3

Wavelet Analysis

3.1 Introduction

This chapter provides the background necessary to understand wavelet-based image enhancement techniques that will be discussed in the next two chapters.

In a large variety of image-processing techniques, the image is transformed from the spatial domain to a different space for processing. The extra effort required to transform the image to a different domain is often justified by the fact that some image features become more apparent and distinguishable in a non-spatial domain. As an example, there may be a better separation between the signal and noise power in the frequency domain, and hence a simple filter in the frequency domain can be used to improve the SNR of a noisy image. Unfortunately, frequency domain techniques have a main drawback. They process an image in a global sense, so that they cannot be easily adjusted to the local features of the image. As a result, reducing noise in an image using frequency domain techniques involves a trade-off between reducing the spatial resolution in the image (blurring) and noise reduction.

In contrast to applying the discrete Fourier transform to an image, and losing all the spatial information of the signal, applying the short-time Fourier transform (STFT) or wavelet transform provides a compromise between the spatial and frequency domain information. These transforms map the signal to a spatial-frequency space in which there is simultaneous access to both spatial and frequency domain information.

Fourier analysis and the STFT [Allen77] are reviewed in the first part of this chapter. Then, the basics of wavelet analysis (both continuous and discrete) and of the subband coding technique are explained [Mallat89a, Rioul91, Daubechies92, Mallat98]. Finally, a fast wavelet transform (FWT) algorithm is presented which is based on the subband coding technique [Mallat89a, Mallat98]. The FWT algorithm is used frequently in the techniques discussed in this thesis.

3.2 Fourier Analysis

The Fourier transform provides an alternative representation of a spatial domain signal, $x(t)$, as a sum of sinusoids of different frequencies. In other words, it is a mathematical technique for mapping a signal to the frequency domain. The Fourier transform is an invertible transform. This means that the frequency domain representation can be converted back to the original domain by using the inverse Fourier transform.

For a discrete signal $x[n]$, where $x[n] = x(t + n\Delta t)$, $n = 0,1,2,....,N-1$, the discrete Fourier transform (DFT) is defined as:

$$X[u] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi un/N},$$

where $u = 0,1,2,....N-1$ and both signals, $x[n]$ and $X[u]$, are assumed to be periodic with period N. The inverse DFT is then given by:

$$x[n] = \frac{1}{N} \sum_{u=0}^{N-1} X[u] e^{-j2\pi un/N}.$$

The discrete Fourier coefficients, $X[u]$, are computed as the inner product of the original signal and sinusoids of finite duration. The coefficients can be considered as a

measure of the similarity between the signal and the sinusoids. The measure is computed over the whole range of the signal, and therefore is meaningful only for stationary signals that have similar properties along their whole length. Any abrupt change in signal behaviour is spread out over the coefficients of the whole range of frequencies, making the DFT an inappropriate tool for analysis. Therefore, more powerful techniques are required to analyse non-stationary signals.

Figs. 3.1(a) through (d) show two simple sinusoids and their associated DFTs. Fig. 3.1(e) shows a signal that is a simple sum of the two sinusoids across the full window; the DFT of this signal (Fig. 3.1(f)) describes the signal in terms of its global similarity to different sinusoids, which is quite meaningful for this stationary signal. Fig. 3.1(g) shows the combination of the same sinusoids in a different manner to create a non-stationary signal. Again the Fourier transform (Fig. 3.1(h)) is a measure of the global similarity of the signal to different sinusoids. However, this global measure is not suitable for analysing the non-stationary signals. The non-stationary signal in Fig. 3.1(g) has different behaviours in different regions of the space domain. which is completely neglected in its DFT.

There are two main approaches for analysing non-stationary signals: STFT and wavelet analysis. These techniques are discussed in the following sections.

3.3 Short-Time Fourier Transform

In an effort to adapt Fourier analysis to non-stationary signals, the STFT was introduced by Dennis Gabor [Gabor46]. In this approach, the Fourier transform is used to analyse a small section of the signal at a time. The frequency content of a non-stationary

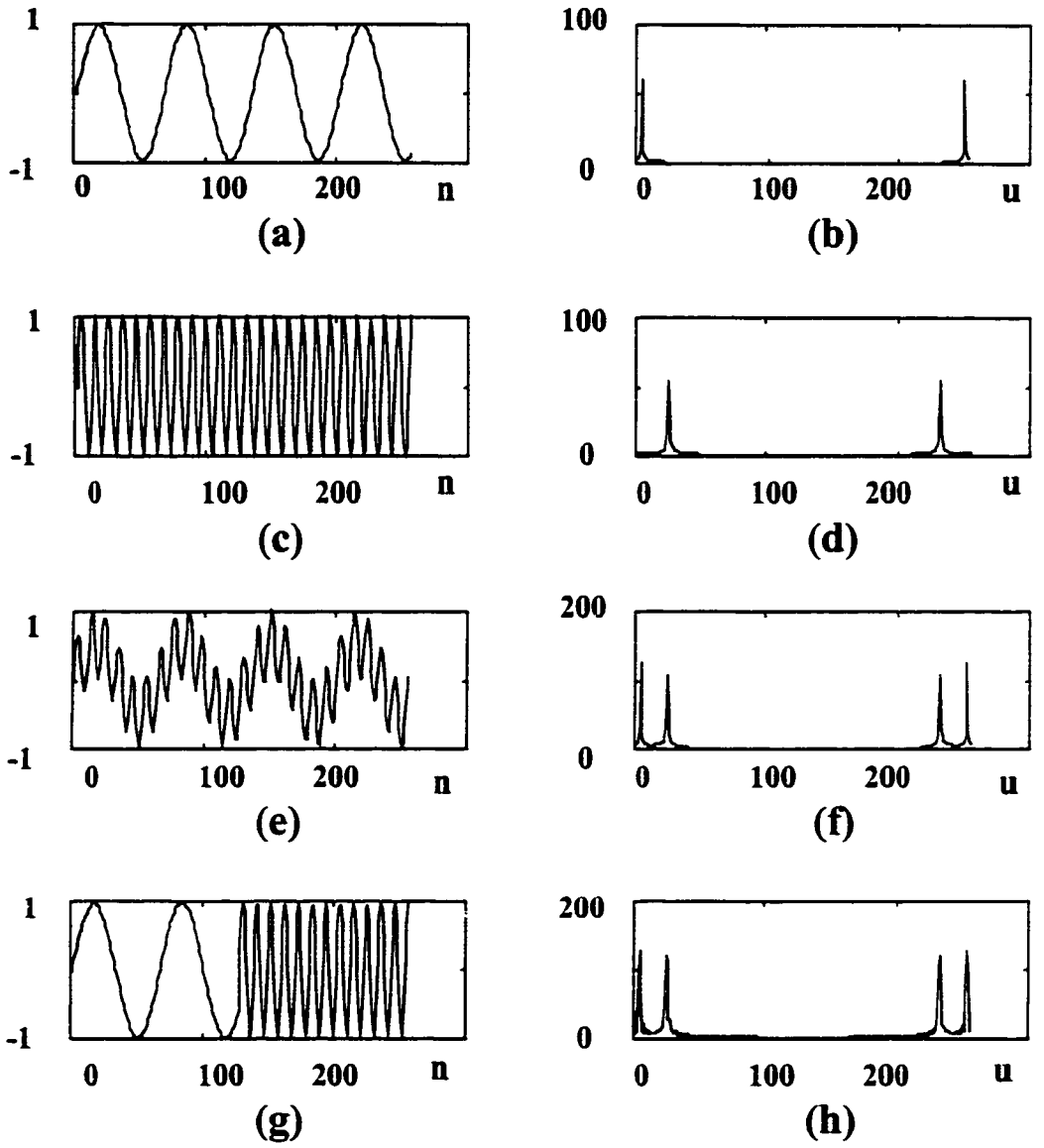


Figure 3.1 *Frequency domain signal analysis. The signals on the right are the discrete Fourier transforms of the signals on the left.*
(a) A low-frequency sinusoid, sin(3πn/100) and (b) its DFT.
(c) A high-frequency sinusoid, sin(19πn/100) and (d) its DFT.
(e) $y_1(n) = 0.5 \times [\sin(3\pi n/100) + \sin(19\pi n/100)]$ *and (f) its DFT.*
(g) $y_2(n)$ *is equal to* $\sin(3\pi n/100)$ *for* $n=0,\ldots,127$ *and* $\sin(19\pi n/100)$ *for* $n=128,\ldots,255$ *and (h) its DFT.*

signal is different for each section. Therefore the STFT is a function of the spatial location of the selected section.

Consider a signal $x(t)$, and assume it to be stationary when seen through a window $g(t)$ of limited extent, centred at location τ. The STFT of this signal is defined as the Fourier transform of the windowed signals $x(t)g^*(t-\tau)$:

$$STFT(\tau, f) = \int_{-\infty}^{\infty} x(t)g^*(t-\tau)e^{-j2\pi ft}\,dt,$$

where $g^*(t)$ is the complex conjugate of $g(t)$. This transform maps the signal to a two-dimensional function of position and frequency. In the spatial domain we can detect the location of an event quite accurately, but we have no information about the frequency content of that event. Similarly, in the frequency domain we can study the frequency behaviour of an event, but we have no spatial information about the event. The STFT provides some information about both the frequency content and the location of an event. To determine the position of an event more accurately, we have to use smaller windows, which leads to less accuracy in the frequency domain. In order to discriminate two pulses in the spatial domain (spatial resolution) they must be more than Δt apart. Similarly, two sinusoids will be discriminated in the frequency domain (frequency resolution) only if they are more than Δf apart. Resolution in position and frequency cannot be arbitrary small; in fact their product is lower bounded [Rioul91]:

$$\Delta t \Delta f \geq \frac{1}{4\pi}.$$

This implies that to improve the spatial resolution we have to give up frequency resolution and vice versa. This is a variant of the uncertainty principle or the Heisenberg inequality.

The main drawback of the STFT is that once we choose a particular size for the spatial window, the window is the same for all frequencies. This means that the position and frequency resolutions are the same for the entire spatial-frequency plane. High-frequency components are always associated with rapid changes in the spatial domain and hence a better spatial resolution is needed for the higher frequencies in the spatial-frequency plane. Therefore, we need a more flexible approach where the STFT window size is allowed to change in the spatial-frequency plane. This will allow us to selectively improve either spatial or frequency resolution at the cost of decreasing the resolution in the other domain [Rioul91].

3.4 Wavelet Analysis

Fourier analysis consists of breaking up a signal into sine waves of various frequencies. Similarly, wavelet analysis is the breaking up of a signal into a particular basis of functions, called wavelets.

In Fourier analysis the basis functions are sinusoids which are very poorly localised in the spatial domain, but perfectly localised in the frequency domain. The wavelets make a compromise between spatial and frequency supports. They have finite support in both domains.

3.4.1 Continuous Wavelet Analysis

The continuous wavelet transform of a signal is a set of coefficients which shows the similarity between the signal and a set of scaled, shifted versions of an original wavelet

function. To measure this similarity, we calculate the correlation between the original signal and the shifted and scaled wavelet [Grossman84, Matlab96]:

$$C(a,b) = \int_R f(x).\frac{1}{\sqrt{a}}\Psi(\frac{x-b}{a})dx, \quad (3.1)$$

where $\Psi((x-b)/a)$ is obtained from the original wavelet, $\Psi(x)$, by first scaling it by a factor of a and then shifting it by b. The factor $(1/\sqrt{a})$ is used to normalise the power of the scaled wavelet.

The original or mother wavelet $\Psi(x)$ is concentrated around $x = 0$, oscillates around the x-axis and averages to zero. The scaled and shifted wavelet, $\Psi((x-b)/a)$, is concentrated around the point b and its region of support is a times larger than the region of support of the original wavelet.

The size of the revealed details in the signal $f(x)$ is proportional to the size of the domain in which the scaled wavelet is not too close to zero. Therefore, for small scales (*i.e.*, small a's), local features of a signal mainly affect the wavelet coefficients, while for larger scales the coarser features in the signal affect the wavelet coefficients. Local features of a signal are associated with its high-frequency components while global features are associated with the low-frequency components.

If the wavelet transform of a function, $f(x)$, has a non-zero coefficient $C(a,b)$, it means that the function has a contribution in the frequency band associated with the scale a. The second parameter b indicates the spatial location of the signal $f(x)$ which has this frequency content. Therefore, in addition to representing the frequency content of a signal, wavelet coefficients can provide information about the location of the components.

To synthesise the signal from its continuous wavelet coefficients we multiply each coefficient with a properly shifted and scaled wavelet and add the results [Matlab96]:

$$f(x) = \frac{1}{K_\Psi} \int_{R^-} \int_{R} C(a,b) \frac{1}{\sqrt{a}} \Psi(\frac{x-b}{a}) \frac{dadb}{a^2},$$

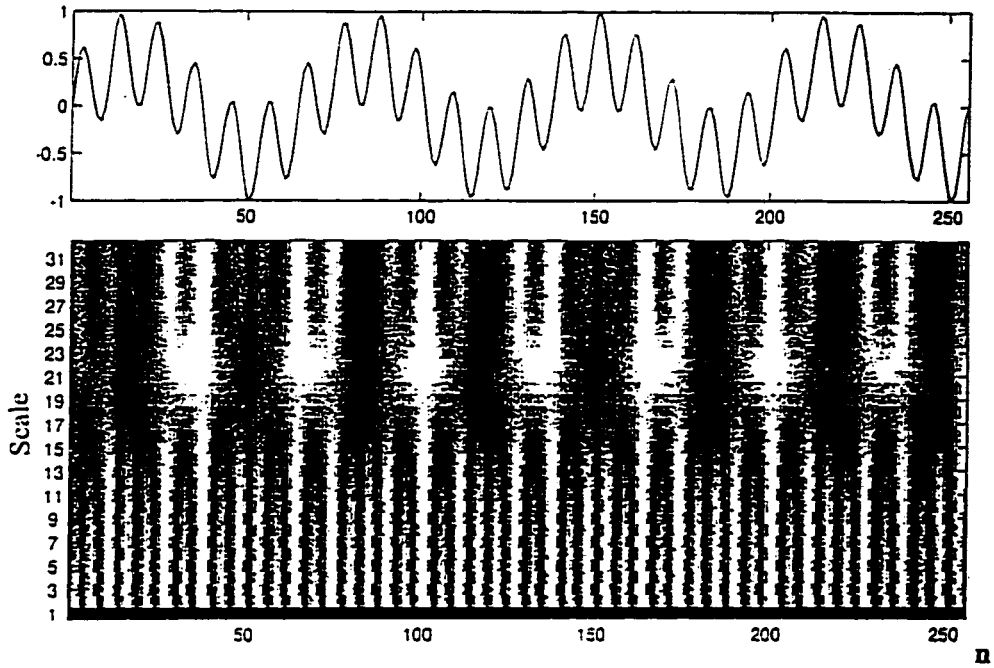
where K_Ψ is a constant which depends on the selected wavelet. Fig. 3.2 shows the continuous wavelet transforms of a stationary and a non-stationary signal. These test signals are exactly the same as the signals given in Fig. 3.1. In Fig. 3.2(b), the x-axis represents position, the y-axis represents scale, and the intensity at each x-y point represents the magnitude of the corresponding wavelet coefficient. Both the frequency and spatial behaviour of the signals are apparent from the wavelet coefficients.

3.4.2 Discretization of the Time-Scale Parameters

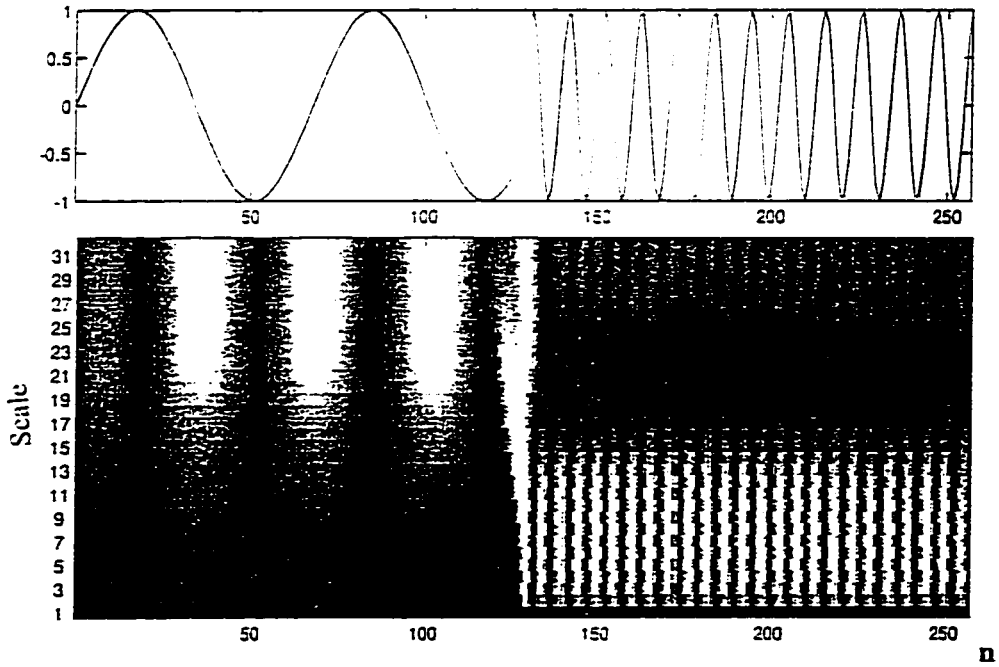
Calculating the wavelet coefficients at all possible scales is very expensive in terms of computation time. It also generates a huge number of coefficients, which are not easy to handle. However, if we choose scales and positions based on powers of two (dyadic scales and positions) then our analysis will be much more efficient but just as accurate [Matlab96]. By choosing the scales and shifts based on powers of two:

$$a = 2^j, \qquad b = ka = k2^j, \qquad (j,k) \in Z^2.$$

a shifted and scaled wavelet can be written in the following simple form:



(a) *A stationary signal and its continuous wavelet transform.*



(b) *A non-stationary signal and its continuous wavelet transform.*

Figure 3.2. *Wavelet domain signal analysis.*
(a) A stationary signal ($y_1(n)=0.5\times[\sin(3\pi n/100)+\sin(19\pi n/100)]$) and its continuous wavelet transform (b) A non-stationary signal ($y_2(n) = \sin(3\pi n/100)$ for $n=0,\ldots,127$ and $y_2(n) = \sin(19\pi n/100)$ for $n=128,\ldots, 255$) and its continuous wavelet transform.

$$\frac{1}{\sqrt{a}}\Psi(\frac{x-b}{a}) = 2^{-j/2}\Psi(2^{-j}x-k).$$

If we define:

$$\Psi_{j,k}(x) = 2^{-j/2}\Psi(2^{-j}x-k), \qquad (j,k)\in Z^2,$$

the wavelet coefficients become [Matlab96]:

$$C(j,k) = \int_{R} f(x)\Psi_{j,k}(x)dx \qquad (j,k)\in Z^2 .$$

To reconstruct the original signal from the wavelet coefficients, $C(j,k)$, we weight each coefficient with a properly shifted and scaled wavelet and add the results:

$$f(x) = \sum_{j\in Z}\sum_{k\in Z} C(j,k)\Psi_{j,k}(x). \qquad (3.2)$$

However, the accurate reconstruction is only possible for very special choices of the basic wavelet $\Psi(x)$ [Daubechies90, Meyer90, Matlab96].

3.4.3 Calculating the Wavelet Coefficients

Although in Eqn. 3.1 the signal $f(x)$ is assumed to be a continuous signal, in the real world, when we use a computer to process a signal, all the computations must be performed on a discrete signal.

To calculate the wavelet coefficients $C(a,b)$ associated with the samples of the signal $f(x)$ we take the original wavelet and calculate its correlation with the signal. The result is the first wavelet transform coefficient. Then we shift the wavelet one sample to the right and calculate the next coefficient. We repeat this process until the whole length of the sampled signal is covered. To compute the wavelet coefficients at the scale a, we scale the wavelet and repeat the whole process [Matlab96].

For a signal of length L and a wavelet of length W, the number of multiplications required to compute the coefficients at level n is nWL. Therefore, the number of multiplications needed to compute an N-level decomposition is:

$$WL + 2WL + ... + NWL = \frac{N(N+1)}{2} WL \cong \frac{N^2 WL}{2}.$$

In practice, however, we never use the above procedure to compute the wavelet transform of a signal. As we will see later in this chapter, there is a more efficient algorithm for computing the wavelet coefficients. This algorithm, introduced by Mallat [Mallat89a], is based on subband coding theory of signals [Croisier76, Matlab96].

3.5 An Alternative Viewpoint: Subband Coding

Wavelet theory provides a unified framework for a number of techniques, which had been developed independently for various signal-processing applications. For example, the wavelet series expansion developed in applied mathematics and subband coding developed for speech and image compression, are different views of a single theory [Portnoff80, Rioul91]. In this section we briefly discuss the subband coding technique, which is the basis for a fast wavelet transform algorithm introduced by Mallat [Mallat89a, Mallat98].

3.5.1 Approximation and Details of a Signal

For many signals, the low-frequency content is the most important part. It is what gives the signal its identity and can be used to approximate the signal. High-frequency components, on the other hand, carry the information required to generate the details in the signal [Matlab96].

Using subband coding terminology, we call the low- and high-frequency components of a signal "Approximation" and "Details" respectively. Fig. 3.3 represents the filtering process required to generate the approximation and detail signals. By using two complementary (high-pass and low-pass) filters we can decompose the original signal $f(n)$ to its approximation, $cA(n)$, and detail coefficients, $cD(n)$ [Matlab96].

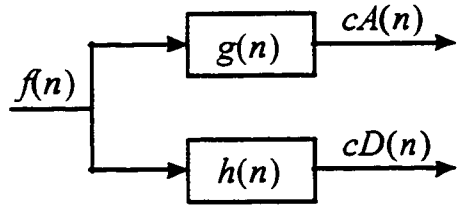


Figure 3.3 *Filtering process required to generate the approximation and detail coefficients of an input signal. h(n) and g(n) are the high-pass and low-pass filters, respectively.*

The filtering process effectively doubles the size of the data set to be manipulated. In fact the number of the approximation and detail coefficients is slightly more than the length of the original signal (as a result of convolving the input signal with the filter coefficients). See Chapter 6 regarding the consequences of this convolution growth. To reduce the size of the output data to the same level as the input data, we add a down-sampling step to the filtering process (Fig. 3.4) [Matlab96].

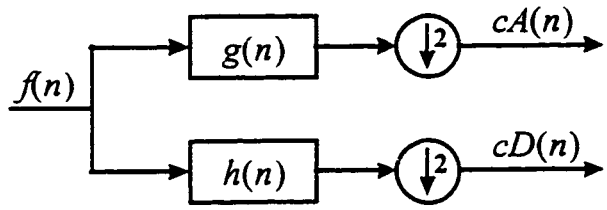


Figure 3.4 *Reducing the number of output coefficients to half by down-sampling.*

In general down-sampling causes aliasing. Therefore, it is not always possible to reconstruct the original signal from the down-sampled coefficients. If $g(n)$ and $h(n)$ were ideal half-band low-pass and high-pass filters (having a frequency pass-band equal to 1 over the normalised frequency range, and zero elsewhere) perfect reconstruction of the original signal would be possible [Rioul91]. To reconstruct the original signal from its sub-sampled coefficients. it is not necessary to use ideal, but impractical filters [Rioul91]. For perfect reconstruction of the original signal, we need two specific reconstruction filters ($g'(n)$ and $h'(n)$ in Fig. 3.5). We shall discuss later the close relationship between these specific reconstruction filters and the non-ideal decomposition filters ($g(n)$ and $h(n)$).

After up-sampling the coefficients, by inserting zeros between the samples, we pass them through the reconstruction filters to generate the approximation and detail signals ($A(n)$ and $D(n)$). Finally we add these two signals to generate the reconstructed signal ($\hat{f}(n)$ in Fig. 3.5).

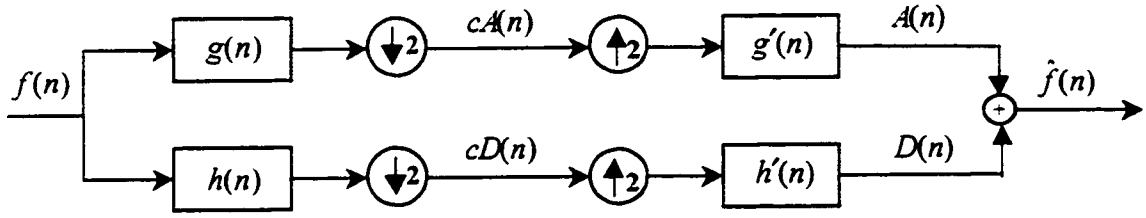


Figure 3.5 *Reconstructing the original signal from its approximation and detail coefficients by passing the up-sampled coefficients through the reconstruction filters.*

Unless the decomposition and reconstruction filters meet some specific constraints, the reconstructed signal, $\hat{f}(n)$, is not equal to the original signal. Filters that meet these constraints are said to have perfect reconstruction property [SmithM86, Rioul91]. The easiest case to analyse appears when $h(n)$ and $g(n)$ are quadrature mirror filters, which means that they are related by:

$$h(n) = (-1)^n g(L-1-n) \qquad (3.3)$$

where L is the filter length, and the decomposition and reconstruction filters are identical within time reversal, that is:

$$\begin{aligned} h'(n) &= h(L-1-n) \\ g'(n) &= g(L-1-n). \end{aligned} \qquad (3.4)$$

Under these conditions, the reconstruction filters, $h'(n)$ and $g'(n)$ are also quadrature mirror filters which means Eqn. 3.3 holds for them [Matlab96].

The approximation sequence $cA(n)$ from the first low-pass filter can itself be broken down further using multiple level decomposition as is demonstrated in the next section.

3.5.2 Multiple Level Decomposition

In the previous section we used a specific bank of filters (decomposition filters) to decompose a signal into two sub-sequences at half rate. This decomposition process can be iterated on either or both sequences. In particular, to achieve finer frequency resolution at lower frequencies, we iterate the scheme on the lower band only (see Fig. 3.6) [Rioul91, Matlab96].

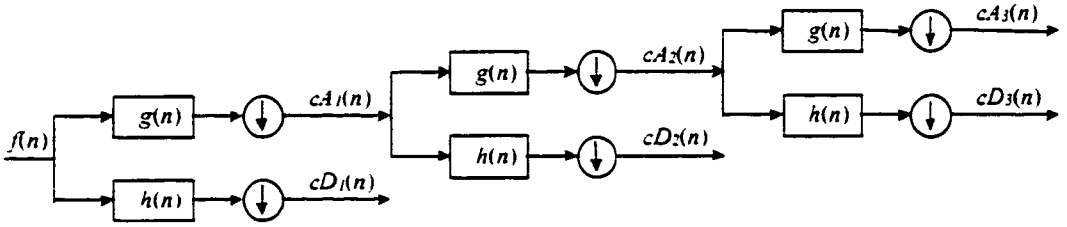


Figure 3.6 Iterating the decomposition process for the approximation coefficients.

If $h(n)$ is a half-band high-pass filter then its quadrature mirror filter, $g(n)$, is a half-band low-pass filter. Therefore, in the first level of the decomposition (see Fig. 3.6), the detail coefficients, $cD_1(n)$, correspond to upper band and the approximation coefficients, $cA_1(n)$, to the lower band of the frequency spectrum. We can continue this process for several levels. In each iteration we divide the lower band generated in the previous level into two new subbands.

Each further iteration halves the width of the low-band or increases the frequency resolution by two, but due to down sampling halves the spatial resolution [Rioul92].

3.5.3 Subband Coding and Discrete Wavelet Analysis

We will now show the relationship between subband coding and wavelet analysis. As mentioned in Section 3.4.2, to reconstruct a signal from its wavelet coefficients, we have

to multiply each coefficient with a properly scaled and shifted wavelet and add the results.

$$f(x) = \sum_{j \in Z} \sum_{k \in Z} C(j,k) \Psi_{j,k}(x).$$

Here j determines the wavelet scale. All wavelet coefficients, $C(j,k)$, with the same index of j correspond to the same band of frequencies. The detail signal at the j^{th} level of a subband coder, $D_j(x)$, can be written as:

$$D_j(x) = \sum_{k \in Z} C(j,k) \Psi_{j,k}(x). \qquad (3.5)$$

Therefore, based on Eqn. 3.2, the original signal $f(x)$ can be written as the sum of all its details [Matlab96]:

$$f(x) = \sum_{j \in Z} D_j(x).$$

Next by considering level $j = J$ as the reference level, we can write:

$$f(x) = \sum_{j \in Z} D_j(x) = \sum_{j \le J} D_j(x) + \sum_{j > J} D_j(x).$$

The details associated with indices $j > J$ correspond to the scales greater than 2^J. We group these coarser details into one component, $A_J(x)$, that is called the approximation at level J. Now we can write [Matlab96]:

$$f(x) = \sum_{j \le J} D_j(x) + A_J(x). \qquad (3.6)$$

In other words, wavelet analysis decomposes a signal to detail signals at successive scales plus a low-frequency (approximation) component. This is exactly what we get from a subband coder. In fact wavelet analysis and subband coding are two views of the same theory [Rioul91]. To calculate the wavelet coefficients more efficiently, Mallat [Mallat89a] applied the subband coder shown in Fig. 3.6 to the signal.

For a signal of length L and analysis filters of length F, the number of multiplications required to compute the coefficients at level n is $\frac{1}{2^{n-1}}FL$. Therefore, the number of multiplications needed to compute an N-level decomposition is:

$$2FL + FL + \frac{1}{2}FL... + \frac{1}{2^{N-2}}FL \leq 4FL .$$

From the previous discussion in Section 3.4.3, we know that the number of multiplications required to calculate an N-level wavelet decomposition directly is $N(N+1)WL/2$, where W is the length of the wavelet and L is the length of the signal. The shape of the wavelet (and hence its length, W) is determined by its decomposition filters [Mallat 89a]. The length of a wavelet, W, is always larger than the number of its corresponding filter coefficients, F [Matlab96]. Therefore, using the FWT algorithm can dramatically decrease the number of multiplications required to perform an N-level wavelet decomposition (from $N(N+1)WL/2$ to $4FL$).

3.6 The Two-Dimensional Fast Wavelet Transform

The subband coding approach was extended by Mallat [Mallat89a, Mallat98] to compute the wavelet transform of two-dimensional signals.

In the analysis (decomposition) phase of the two-dimensional fast wavelet transform (Fig. 3.7) each row of the image is separately filtered by the low-pass and high-pass decomposition filters. The resulting pair of row-transformed images is similarly filtered in the column direction. The result is a sort of four subband images at the first decomposition level. The subband image obtained by low-pass filtering of the image in both directions

(along rows and columns) is a low-pass filtered version of the original image. This image (cA_{j+1} coefficients) is passed through to the next step for further subband decomposition. The three detail images (cD^{H}_{j+1} , cD^{V}_{j+1} and cD^{D}_{j+1} coefficients) correspond to specific, non-overlapping bands in the frequency domain. The cD^{H}_{j+1} coefficients, for example, are formed by low-pass filtering the rows followed by high-pass filtering the columns, and is therefore sensitive to horizontally-oriented features. In the same way cD^{V}_{j+1} and cD^{D}_{j+1} coefficients are sensitive to vertical and diagonal features of the image.

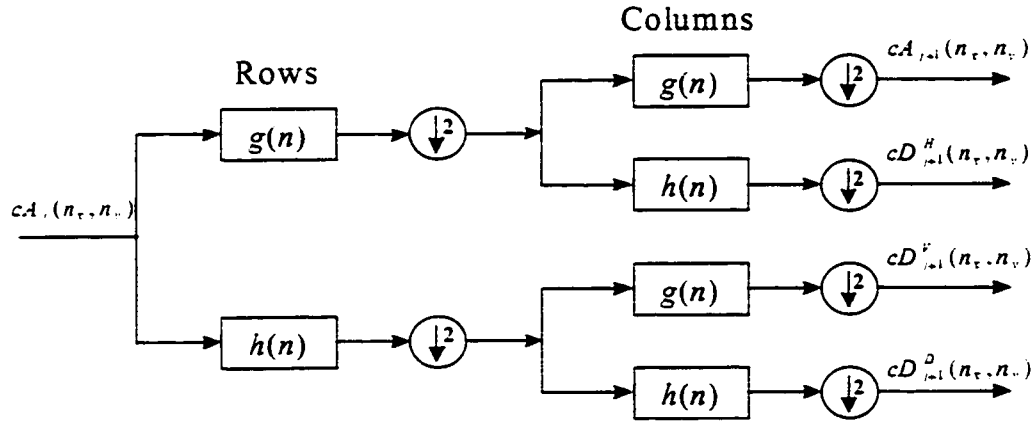
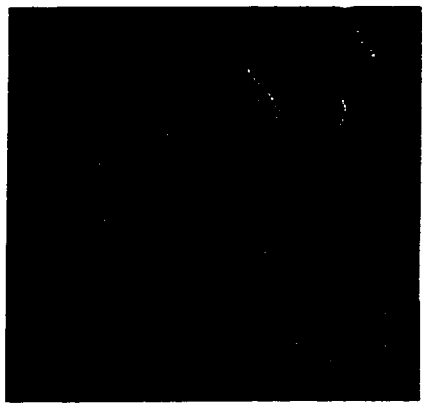


Figure 3.7 *Two-dimensional decomposition process at a general level (level j).*

The approximation and detail images obtained from one level decomposition of a test image are shown in Fig. 3.8. To generate these images, we did not down-sample the wavelet coefficients, which makes it easier to compare the detail images with the original image. Therefore, the detail and approximation images have the same size as the original image (256×256). Each detail image has extracted the high frequency components (i.e., edges) located in a specific direction (horizontal, vertical, or diagonal). The horizontal edges, for instance, are highlighted in the horizontal image (Fig. 3.8(c)). The

approximation image, on the other hand contains all the lower-frequency components (the lower half-band of the spectrum) which remain after subtracting the details from the original image. Now we can repeat the decomposition process for the approximation image obtained in the first level. The result would be the second level approximation and detail images (Fig. 3.9). The new detail images (Figs. 3.9(c) to 3.9(e)), represent the frequency components in a lower band of the spectrum compared to the detail images of the first level (Figs. 3.8(c) to 3.8(e)). The range of frequencies in the second-level approximation image (Fig. 3.9(b)), is now limited to the lower quarter of the spectrum while the first-level approximation image (Fig. 3.8(b)), includes all the components in the lower half-band of the spectrum. By iterating this process for several levels, we can completely classify the frequency components of an image into a series of detail images, each of them containing the components in a specific range of frequencies. Therefore, similar to the Fourier analysis, the wavelet transform gives us the power to analyse an image based on its frequency content. In fact, the wavelet transform has much of the frequency analysing power of the Fourier transform, but unlike the Fourier transform it preserves the spatial information. The Fourier transform of the same test image is shown in Fig. 3.10(b). All the spatial information, like edges, are lost in the frequency domain, while they are preserved by the wavelet transform (Figs. 3.8 and 3.9).



(a)- Original

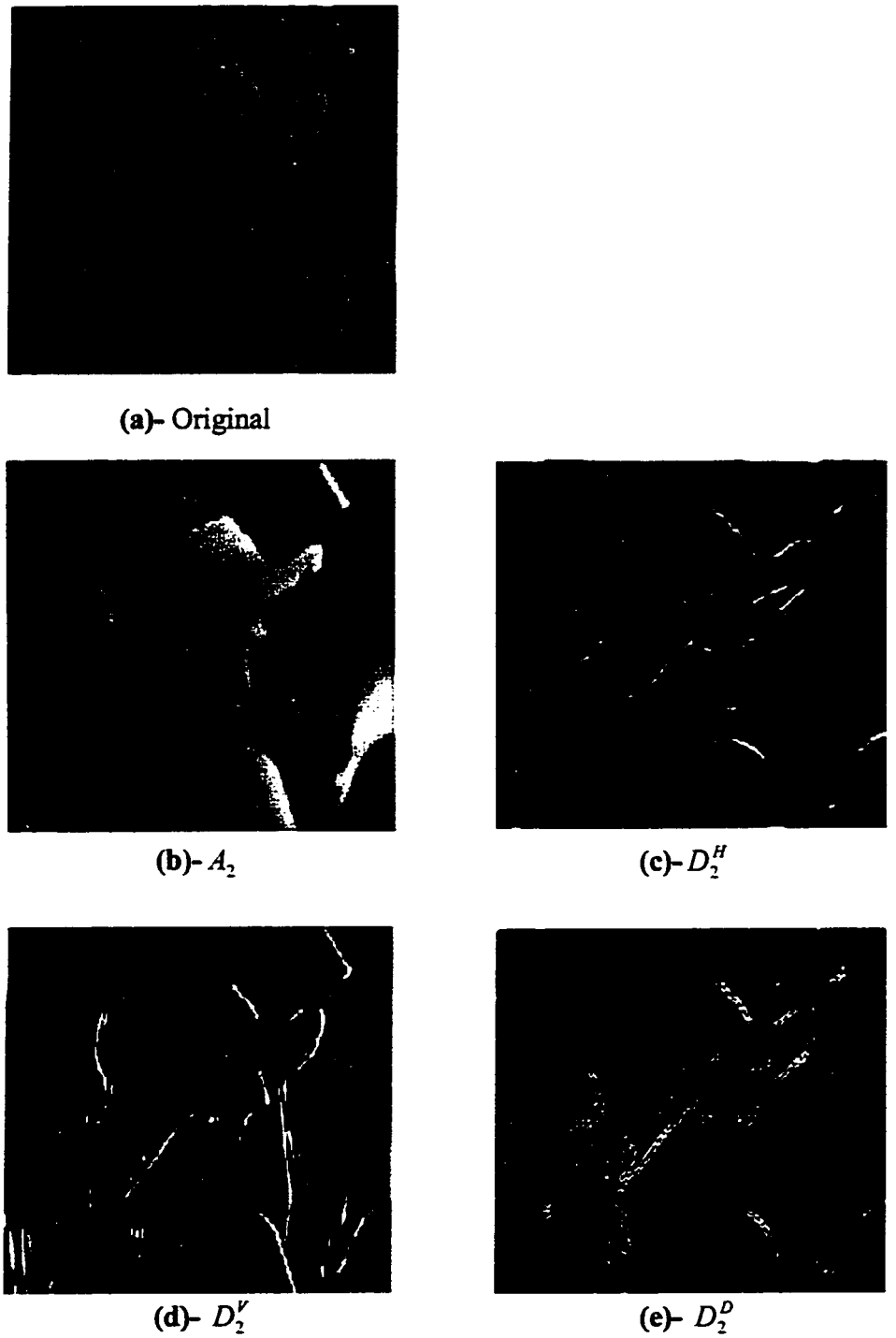
(b)- A_1

(c)- D_1^H

(d)- D_1^V

(e)- D_1^D

Figure 3.8 *The two-dimensional discrete wavelet transform. The approximation and detail images at the first decomposition level. (a) The original 256× 256 image. (b) The approximation coefficients. (c) The horizontal detail image. (d) The vertical detail image. (e) The diagonal detail image.*



(a)- Original

(b)- A_2

(c)- D_2^H

(d)- D_2^V

(e)- D_2^D

Figure 3.9 *The two-dimensional discrete wavelet transform. The approximation and detail images at the second decomposition level. (a) The original 256× 256 image with the range of grey levels between 0 to 255. (b) The approximation coefficients. (c) The horizontal detail image. (d) The vertical detail image. (e) The diagonal detail image.*

(a)- Original **(b)**- Discrete Fourier Transform

Figure 3.10 *The two-dimensional Discrete Fourier Transform.*

To reconstruct the image from its two-dimensional discrete wavelet coefficients, we start from the approximation and detail images at the lowest level and combine them to generate the approximation image at the next upper level. We continue this combining process up to the highest level. Fig. 3.11 shows one step of the reconstruction process.

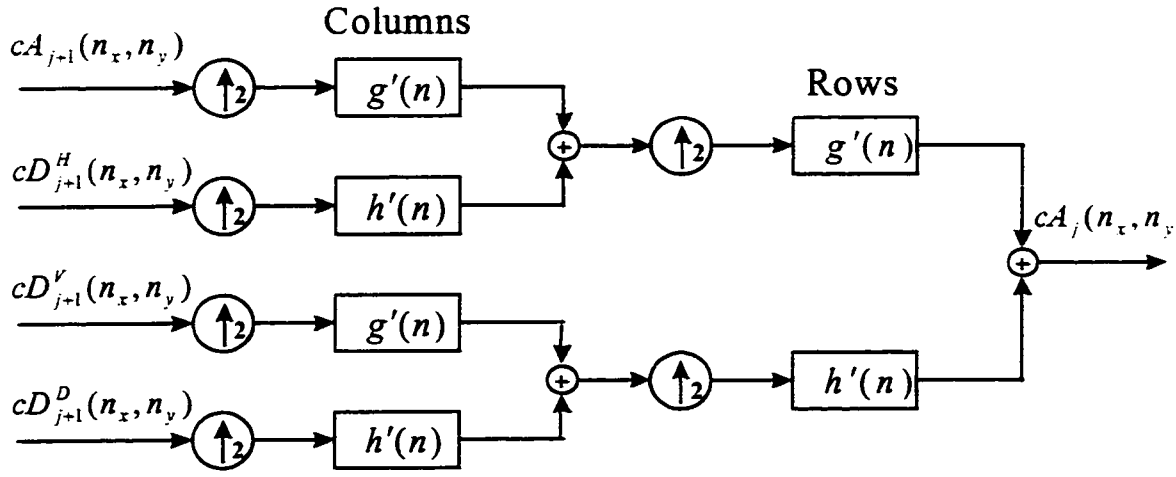


Figure 3.11 *Two-dimensional reconstruction process at a general level (level j).*

3.7 Specialised Wavelets for Image Processing

3.7.1 Biorthogonal Wavelet Transform

In Section 3.5.1, when we introduced the subband coders, we said that the decomposition and reconstruction filters are identical within time reversal (Eqn. 3.4). However, this is not always true. There are some classes of wavelets for which the corresponding decomposition and reconstruction filters are completely different (biorthogonal wavelets). From the wavelet theory viewpoint, this is associated with using two different wavelets, $\Psi(x)$ and $\tilde{\Psi}(x)$, to analyse and synthesise the signal [Daubechies90].

For some of the algorithms that will be discussed later in this thesis, it is desirable to make the wavelet coefficients more sensitive to the edges of the image. For this purpose, the wavelet function we use to decompose the image must have a shape similar to the difference of Gaussian functions [Canny85], which can only be achieved by using biorthogonal wavelets.

3.7.2 The Redundant Wavelet Transform

In the original FWT we sub-sample the coefficients after each filtering step. As a result, the number of coefficients is not the same at the different levels. Although this reduction in the number of coefficients is critical in some applications (such as data compression), it is not a key factor in image enhancement [Xu94, Malfait97]. In image processing applications we normally prefer to have the same number of coefficients in all levels, which makes it easier to compare and link the coefficients across the scales. This

can be obtained by removing the down-sampling stage of the FWT. The resulting transform is normally called the redundant FWT [Xu94, Malfait97, Strickland97].

3.8 Using the Wavelets

Choosing a proper wavelet is the first step in wavelet analysis. The selection of a wavelet depends mainly on the application. We will discuss about this topic in Chapters 4 and 5, when we explain real applications of wavelet analysis. After selecting the proper wavelet, Mallat's FWT algorithm can be used to analyse (and synthesise) the signal.

To implement Mallat's FWT algorithm, the quadrature mirror filter coefficients (Section 3.5.1) associated with the chosen wavelet are required. These filter coefficients are usually given directly in the literature [Mallat98]. However, for some wavelets, they are not given directly and should be derived from other coefficients.

As we mentioned before, there is a very close relationship between wavelet and subband coding theory. In the subband coding scheme, the role of the wavelet is played by the high-pass filter, $h(n)$. In wavelet theory, there is another function which has a close relation to the low-pass filter, $g(n)$. This function, $\Phi(n)$, which is called the scaling function or father wavelet, satisfies the fundamental relation (twin scale relation) [Matlab96]

$$\frac{1}{2}\Phi(x/2) = \sum_{n\in Z} w_n \Phi(x-n), \qquad (3.7)$$

which explains the relation between the scaling functions at different scales. Given the w_n coefficients, the low-pass and high-pass filter coefficients can be derived.

The sum and the norm of the w_n sequence are 1 and $1/\sqrt{2}$ respectively. This sequence can be considered as the coefficients of a low-pass filter $W(n)$. The decomposition filter coefficients can be obtained from the $W(n)$ coefficients using the following equations [Daubechies92, Matlab96]:

$$g(n) = \frac{W(n)}{||W||},$$

$$h(n) = (-1)^n g(L-1-n).$$

For orthogonal wavelets, the reconstruction filters are the same as the decomposition filters within a time reversal (Eqn 3.4):

$$h'(n) = h(L-1-n),$$
$$g'(n) = g(L-1-n).$$

However for biorthogonal wavelets, the scaling function associated with the synthesis phase of the transform, $\tilde{\Phi}(x)$, is different from the analysis scaling function $\Phi(x)$. This generally leads to a different sequence of $\tilde{w}_n$ in the twin scale equation (Eqn. 3.7) and hence different synthesis filters.

3.9 Summary

This chapter acts as a mathematical basis for the methods discussed in the next few chapters. First, the Fourier and Short-Time Fourier Analysis were reviewed and the basics of the wavelet analysis were presented. Next, the subband coding technique was described and shown to be a different view of the wavelet theory. Finally, a fast wavelet transform technique, based on the subband coding technique, was explained. This fast wavelet

transform algorithm is used extensively in the wavelet-based techniques discussed in Chapters 4 and 5.

Chapter 4

Wavelet-Based Denoising Filters

4.1 Introduction

During recent years, various wavelet-based denoising filters have been proposed as alternatives to Fourier domain filters [Weaver91, Mallat92, Xu94, Donoho95, Malfait97]. Since the wavelet transform retains both the spatial and frequency domain information, it offers denoising opportunities that the Fourier transform does not.

A typical wavelet-based denoising algorithm is composed of three steps: computing the wavelet transform of the signal, manipulating the coefficients, and finally reconstructing the filtered signal from the new coefficients. The difference between the various denoising methods is mainly in the second step; the way they modify the coefficients.

In this chapter some of the proposed wavelet domain noise filters are discussed and simulated. We use both qualitative and quantitative measures to compare the performance of these methods. In Chapter 5 we shall compare the results from these filters with the results from some new wavelet denoising filters we have proposed.

As with Chapter 2, we shall use flat regions of the image to estimate the noise level and leave the practical measurement of noise to Chapter 6.

4.2 Basic Thresholding Method

4.2.1 Introduction

The basic thresholding method was first introduced by Weaver *et al.* [Weaver91] to remove noise from images, As was shown in Chapter 3, the wavelet transform can split a signal into different bands of frequency without losing the spatial domain information. This important feature enables us to adjust the filter parameters to the local characteristics of the image. However, the basic thresholding method did not involve this spatial information in the noise filtering process. In this aspect, this method is very close to the frequency domain techniques using the wavelet transform simply as a tool to identify the image power at different frequency ranges.

The signal to noise ratio of the image can be improved by suppressing the wavelet coefficients associated with the high frequency components. This method is similar to the frequency domain high-pass filtering. However, it introduces less artefacts into the image because the basis functions (wavelets) have local support in both time and frequency. As a result, if by mistake we suppress a noise-free coefficient or insufficiently suppress a noisy coefficient, the introduced artefacts can not spread over the whole range of the restored signal [Weaver91, Mallat92].

The basic thresholding method is similar to a frequency domain method called spectral pruning or subtraction [Wahl87, Lim90]. In the spectral pruning method, first we estimate the noise power from the high frequency parts of the spectrum. We restore the signal spectrum by subtracting the noise power from the noisy image spectrum with the assumption that the noise power is constant over the whole range of the frequencies. In practice, the amount of spectral subtraction should not be the same as the estimated noise

power with a multiplicative factor used to control the amount of spectral subtraction. A large multiplicative factor (a large spectral subtraction) will lead to a better noise removal but more blurring and ringing at edges [Wahl87].

Similarly, in the basic wavelet thresholding method, we can subtract the noise power from the wavelet coefficients of the noisy image. However, first we need to estimate the noise power in the wavelet domain.

The two-dimensional wavelet transform decomposes an image into a hierarchy of approximation and detail images [Mallat89A]. Each detail image is associated with the frequency components in a specific band of frequency (scale) and direction (Horizontal, Vertical or Diagonal). To subtract the noise power from the detail images, we need to estimate the noise power for each detail image separately.

In practical images, there are almost always signal-free regions near the boundaries of the image [Weaver 91, Xu94]. We can use these regions to characterise the noise in the image adjusted, as necessary, for the noise bias discussed in Chapter 6. The energy of each pixel is spread over the whole Fourier coefficients in the frequency space. This is not the case in the wavelet domain where the energy of each pixel is spread over only the few wavelet coefficients located in a small neighbourhood around that pixel. Therefore, the wavelet coefficients in a region close to the boundaries of a detail image can be used to estimate the noise power in that detail image. To control the amount of noise power subtraction, as with the spectral pruning method, we use a multiplicative factor that controls the trade-off between noise reduction and image resolution. Although this method of power subtraction in the wavelet domain does not blur the edges significantly, it can

still affect the small structures in the image or cause some local artefacts around the edges of any object in the image.

The subtraction of the noise power from the spectrum of the signal in the wavelet domain can be implemented in two different ways -hard and soft thresholding which are discussed in the next section.

4.2.2 Hard and Soft Thresholding Methods

To detect the noisy pixels in each detail image, we set a threshold value and treat all pixels with a wavelet coefficient amplitude below that threshold as noisy. The threshold for each detail image is a function of the noise power in the detail image. For the detail image D_j^{DIR} (where j is the decomposition level and DIR can be horizontal, vertical or diagonal), the threshold value can be written as [Weaver91, Donoho95]:

$$T_j^{DIR} = k_{mult} P_j^{DIR}(noise) \qquad (4.1)$$

where k_{mult} is a multiplicative factor which controls the trade off between noise reduction and image resolution, and $P_j^{DIR}(noise)$ is the energy of noise (estimated over a uniform region) for detail image D_j^{DIR}.

To implement the noise-filtering algorithm, the estimated threshold can be used in two different ways. The resulting methods are called hard and soft-thresholding techniques [Weaver91, Donoho95].

In the hard-thresholding method, the wavelet coefficients below the given threshold value are simply discarded, while all other coefficients remain unchanged. In other words, the filtered coefficients, $\hat{D}_j^{DIR}$, for the detail image D_j^{DIR} can be expressed as [Weaver91]:

$$\hat{D}_j^{DIR}(n_x,n_y)=\begin{cases} D_j^{DIR}(n_x,n_y), & if \quad D_j^{DIR}(n_x,n_y) > T_j^{DIR} \\ 0, & if \quad -T_j^{DIR} \le D_j^{DIR}(n_x,n_y) \le +T_j^{DIR} \\ D_j^{DIR}(n_x,n_y), & if \quad D_j^{DIR}(n_x,n_y) < -T_j^{DIR} \end{cases}$$

$$n_x = 0,1,\ldots,N_x \quad \text{and} \quad n_y = 0,1,\ldots,N_y$$

where D_j^{DIR} is the detail image in direction *DIR* at level *j* and T_j^{DIR} is the estimated threshold value for this detail image (Eqn. 4.2).

In the soft-thresholding method, after removing all the wavelet coefficients whose absolute values are less than the threshold, we subtract the threshold amplitude (a function of the noise power) from the other wavelet coefficients. After soft thresholding, the coefficients of the detail image *DIR* at level *j* are [Weaver91, Matlab96]:

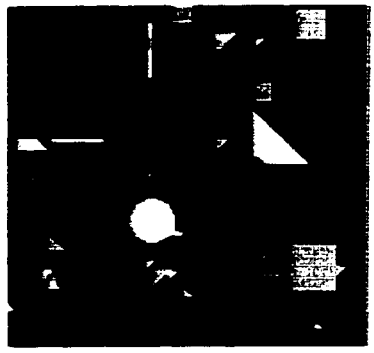
$$\hat{D}_j^{DIR}(n_x,n_y)=\begin{cases} D_j^{DIR}(n_x,n_y)-T_j^{DIR}, & if \quad D_j^{DIR}(n_x,n_y) > T_j^{DIR} \\ 0, & if \quad -T_j^{DIR} \le D_j^{DIR}(n_x,n_y) \le +T_j^{DIR} \\ D_j^{DIR}(n_x,n_y)+T_j^{DIR}, & if \quad D_j^{DIR}(n_x,n_y) < -T_j^{DIR} \end{cases}$$

The hard-thresholding method causes some discontinuities in the range of the modified wavelet coefficients as the modified coefficients are not allowed to have a value in the range $-T_j^{DIR}$ and $+T_j^{DIR}$ except for zero. The modified coefficients in the soft-thresholding method, however, are spread over a continuous range [Weaver91, Donoho95]. As we will see in the next section, the soft-thresholding method removes the noise more effectively.

4.2.3 Implementation and Results

A typical wavelet based noise filtering method is composed of three steps: *Decomposing* the image into multi-scale wavelet coefficients, *Modifying* the coefficients, and *Reconstructing* the filtered image from the modified coefficients.

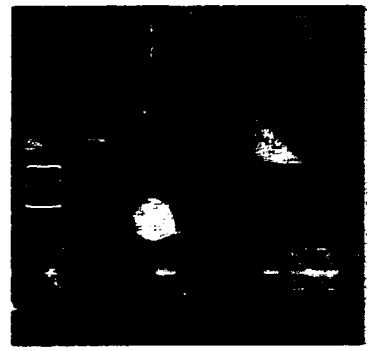
To decompose the image, we need a proper wavelet and also an algorithm to calculate the coefficients. Weaver *et al.* [Weaver91] used a one-dimensional approach to decompose the image into its wavelet coefficients. First they applied a one-dimensional transform to the rows of the image and, after thresholding the rows, they repeated the process for the columns. In our implementation, we directly calculated the two-dimensional wavelet transform. For this purpose we used the Mallat fast wavelet transform algorithm (see Section 3.6). A redundant transform (see Section 3.7.2) is used to calculate the wavelet coefficients so that the detail images at different levels have the same number of coefficients. Using a redundant transform makes it easier to relate the signal-free region in the original image with the corresponding regions in the detail images. Because of the simplicity of the method, there is almost no limitation on the choice of the wavelet. Any wavelet with a local support can be used to decompose the image in this method. However, smooth wavelets lead to better results, because the artefacts caused by excessive suppression of the coefficients have a form similar to the wavelet waveform. Weaver *et al.* [Weaver91] used a set of orthogonal wavelets first described by Lemarie *et al.* [Lemarie86]. In our implementation we used the Bior.3.3 wavelet [Cohen92] which has a smoother reconstruction wavelet. To test the method, we used the 128×128 synthetic image shown in Fig. 4.1(a) the range of grey levels for this



(a)- Original

(b)- Noisy

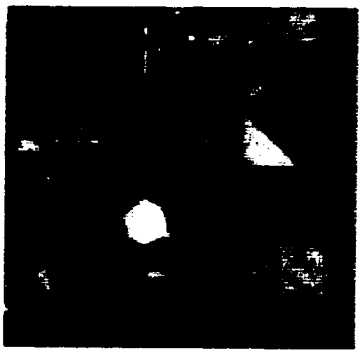
(c)- Hard Thresholding
Tr. = $1.5\sigma'_n$



(d)- Hard Thresholding
Tr. = $2.5\sigma'_n$

Figure 4.1 *Applying the basic thresholding techniques to a noisy synthetic image: (a) Original synthetic 128×128 image with the range of grey levels between 0 and 255. (b) After adding Gaussian white noise with* $\sigma_n^2 = 400$ *(grey level)*^{2}. *Results of: (c) Hard thresholding with Tr.* $= 1.5\sigma'_n$, σ'_n *is the estimated value for* σ_n *(d) with Tr.* $= 2.5\sigma'_n$.

(e)- Soft Thresholding
Tr. = $1.0\sigma'_n$



(f)- Soft Thresholding
Tr. = $1.5\sigma'_n$

(g)- Soft Thresholding
Tr. = $2.0\sigma'_n$

(h)- Soft Thresholding
(Bior. 5.5 Wavelet)

Figure 4.1 *(Continued) Results of: (e) Soft thresholding with Tr. =* $1.0\sigma'_n$. *(f)* $Tr. = 1.5\sigma'_n$. *(g)* $Tr. = 2.0\sigma'_n$. *(h)* $Tr. = 1.5\sigma'_n$. *using a different type of wavelet for analysis (Bior. 5.5 instead of Bior. 3.3 [Cohen92]).*

picture was between 0 and 255. As discussed earlier we avoid the bias in estimating the noise by using a uniform region added to the image prior to introduction of the noise.

A zero mean Gaussian noise with variance of 400 (grey levels)^2 was added to the image. The result is shown in Fig. 4.1(b).

Figs. 4.1(c) and 4.1(d) are obtained using the hard thresholding method. In Fig. 4.1(c) the threshold is chosen as $1.5\sigma_{noise}$, which means, any wavelet coefficient with an absolute value less than $1.5\sigma_{noise}$ is set to zero before reconstruction in the detail images. The standard deviation of noise, σ_{noise}, is estimated for each detail image individually. The filtered image (Fig. 4.1(c)) is still quite noisy and the SNR improvement is only 2.98 dB (see Table 4.1). Fig. 4.1(d) shows the result of using a higher threshold ($2.5\sigma_{noise}$). The noise is smoothed more effectively (SNR improvement is 5.45 dB). However in some locations, such as the one marked on the image, some noisy pixels are almost left unchanged. The strong difference between these noisy pixels and their neighbouring pixels are caused by the discontinuity in the range of modified wavelet coefficients. Small coefficients (smaller than the threshold) are mapped to zero but large coefficients remained unchanged after filtering. Therefore this method will emphasise the difference between the pixels with very strong noise (for which the wavelet coefficients remain unchanged) and their cleaner neighbouring pixels (for which the coefficients are set to zero). In the soft-thresholding method we shift the large coefficients toward zero to resolve this problem.

Figure	*Description*	*MSE* *(Grey level)²*	*ISNR* *(dB)*
4.1.b	Additive noise , σ^2 = 400 (Grey level)^2	393.05 ± 1.74	0.00 ± 0.00
4.1.c	Hard with Tr. = 1.5 σ, Wavelet = Bior. 3.3	197.8 ± 1.28	2.98 ± 0.02
4.1.d	Hard with Tr. = 2.5 σ,	112.4 ± 0.85	5.43 ± 0.04
4.1.e	Soft with Tr. = 1.0 σ,	126.7 ± 0.41	4.92 ± 0.01
4.1.f	Soft with Tr. = 1.5 σ,	128.6 ± 0.57	4.85 ± 0.02
4.1.g	Soft with Tr. = 2.0 σ,	166.3 ± 1.12	3.74 ± 0.02
4.1.h	Soft with Tr. = 1.0 σ, Wavelet = Bior 5.5	130.6 ± 0.76	4.79 ± 0.02

Table 4.1 *Quantitative measures (MSE and ISNR) for the results obtained by applying the basic thresholding method to the noisy synthetic image shown in Figure 4.1(b).*

The shifting of the coefficients in the soft thresholding approach is simply done by subtracting the threshold value from the large coefficients. Figs. 4.1(e) and 4.1(f) give the result of using the soft-thresholding method. In Fig. 4.1(e) the selected threshold value of $1.0\sigma_{noise}$ is not large enough. As a result the noise is not removed effectively from this image. On the other hand, using large threshold values can blur the image ($2.0\sigma_{noise}$ for Fig. 4.1(g)). The threshold value used in Fig. 4.1(f) is $1.5\sigma_{noise}$, which provides a better trade off between noise smoothing and edge sharpness (see Table 4.1). Again, σ_{noise}, the standard deviation of noise, is estimated for each detail image individually.

In both hard and soft-thresholded images (Figs. 4.1(d) and 4.1(f)), the major edges are not degraded significantly. However, by comparing the figures it can be seen that, unlike the hard-thresholding method, the soft-thresholding method is more successful in removing the noise from the strongly corrupted pixels.

In Fig. 4.1(h) we used a different type of wavelet (Bior.5.5 [Daubechies92, Cohen92]) to decompose the image. The result looks very similar to the result obtained using the

Bior.3.3 wavelet (Fig. 4.1(f)). This similarity is also confirmed by almost equal MSE measures associated with these two images (see Table 4.1).

In Fig. 4.2, both thresholding methods are applied to a natural test image. The results obtained for this test image are consistent with the previous results obtained for a synthetic image (see Table 4.2).

The MSE and ISNR for the images shown in Figs. 4.1 and 4.2 are represented in Table 4.1 and 4.2. The quantitative measures are slightly better for the hard thresholded images than the soft thresholded ones. However, it is not sufficient to rely on quantitative measures since they can not take into account all aspects of the image quality for which the human eye is sensitive. Although the soft thresholded images are more blurred than their hard thresholded counterparts, they show a better trade-off between noise filtering and image blurring (Figs. 4.1 and 4.2).

Figure	*Description*	*MSE* *(Grey level)*^{2}	*ISNR* *(dB)*
4.2.b	Additive noise , $\sigma^2 = 400$	384.03 ± 1.65	0.00 ± 0.00
4.2.c	Hard with Tr. = 2.5 σ. Wavelet = Bior. 3.3	101.6 ± 0.83	5.77 ± 0.04
4.2.d	Soft with Tr. = 1.5 σ. Wavelet = Bior. 3.3	105.5 ± 0.77	5.61 ± 0.03

Table 4.2 *Quantitative measures (MSE and ISNR) for the results obtained by applying the basic thresholding method to the noisy natural image shown in Figure 4.2(b).*

The basic thresholding methods have serious drawbacks. First of all, there is no automatic mechanism for detecting the optimum threshold level. The threshold is set by trial and error. The threshold must be large enough to remove the noise effectively, but not so large as to blur the image. Since most noise power is confined to the small-scale

(a)- Original

(b)- Noisy

(c)- Hard Thresholding

(d)- Soft Thresholding

Figure 4.2 *Applying the basic thresholding techniques to a noisy natural image: (a) Original 256×256 image with the range of grey levels between 0 and 255. (b) After adding Gaussian white noise with* $\sigma_n^2 = 400$ *(grey level)*^{2}. *Results of: (c) Hard thresholding with Tr.* $= 2.5\sigma'_n$, *where* σ'_n *is the estimated value for* σ_n. *(d) Soft thresholding with Tr.* $= 1.5\sigma'_n$.

detail images reducing the coefficients in these images will reduce the noise preferentially. However to keep the image sharp, the small scale information is required. Therefore, any attempt to remove the noise can affect the sharp features in this filtering method.

The basic thresholding method is a crude approach similar to the frequency domain techniques in which we reduce the noise preferentially by suppressing the high frequency components. The finite support of the basis functions in the wavelet domain gives us the opportunity to process the image in a different way where the filter behaviour is adapted to the local features of the image. In the rest of this chapter, we look at some more advanced wavelet domain techniques that adapt the filter to the spatial content of the image.

4.3 Spatially Selective Wavelet Domain Filters

4.3.1 Introduction

The basic goal in a noise filtering technique is to suppress the noise without blurring the image. In an advanced noise removal technique, edges and other sharp features should be recognised as different from the noise and retained. As we saw in Chapter 3, Fourier analysis has a serious drawback. In transforming to the frequency domain, spatial information is lost. One major advantage afforded by the wavelet transform, not exploited in the Fourier method, is the ability to treat different regions of an image, i.e. edges, smooth areas and so on, differently.

In wavelet analysis, the signal power at small scales corresponds to high frequency components and is required to keep edges sharp in an image. By removing the high frequency components from the smooth regions and retaining them around identified

edges, noise is reduced without blurring any identified edges. The key to this technique is to identify edges and other sharp features in a noisy image.

The idea of using the scale-space correlation of a signal to filter the noise was first introduced by Witkin [Witkin83]. He developed an algorithm to distinguish major edges from noise in finer scales. Several more advanced noise filtering techniques, based on the wavelet and sub-band decomposition techniques, have been proposed in recent years [Mallat92, Xu94, Malfait97]. In the following sections, we briefly explain some of the recent advanced wavelet-based filtering methods.

4.3.2 Mallat's Method

The most important information of an image is often carried by its discontinuities and irregular structures. Edges, for instance, are among the most important features of an image for recognition purposes. In a noisy image, noise adds some new irregular structures to the image, in addition to the inherent irregularities.

For most practical images, the built-in irregularities have different characteristics from those introduced by the noise. Mallat *et al.* [Mallat92] used these differences to distinguish and remove the noise. The local regularity of a signal is often measured by Lipschitz exponents. In the rest of this section, after a brief review of this regularity measure and definition of the modulus maxima, Mallat's denoising algorithm [Mallat 92] is explained.

The Regularity Measures

The local regularity of a function is usually measured with Lipschitz exponents [Mallat92, Malfait97]. A function has an order of smoothness (or Lipschitz exponent) of α $(0 < \alpha \leq 1)$ at point x_0, if and only if there exist two constants A and h_0 $(0 < h_0 < 1)$, such that for any $h < h_0$:

$$|f(x_0) - f(x_0 + h)| \leq A|h|^{\alpha}.$$

The Lipschitz regularity can be extended to α values greater than unity (or less than zero). When extending the definition of the Lipschitz exponents, we examine the behaviour of the derivative (or integral) of the signal in a neighbourhood of x_0 [Mallat92]. In general, a larger Lipschitz exponent, α, at a point x_0 is associated with smoother behaviour of the signal in a neighbourhood of this point. For example a step signal has a Lipschitz exponent of zero at $x_0 = 0$, while a ramp signal has a Lipschitz exponent equal to unity. For signals with smoother behaviours, the Lipschitz exponent is larger. For example, $y = x^2$ has a smoothness of degree 2 ($\alpha = 2$) at $x_0 = 0$. For the signals with a more irregular behaviour than a step function, the Lipschitz exponent is less than zero. As an example, the delta function has an order of smoothness of -1 at $x_0 = 0$.

The definition of Lipschitz exponents for two-dimensional signals is a simple extension of one-dimensional case [Mallat92]. A function $f(x, y)$ has an order of smoothness (or Lipschitz exponent) of α $(0 \leq \alpha \leq 1)$ at point (x_0, y_0), if and only if there exist two constants $k_0 > 0$ and $h_0 > 0$, for $A > 0$, such that for any $h < h_0$ and $k < k_0$ [Mallat92]:

$$|f(x_0 + h, y_0 + k) - f(x_0, y_0)| \leq A(h^2 + k^2)^{\alpha/2}.$$

In a noiseless image, the worst singularities are often discontinuities (edges) for which the local order of smoothness is equal to zero. Hence, the Lipschitz regularity is almost always non-negative for a noiseless image. On the other hand, noise generated singularities will have Lipschitz exponents that are almost always negative [Mallat92]. By computing the local regularity of the image, we can detect the noisy pixels. It is usually easier to calculate the local regularity of the signal in the scale space than the spatial domain [Mallat92, Daubechies92]. In fact, it can be proved that the local Lipschitz exponent of a function can be calculated from the behaviour of its wavelet coefficients across the successive scales [Mallat92].

Wavelet Transform Modulus Maxima

For one-dimensional signals, a local wavelet modulus maximum is a wavelet coefficient at scale *s* whose modulus (absolute value) is greater than all other coefficients in its neighbourhood. Mallat *et al.* [Mallat91] proved that almost all the information in a signal is carried by its local modulus maxima. Although the modulus maxima can not uniquely characterise the original signal, they can be used to reconstruct a close approximation of the signal.

The modulus maxima carry all the information about the singularities of a function. In the vicinity of a singularity, there are always some local maxima at the fine scales of the wavelet transform and possibly even present in the coefficients in some coarser scales [Mallat92]. A maxima line is a line that connects the corresponding maxima at different scales. All singularities of a signal can be located by following the maxima line from the coarser scales to the finest scale. To compute the order of smoothness of a singularity, we

measure the decay rate of the modulus maxima along a maxima line. The Lipschitz exponent is derived by fitting an exponential function, $A.s^{\alpha}$, to the curve which shows the decay of modulus maxima across scales [Mallat91, Mallat92].

For two-dimensional signals the definition of the modulus maxima is slightly different:

$$Modulus(j,n_x,n_y) = (|D_j^H(n_x,n_y)|^2+|D_j^V(n_x,n_y)|^2)^{1/2}. \qquad (4.2)$$

However, to detect the singularities of the signal and to measure their regularity, the same maxima linkage approach can be used [Mallat91, Mallat92].

One-Dimensional Signal Denoising

Wavelet transform maxima carry almost all the information of a signal [Mallat91, Mallat92]. To process a signal we can modify its wavelet transform maxima and reconstruct the processed signal from the new maxima.

Noise generates singularities with negative Lipschitz exponents, while the worst singularities in practical signals are steps for which the Lipschitz exponent is zero. This property is used by Mallat *et al.* [Mallat92] to discriminate the noise from the signal. In Mallat's method, we examine the evolution of the wavelet maxima across the scales. A negative Lipschitz exponent is characterised with a strong increase in the amplitude of maxima when scale decreases. Such maxima are associated with noise and must be removed from the signal. However, the presence of noise can change the overall regularity to a negative number in the locations where the signal has positive Lipschitz exponents. To detect the noise from the signal we must examine the maxima at larger scales since the influence of noise decreases with increasing the scale. However, the presence of maxima at

larger scales is associated with the presence of a discontinuity in the original signal and therefore the maxima chain must be preserved.

In order to be able to examine the behaviour of the maxima across scales, we need to detect the maxima lines. A maxima line is a line in the scale space domain that connects all the maxima associated with the same singularity. Increasing the scale decreases the number of maxima. To generate the maxima lines, we have to decide which maxima propagate to a larger scale and which ones do not. For this purpose, one can compute the wavelet transform on a larger number of scales, instead of using the dyadic scales. To save computations however, Mallat *et al.* [Mallat92] used an *ad hoc* algorithm: "A maximum is considered to propagate from scale 2^j to a coarser scale 2^{j+1} if it has a large amplitude and it is close to a maximum at the scale 2^{j+1} having the same sign".

Mallat's denoising algorithm removes all maxima whose amplitude increases on average when the scale decreases, or which do not propagate to larger scales. The wavelet coefficients in the finest scale, 2^1, are mostly dominated by the noise and hence Mallat preferred to ignore them completely. Removing all the maxima in the finest level will greatly reduce the noise effect but the resultant signal will be completely blurred. Mallat *et al.* suggested that we should create a maximum at the scale 2^1 at every location where there exists a maximum at the scale 2^2. To compute the amplitude of these maxima, we calculate the decay rate of the maxima at the corresponding location for the scales larger than 2^2. The size of maxima at level 2^1 is chosen to be consistent with this decay rate.

After selecting the proper maximum, the denoised signal is reconstructed from the new set of maximum using the alternative projection algorithm [Mallat91, Mallat92].

Two-Dimensional Signal (Image) Denoising

Mallat *et al.* discriminated 2D noise singularities from the image singularities [Mallat92] using a similar approach as the one-dimensional case. The modulus maxima (Eqn. 4.2) at corresponding positions in different scales have to be linked in so-called maxima lines. To relate the modulus maxima across the scales Mallat [Mallat 92] used a similar *ad hoc* approach as in one-dimensional case. "A modulus maximum is considered to propagate to a coarser scale, only if there is another modulus maximum close to its position in the coarser scale which has a similar angle". The angle of a modulus maximum is defined as:

$$Angle(j, n_x, n_y) = Arctan(D_j^H(n_x, n_y) / D_j^V(n_x, n_y)).$$

The decay rates of maxima along these maxima lines are used to discriminate noise from the image singularities. The worst non-noise related singularities are discontinuities with Lipschitz regularity of zero in the regions where the image does not have irregular textures (see "Regularity Measures" in section 4.3.2). The regularity for the noise singularities is almost everywhere negative. Therefore, it is possible to discriminate the noise singularities from those singularities inherent in the image by examining the evolution of the maxima along scales.

As with the one-dimensional case, the two-dimensional denoising algorithm removes all modulus maxima that do not propagate to coarser scales, or have a negative Lipschitz regularity. Unlike the one-dimensional case, the geometrical properties of the image are used to remove all the remaining maxima that can not be linked in a chain [Mallat92]. A threshold can be selected for the length of the chains and any chain with a length smaller than this threshold is deleted. Again this algorithm replaces the strongly corrupted maxima

in the finest level with a new set of maxima. As with the one-dimensional case, we put a new maximum at any location where there is a maximum at the second finest scale. The magnitude of this new maximum is estimated from the decay rate of the corresponding maxima line.

4.3.3 Xu's Method

Although Mallat's method [Mallat92] is quite effective in improving the visual quality and signal to noise ratio of noisy images, it is computationally expensive. To avoid this expense Xu *et al.* [Xu94] proposed a faster technique, which is reviewed in this section.

To keep the edges sharp, a noise filtering technique must be able to discriminate the edges from noise. In the noise filtering technique suggested by Xu *et al.* [Xu94], a method similar to the Rosenfeld technique [Rosenfeld70, Rosenfeld71] is used to detect the major edges. This algorithm is based on the observation that the correlation between the coefficients at different scales is much larger for the coefficients that are associated with major edges. Xu *et al.* [Xu94] used the direct multiplication of the coefficients at adjacent scales to locate the location of edges. Although this method is not as accurate as the method used by Mallat [Mallat92], it is quite straightforward and much easier to implement. After finding the edges, the small-scale coefficients are retained in the vicinity of the identified edges and suppressed in other regions. The small-scale coefficients correspond to the high frequency data in the Fourier transform domain, and hence this filter acts as a spatially selective low-pass filter. High frequency data are mostly suppressed, except where a highly correlated feature (an edge) is detected. This locally

adaptive filter is capable of removing the noise from the smooth regions without blurring the edges.

Implementation and the Results

To implement Xu's noise filtering method [Xu94], we need to first decompose the image into its multi-scale wavelet coefficients. The selected wavelet should behave as a good edge detector, because it is appropriate to have large responses (coefficients) in the vicinity of the edges in each scale. An optimal edge detector for step edges has a shape very close to the first derivative of a Gaussian function [Canny86]. To implement their algorithm, Xu *et al.* [Xu94] used the biorthogonal quadratic-spline wavelets first introduced by Mallat and Zhong [Mallat89]. These wavelets have a shape very close to the derivative of a Gaussian function and hence behave like an optimum edge detector. In our implementation, we used the same wavelet as that used by Xu *et al.*. To examine the sensitivity of the method to the type of wavelet we also have checked the results for the Bior.3.3 wavelets first introduced by Cohen and Daubechies [Cohen92].

To compute the wavelet coefficients we used Mallat's fast wavelet transform algorithm (Section 3.6 and [Mallat89A]). A redundant transform (Section 3.7.2) is used to calculate the wavelet coefficients so that the detail images at different levels have the same number of coefficients. This equality makes it easier to calculate the correlation between scales through direct multiplication of the coefficients.

After decomposing the image into a hierarchy of horizontal, vertical and diagonal detail images, we compute the direct spatial correlation, $Corr_l^{DIR}$ on each group of detail images to locate the edges and other important features.

$$Corr_l^{DIR}(m,n_x,n_y)=\prod_{i=0}^{l-1} D_{m+i}^{DIR}(n_x,n_y), \qquad (4.3)$$

$$1\le n_x \le N_x, \quad 1\le n_y \le N_y, \quad 0<m<M-l+1$$

where l is the number of adjacent scales involved in computing the correlation and m is the finest scale involved in this computation. D_j^{DIR} is the detail image at direction DIR (Horizontal, Vertical or Diagonal) at level j (Section 3.6).

Xu *et al.* suggests that at most three scales should be involved in the edge detection process ($l \le 3$) for the best filtering results. To generate the results given in the paper, [Xu94], they calculated the correlation only over two scales. Therefore in our simulation of Xu *et al.* work, we also detect the location of edges by directly multiplying each detail image by a detail image in the next coarser scale. That is, if $D_j^{DIR}(n_x,n_y)$ is a detail image in level j we calculate:

$$Corr_2^{DIR}(j,n_x,n_y)=D_j^{DIR}(n_x,n_y).D_{j+1}^{DIR}(n_x,n_y).$$

Sharp edges generate large coefficients over many wavelet scales, but the noise coefficient dies out swiftly with increasing scale. The direct multiplication of coefficients enhances major edges while suppressing noise and small features. Therefore by comparing the coefficients in the detail image, $D_j^{DIR}(n_x,n_y)$, with the correlation image, $Corr_2^{DIR}(j,n_x,n_y)$ it is possible to find which detail coefficients belong to the edges. To make the comparison of the detail images with the correlation images meaningful [Xu94], first we need to re-scale the power of the correlation function, $Corr_2^{DIR}(j,n_x,n_y)$, to that of $D_j^{DIR}(n_x,n_y)$ that is:

$$\overline{\overline{Corr_2^{DIR}(j,n_x,n_y)}} = Corr_2^{DIR}(j,n_x,n_y) \times \sqrt{\frac{PD_j^{DIR}}{PCorr_2^{DIR}(j)}},$$

where:

$$PD_j^{DIR} = \sum_{n_x}\sum_{n_y}[D_j^{DIR}(n_x,n_y)]^2 \quad and \quad PCorr_2^{DIR}(j) = \sum_{n_x}\sum_{n_y}[PCorr_2^{DIR}(j,n_x,n_y)]^2 .$$

After rescaling the power of the correlation image, the major edges are identified by comparing the absolute values of $\overline{\overline{Corr_2^{DIR}(j,n_x,n_y)}}$ and $D_j^{DIR}(n_x,n_y)$. An edge is identified at any position (n_x,n_y) at which $|\overline{\overline{Corr_2^{DIR}(j,n_x,n_y)}}|$ is greater than $|D_j^{DIR}(n_x,n_y)|$. All the detail coefficients at these positions are passed through the filter.

To extract less important edges, the whole process should be repeated on the remaining coefficients. First we extract the coefficients located on identified edges from both detail and correlation images by resetting the values of these coefficients to 0's at the positions identified. Then we re-scale the power of the modified correlation image to that of the modified detail image and compare their absolute values to identify and extract the second most significant edges. This procedure of power normalisation, data value comparison, and edge information extraction can be iterated many times until the power of the unextracted data points in $D_j^{DIR}(n_x,n_y)$ is nearly equal to the estimated noise power at the detail image.

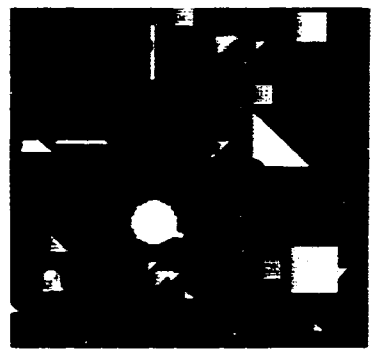
After filtering, the wavelet coefficients in the vicinity of the edges will be preserved while they are completely suppressed in the smooth regions of each detail image. This spatially selective low-pass filter removes most of the noise power without blurring the edges.

The above filtering process must be repeated for all detail images in all scales. After modifying all detail images, we use the inverse wavelet transform to reconstruct the filtered image.

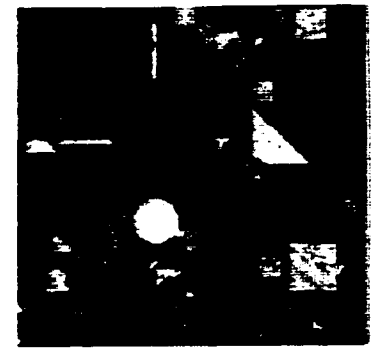
Fig. 4.3(a) represents a 128×128 synthetic test image with a maximum grey level of 255. The image in Fig. 4.3(b) was derived from this test image by adding a zero-mean Gaussian noise with variance 400 (grey level)^2. Mallat's fast wavelet transform was used to decompose the image.

Fig. 4.3(c) shows the result obtained using Xu's noise filtering method. For this purpose, a six-level wavelet decomposition is calculated and the coefficients in the first five scales are filtered. To estimate the location of the major edges, only two adjacent levels are involved in calculating the direct spatial correlation.

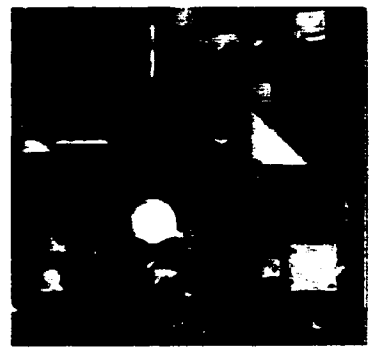
Compared to the basic thresholding method (Figs. 4.1(c) and 4.1(f)), Xu's method is more successful in preserving the edges of this test image. The average SNR is improved by 5.15 dB (see Table 4.3). Although most parts of the image are restored successfully, some small features are not well recovered. These small, low-contrast features do not have a strong spatial correlation across scales and hence Xu's method cannot discriminate them from the noise. In the neighbourhood of edges, where there is a high correlation between scales, a great amount of noise is passed through the filter.



(a)- Original



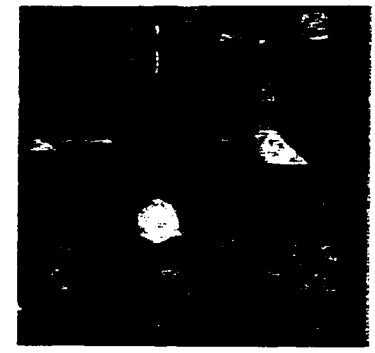
(b)- Noisy



(c)- Xu's Method
Rejected Power = Noise Power

(d)- Xu's Method
Rejected Power = 1.2×Noise Power

(e)- Xu's Method
Rejected Power = 0.9×Noise Power



(f)- Xu's Method
3 layer correlation

Figure 4.3 *Applying Xu's noise filtering method to noisy synthetic image:*
(a) Original synthetic 128×128 image with grey levels between 0 and 255.
(b) After adding Gaussian white noise with variance 400 (grey level)^2
Results of: (c) Xu's method, Rejected Power=Noise Power , 2 layer Correlation.
(d) Xu's method, Rejected Power=1.2×Noise Power, 2 layer Correlation.
(e) Xu's method, Rejected Power=0.9×Noise Power, 2 layer Correlation.
(f) Xu's method, Rejected Power=Noise Power, 3 layer Correlation.

Figure	*Description*	*MSE* *(Grey level)²*	*ISNR* *(dB)*
4.3.b	Additive noise, $\sigma^2 = 400$.	393.05 ± 1.74	0.00 ± 0.00
4.3.c	Xu's method, rejected power = noise power, 2 layer correlation.	120.6 ± 0.67	5.13 ± 0.03
4.3.f	Xu's method, *rejected power=1.2xnoise power*, 2 layer correlation.	121.0 ± 0.64	5.11 ± 0.03
4.3.g	Xu's method, *rejected power=0.9xnoise power*, 2 layer correlation.	136.8 ± 0.73	4.59 ± 0.02
4.3.h	Xu's method, rejected power = noise power, *3 layer correlation*.	122.6 ± 0.58	5.06 ± 0.02

Table 4.3 *Quantitative measures (MSE and ISNR) for the results obtained by applying Xu's noise filtering method to the noisy synthetic image shown in Figure 4.3(b).*

In each iteration of the filtering process Xu's method passes more wavelet coefficients through the filter until the energy of the remaining wavelet coefficients is almost equal to the estimated noise energy. In practice, the ratio of the energy of the remaining coefficients to the estimated energy of noise does not need to be unity. This ratio can be used as a parameter in the filtering process to control the trade-off between the noise removal and the image resolution. The effect of this parameter on the filtered image is shown in Figs. 4.3(d) and 4.3(e). In Fig. 4.3(d) this parameter was chosen to be 1.2 so that few coefficients are passed through the filter. Although the noise is better smoothed in this image, the edges are not as sharp as in Fig. 4.3(c). For the image shown in Fig. 4.3(e) the parameter was set to 0.9. Although the edges are quite sharp in this image, noise is not removed as effectively as in Fig. 4.3(c) or 4.3(d) (see Table 4.3). Therefore by controlling the energy of the suppressed coefficients we can obtain any desired trade-off between image sharpness and noise smoothening.

To generate the previous images (Figs. 4.3(c) to 4.3(e)), only two adjacent scales were used to detect the location of edges. In Fig. 4.3(f), three levels are involved in the edge detection process. The result does not show any improvement over the images obtained using only two levels, which is consistent with the result recorded by Xu *et al.* [Xu94]. In the next chapter, we consider possible improvements to Xu's method, and introduce methods that allow us to involve more scales in the edge detection process.

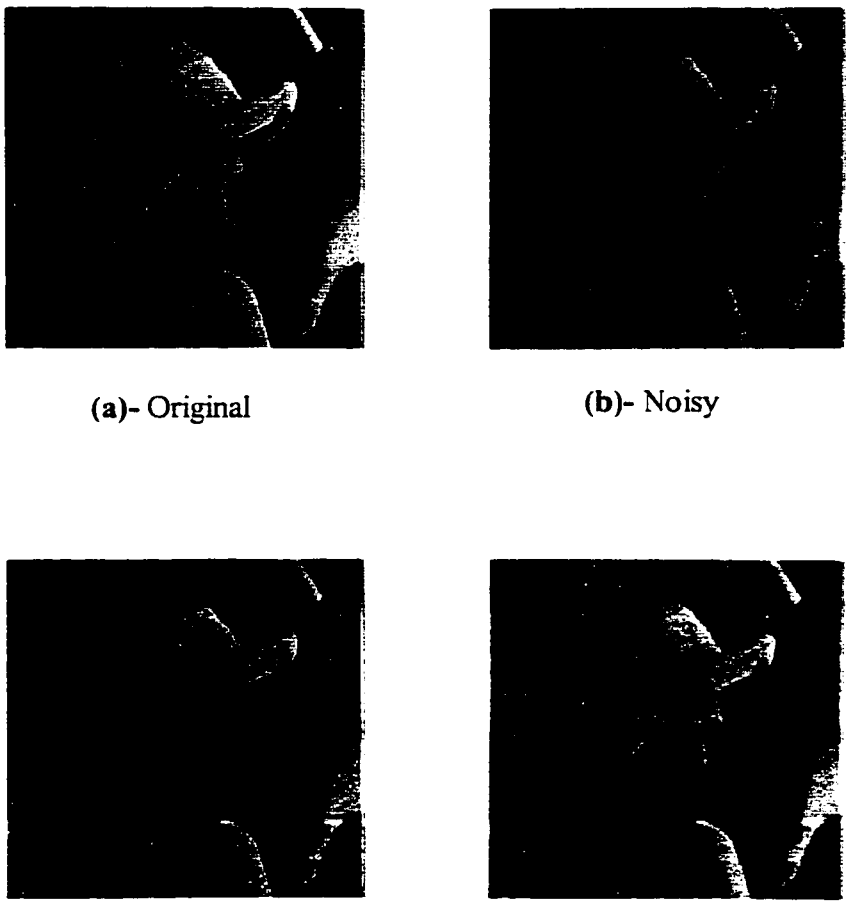
Fig. 4.4(c) represents the result of applying Xu's method to the noisy natural test image shown in Fig. 4.4(b). In comparison to the basic thresholding method, Xu's technique was more successful in removing the noise from the test image. To implement Xu's method the correlation between the detail images was calculated over two adjacent scales. Again, as we can see in Fig. 4.4(d), involving more scales in Xu's correlation operation does not improve the quality of the final result (see Table 4.4).

Figure	*Description*	*MSE* *(Grey level)*^{2}	*ISNR* *(dB)*
4.4.b	Additive Noise , σ^2 =400	384.03 ± 1.65	0.00 ± 0.00
4.4.c	Xu method, Rejected Power = Noise Power, 2 layer Correlation.	97.2 ± 0.48	5.97 ± 0.02
4.4.d	Xu method, Rejected Power = Noise Power, 3 layer Correlation.	103.4 ± 0.73	5.70 ± 0.03

Table 4.4 *Quantitative measures (MSE and ISNR) for the results obtained by applying Xu's noise filtering method to the noisy natural image shown in Figure 4.4(b).*

4.4 Summary

The wavelet transform allows exploitation of both frequency and spatial domain information in the noise filtering process. Several wavelet based noise-filtering



(a)- Original

(b)- Noisy

(c)- Xu's method
2 layer correlation

(d)- Xu's method
3 layer correlation

Figure 4.4 *Applying Xu's noise filtering method to noisy natural image: (a) Original 256×256 image with the range of grey levels between 0 and 255. (b) After adding Gaussian white noise with* $\sigma_n^2 = 400$ *(grey level)*^{2} *. Results of: (c) Xu's method with 2 layer Correlation. (d) Xu's method with 3 layer Correlation.*

techniques

were discussed in this chapter. The performances of the simulated filters in removing the noise from test images were compared using both quantitative and qualitative measures.

Except for the basic thresholding technique, which is a crude filtering method, the general idea behind all the proposed methods is to identify the edges and other sharp features from the noise, and preserve them during low-pass filtering. As a result, the wavelet filters are very effective in preserving the edges. Mallat [Mallat92] used the local regularity measures to detect the edges while Xu *et al.* [Xu94] exploited the correlation between scales for this purpose. In the next chapter, some possible improvements to the above methods are suggested. A new method is proposed which is based on multi-domain processing of images.

Chapter 5

A New Noise Filtering Technique

5.1 Introduction

In this chapter, a new method for removing noise from images together with some new methods for detecting edges are proposed. Preliminary results are given to compare this approach with other methods discussed in this thesis in terms of preserving image sharpness during noise filtering.

The suggested method is basically a wavelet domain technique. However, there are some major differences between the proposed method and other wavelet-based noise filtering techniques. The new method processes the image in both scale-space and spatial domains. Moreover, the new method does not classify the wavelet coefficients into noisy and clean coefficients but considers each to be partially noisy. A major step in the proposed method is to determine to what extent each coefficient has been affected by noise.

The block diagram of the proposed noise filtering technique is shown in Fig. 5.1. The suggested method is composed of two main modules: a module for detecting the edges and another module for filtering the noise. In the noise-filtering module, the image is transformed to the scale-space domain and a bank of adaptive filters is used to suppress the noisy coefficients. The information obtained about the location of edges (the output of the edge detection module) is used to adjust the parameters of the adaptive filters in the

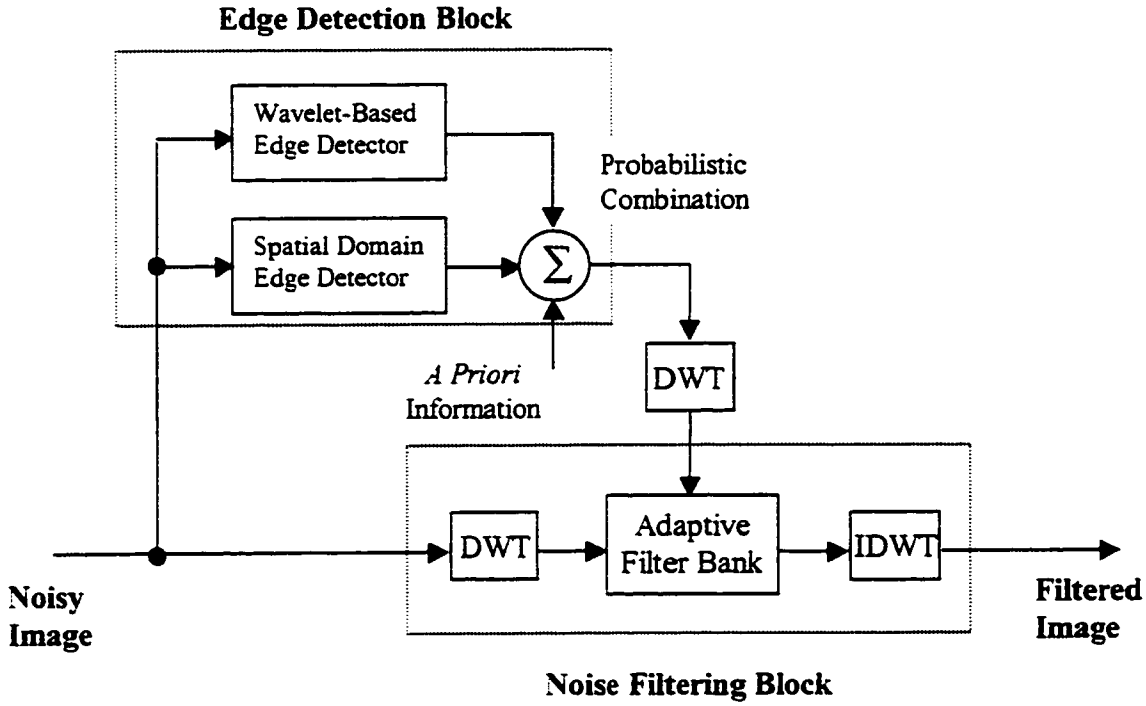


Figure 5.1 The Block diagram of the proposed noise filtering technique.

noise filtering module. As we will see in this chapter, involving the edge information in the noise filtering process will lead to a spatially selective low-pass filter which suppresses the high frequency components of the image except from the vicinity of edges. To implement the edge detection module, a new technique for detecting edges in the presence of noise is propsed.

In sections 5.2 and 5.3 we discuss spatial and wavelet domain techniques to process the image and locate its edges. In section 5.4, a probabilistic formulation is used to combine the spatial and scale-space domain results to make a final decision about the edges. In section 5.5, the information about the edges is used to remove the noise from the image without affecting the edges. In section 5.6 we will compare the experimental results

obtained using the new technique with the results obtained from other methods. Finally a short summary concludes this chapter.

5.2 Detecting the Edges in the Spatial Domain

5.2.1 Derivative operators

Edge detection is a basic step in many image processing techniques. In most applications, a great portion of the useful information in an image is often carried by its edges. To extract this information, several edge detection techniques have been proposed which process the image either directly in the spatial domain [Prewitt70, Fram75, Marr80, Levine85, Clark89] or after transforming it to a different domain [Canny86, Mallat92].

The idea underlying most spatial-domain edge detection techniques is the computation of a local derivative operator. Both the first and second derivatives of the image can be used for this purpose. However first derivative operators are less sensitive to noise [Gonzalez92]. The gradient of an image at location (x, y), $\nabla \mathbf{f}$, can be defined in terms of G_x and G_y which represent the partial derivatives of the image intensity function $f(x,y)$:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

The magnitude of the gradient vector, ∇f, represents the rate of change of image intensity at (x, y).

$$\nabla f = \text{modulus}(\nabla \mathbf{f}) = (G_x^2 + G_y^2)^{1/2}$$

Usually this quantity is approximated by a sum of magnitudes [Gonzalez92]:

$$\nabla f \approx |G_x| + |G_y|$$

The magnitude of the gradient, ∇f, has a local maximum at the position of edges. This property can be used to detect the edges. The local direction of an edge is perpendicular to the angle of the gradient vector, $\alpha(x,y)$, at (x, y):

$$\alpha(x,y) = Arctan(G_y / G_x)$$

and hence, the angle of the gradient vector can be used to detect the direction of edges.

To compute the gradient of a digital image, we need to compute the partial derivatives $\partial f/\partial x$ and $\partial f/\partial y$ at location (x,y). The digital form of these derivatives can be implemented in several ways. In our application, the derivative operators would be applied to noisy images and hence any noise enhancement effect associated with the derivatives can easily degrade the results. Among several implementations of these derivatives, the Sobel operators have the advantage of providing simultaneous differencing and smoothing effects. The smoothing effect of the Sobel operators reduces the effect of noise and hence makes them a good choice for our application. The 3×3 Sobel gradient masks are given in Fig. 5.2 [Levine85].

$$S_1 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad S_2 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad S_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_4 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

$$S_5 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad S_6 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \quad S_7 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad S_8 = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$$

***Figure 5.2** Sobel gradient masks used to detect edges in all possible directions associated with a 3×3 neighbourhood.*

To detect the edges, we convolve the noisy image with each of the eight directional masks. Each mask detects edges having a specific direction. The absolute values of the filtered images are compared at each location (x, y), and the strongest response, $e(x, y)$ is chosen as the value of the gradient image at (x, y).

$$e(x, y) = \max(f(x, y) * S_i) \qquad for \quad i = 1,...,8$$

$$x = 1, 2, ..., N_x \quad and \quad y = 1, 2, ..., N_y$$

where S_i is the i^{th} directional gradient mask given in Fig. 5.2.

Fig. 5.3(a) shows our 128×128 synthetic test image with the range of grey levels between 0 and 255. Fig. 5.3(b) represents the result after adding Gaussian white noise with variance of 400 to the test image. The result of applying the Sobel gradient masks is given in Fig. 5.3(c). Similarly, Fig. 5.4(c) shows the result of applying the Sobel masks to a natural test image.

5.2.2 Extraction of Edges

Although the smoothing effect of the Sobel masks reduces the effect of noise, the amount of noise in the gradient image is still noticeable (Figs. 5.3(c) and 5.4(c)). To obtain a better separation between signal and noise, the gradient image can be thresholded.

In general the gradient of major edges is larger than the gradient of noisy pixels. By setting all the pixels whose gradient is below a certain number (threshold) to zero, we can suppress the noise and highlight the major edges. In practice, however, it is almost impossible to find a proper threshold value. A small threshold value does not remove the

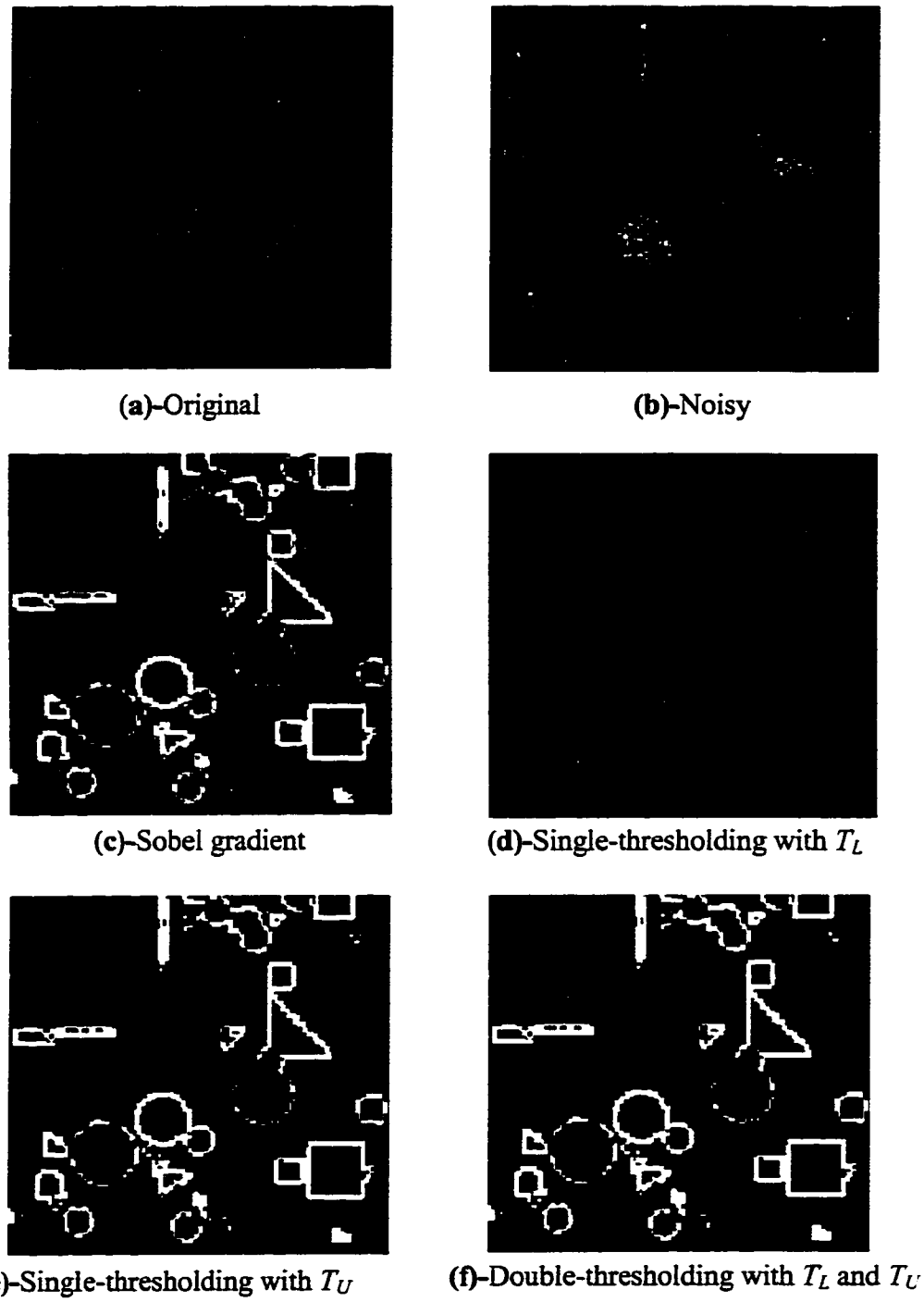


Figure 5.3 *Detecting the edges of a noisy synthetic image. (a) Original synthetic 128×128 image with the range of grey levels between 0 and 255. (b) After adding Gaussian white noise with variance 400 (grey level)^2 . Results of: (c) Applying the Sobel masks. (d) Single-thresholding with a small threshold value T_L.. (d) Single-thresholding with a large threshold value, T_U. (d) Double-thresholding with lower and higher thresholds equal to T_L and T_U respectively.*

(a)-Original

(b)-Noisy

(c)-Sobel gradient

(d)-Single-thresholding with T_L

(e)-Single-thresholding with T_U

(f)-Double-thresholding with T_L and T_U

Figure 5.4 *Detecting the edges of a noisy natural image. (a) Original 256×256 image with the range of grey levels between 0 and 255. (b) After adding Gaussian white noise with variance 400 (grey level)^2 . Results of: (c) Applying the Sobel masks. (d) Single-thresholding with a small threshold value T_L.. (d) Single-thresholding with a large threshold value, T_U.. (d) Double-thresholding with lower and higher thresholds equal to T_L and T_U respectively.*

noise effectively, while a high threshold eliminates some edge points and causes discontinuities in the edge contours. To solve this problem, we used a double thresholding method, which is a modification of Canny's method [Canny86].

In the first stage of our double thresholding method, pixels with gradient values greater than an upper threshold, T_U, are marked as "edge points". We use the edge points as starting points (seeds) for growing the edges. For this purpose, we mark all pixels in the neighbourhood of a seed with a grey level greater than a lower threshold, T_L, as "candidate edge points". The upper and lower threshold values are chosen based on the energy of noise in the image, which can be estimated from the signal-free regions of the image (see also discussion in Chapter 6). This edge growing process continues by checking the neighbourhood of each "candidate edge point" to find new candidates. It stops when no more pixels in the neighbourhood of the grown edge can be included in the region.

Using two different levels for thresholding greatly improves the connectivity of the edges. For an edge to be disconnected with this approach, the edge pixels must fluctuate above the upper threshold and below the lower threshold. In the simple thresholding methods, on the other hand, the connectivity of edges can be achieved only through reducing the threshold value, which increases the chance of choosing noisy pixels as false edges.

In the double thresholding method, the probability of detecting false edge points is also reduced. The noise strength at a noisy pixel must be quite high (above the higher threshold) before it can be detected as a seed point. However in the vicinity of the edge points it is still possible that some noisy points are marked as "candidate edge points". The results of applying single and double thresholding methods on the gradient image of Fig.

5.3(c) are shown in Figs. 5.3(d) through 5.3(f). For the single thresholding method, we either have false edges everywhere (using a low threshold) or edge discontinuities (using a high threshold). For the double thresholding approach, although false edges are removed from the smooth areas, some noisy pixels are wrongly marked as "candidate edge points" in the neighbourhood of edges. The results of applying the single and double thresholding methods to our natural test image in Fig. 5.4 show similar effects.

To reduce the effect of false candidate pixels on our final decisions, we assign a probability factor to each candidate pixel. This probability factor can take any value between zero and one and shows the chance of each candidate pixel of being on an edge. We also assign a probability factor of 1.0 to all pixels that are marked as "edge pixels" (pixels with gradients above T_U). To assign a probability factor to a candidate pixel, we use the fact that in most practical images, there are some similarities between the pixels that are located on an edge. Therefore in evaluating the candidate pixels, we can give a higher chance to the pixels that share some common properties with the previously accepted edge points (seeds). To quantise these similarities, the similarity between pixels must be defined using a mathematical formulation. For this purpose, we use three different similarity measures:

- Measure 1- Similarity in the Amplitude of the Gradient.
- Measure 2- Similarity in the Phase of the Gradient.
- Measure 3- Connectivity of the Direction Field.

The definitions of these measures are provided in the following sections.

Measure 1- Similarity in the Amplitude of the Gradient

In most practical images, the sharpness of an edge varies smoothly along its length. Hence the response of the gradient operator should vary in a similar way. As a result, the "candidate edge points" with the closest gradients to the gradient of an edge pixel must have the highest chances of being selected as edge points.

To measure the similarity between the gradient of pixels, a mathematical definition for the similarity of the gradients must be defined. Suppose the pixel at location (x', y') has a chance of $p_A(x', y')$ of being on an edge. We want to assign an "edge probability" factor to the pixel at location (x, y) based on the similarity of its gradient amplitude to the gradient amplitude of the pixel at (x', y'). This probability factor can be defined as:

$$p_A(x, y) = p_A(x', y').S_A(x, y, x', y') \qquad (5.1)$$

where $S_A(x, y, x', y')$ is our mathematical definition for the similarity between the magnitude of gradients at (x, y) and (x', y') which can be expressed as:

$$S_A(x, y, x', y') = \begin{cases} 1 & |G(x, y)| \le G_1(x', y') \\ 1 - \dfrac{|G(x, y)| - G_1(x', y')}{G_0(x', y') - G_1(x', y')} & G_1(x', y') < |G(x, y)| < G_0(x', y') \\ 0 & G_0(x', y') \le |G(x, y)| \end{cases}$$

$$G(x, y) = (\nabla f(x, y) - \nabla f(x', y')) / \nabla f(x, y),$$

where (x', y') is the location of the reference pixel and $G_1(x', y')$ represents the maximum deviation in the gradient amplitude from that of the reference pixel that a pixel can have and still be considered *quite similar* to the reference pixel. Similarly, $G_0(x', y')$ shows the minimum deviation that causes a pixel to be considered *quite different* from the

reference pixel. Both these quantities are functions of the gradient of the reference pixel at location (x', y').

Fig. 5.5 shows the fuzzy classifier used to assign an amplitude similarity measure to each candidate pixel based on the similarity of its gradient amplitude to that of the reference pixel.

To assign a probability factor, $p_A(x, y)$, to each candidate pixel, we start from the candidate pixels that are inside the 8-connected neighbourhood of a seed pixel (first-layer candidate pixels). For the first-layer candidate pixels, the seed pixels are used as reference points. The seed pixels are the points that are already marked as edge points and hence we assume $p_A(seed) = 1.0$. As a result, Eqn. 5.1 takes the following simple form for the first-layer candidate pixels:

$$p_A(x, y) = S_A(x, y, x', y')$$

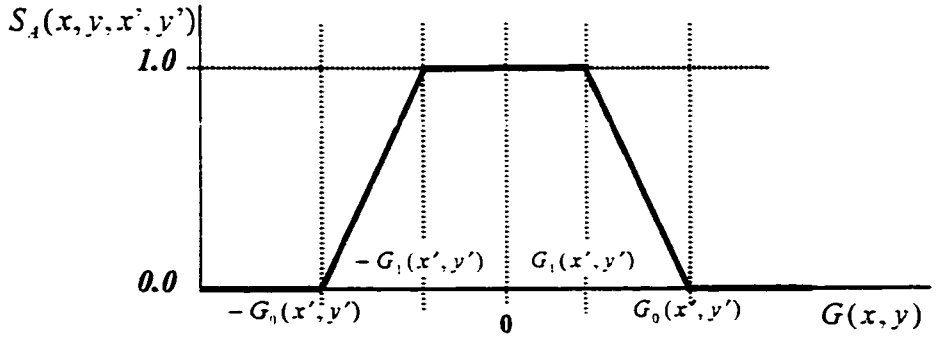


Figure 5.5 *The fuzzy classifier which is used to assign a similarity measure to each candidate pixel based on the similarity of its gradient amplitude to the gradient amplitude of the reference pixel at (x', y').*

It is possible that a candidate pixel is in the 8-connected neighbourhood of more than one reference pixel. As a result several probability factors can be assigned to a candidate pixel. To deal with this situation we will change the probability factor of a pixel, only if the new factor assigned is greater than the current value assigned to that pixel.

After assigning a probability factor to all first-layer candidate pixels (candidate pixels in the 8-connected neighbourhood of seed pixels), we continue this process for the second layer candidate pixels (candidate pixels in the 8-connected neighbourhood of first layer candidate pixels). For this purpose we use Eqn. 5.1 to assign a probability factor to the second-layer candidate pixels. For the second layer candidate pixels the reference pixel at location (x', y') is no longer a seed pixel. In fact $p_A(x', y')$ in Eqn. 5.1 is the probability factor assigned to a first layer candidate pixel and can be different from unity. We continue this process until we assign a probability factor to all candidate pixels.

Measure 2- Similarity in the Phase of the Gradient

In most practical images, the direction of an edge varies smoothly along its length. As a result, we expect to encounter only small variations in the direction of the gradient vector when we track an edge.

Similar to what we did for the amplitude of the gradient, a probability can be assigned to each candidate pixel based on the direction of its gradient. Again pixels with higher probabilities are more likely to be selected as edge points. The probability assigned to each candidate pixel is defined as:

$$p_B(x, y) = p_B(x', y').S_B(x, y, x', y')$$

where (x', y') is the location of the reference pixel and $S_B(x, y, x', y')$ is our mathematical definition of the similarity between the phase of the gradient at (x, y) and (x', y'). This similarity can be expressed as:

$$p_B(x, y, x', y') = \begin{cases} 1 & |\Delta\alpha(x,y)| \le \alpha_1 \\ 1 - \dfrac{(|\Delta\alpha(x,y)| - \alpha_1)}{(\alpha_0 - \alpha_1)} & \alpha_1 < |\Delta\alpha(x,y)| \le \alpha_0 \\ 0 & \alpha_0 < |\Delta\alpha(x,y)| \end{cases}$$

$$\textit{where} \quad \Delta\alpha(x,y) = \alpha(x,y) - \alpha(x',y')$$

where α_1 represents the maximum deviation in the gradient direction that a pixel can have and still be considered *quite similar* to the reference pixel. Similarly, α_0 is the minimum deviation that causes a pixel to be considered *quite different* from the reference pixel. Fig. 5.6 shows the fuzzy classifier used to assign an angle similarity measure to each candidate pixel based on the similarity of its gradient phase to that of the reference pixel.

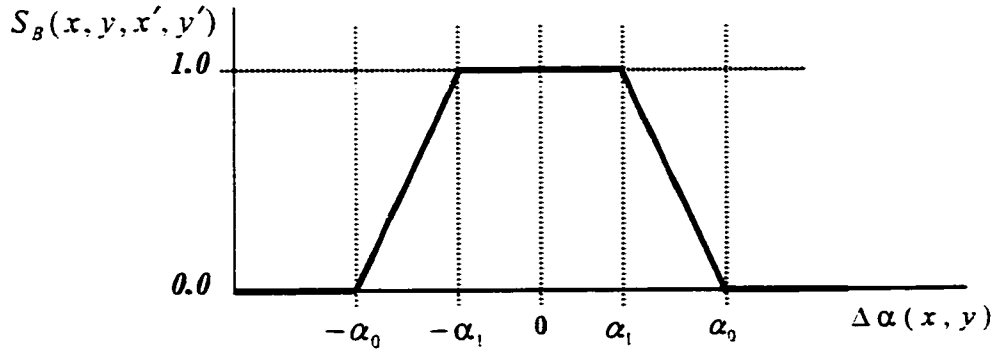


Figure 5.6 *The fuzzy classifier which is used to assign a similarity measure to each candidate pixel based on the similarity of its gradient phase to the gradient phase of the reference pixel (x', y').*

Measure 3- Connectivity of the Direction Field

By assigning a direction to each pixel of the image, a direction field can be defined. First we need to define a procedure for assigning directions to pixels. Fig. 5.7 shows a set of twelve masks that are used for this purpose.

The non-zero elements of each mask are located on a line that passes through the center of the mask. At every location (x, y) of the gradient image, the result of applying each of these masks is computed. By comparing these results, the direction of the

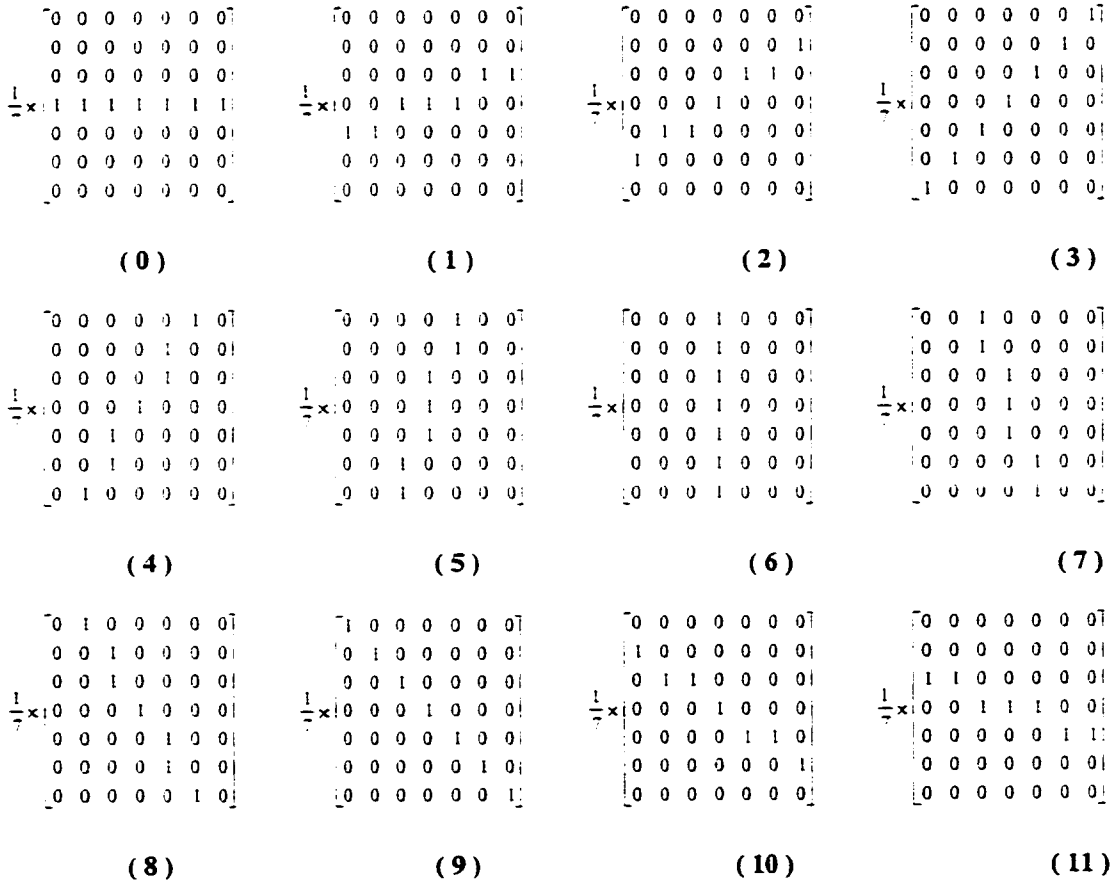


***Figure 5.7** The masks defined for assigning a direction to each pixel.*

prospective edge that passes through (x, y) can be detected. The direction of the non-zero line in the mask with the highest response is the best estimate for the edge direction at location (x, y). We define for each pixel a direction that is equal to the number of the mask with the highest response. Fig. 5.8 shows some direction fields associated with our test images that were obtained using this definition. In these direction fields, there is a high correlation between the pixels that are located along the edges. In other parts of the field, the image intensities are randomly distributed.

A probability can be assigned to each candidate pixel based on its direction as we did for the amplitude and phase of the gradient. Again, pixels with higher probabilities are more likely to be selected as edge points. The probability assigned to each candidate pixel is defined as:

$$p_C(x, y) = p_C(x', y').S_C(x, y, x', y')$$

where the reference pixel is located at (x', y') and $S_C(x, y, x', y')$ is our mathematical definition for the similarity between the directions of pixels located at (x, y) and (x', y') which can be expressed as:

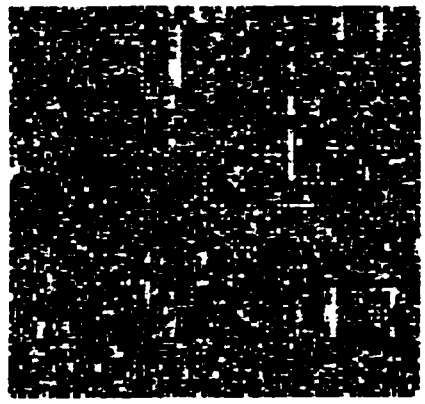
$$S_C(x, y, x', y') = \begin{cases} 1 & \text{if} \quad D(x, y) \le D_1 \\ 1 - \dfrac{D(x, y) - D_1(x, y)}{D_0(x, y) - D_1(x, y)} & \text{if} \quad D_1 < D(x, y) \le D_0 \\ 0 & \text{if} \quad D_0 < D(x, y) \end{cases}$$

and:

$$D(x, y) = \begin{cases} |d(x, y) - d(x', y')| & \text{if} \quad |d(x, y) - d(x', y')| \le 6 \\ 12 - |d(x, y) - d(x', y')| & \text{if} \quad |d(x, y) - d(x', y')| > 6 \end{cases}$$

(a)-Noisy synthetic image

(b)-Noisy natural image



(c)- The phase of gradient for the synthetic image

(d)- The phase of gradient for the natural image

(e)- The direction field for the synthetic image

(f)- The direction field for the natural image

Figure 5.8 *Smooth change in the phase of the gradient and direction of pixels (as defined in Section 5.2.2) along edges: (a) Synthetic 128×128 image and (b) 256× 256 natural image after adding Gaussian white noise with* $\sigma_n^2 = 400$ *(grey level)*^{2}.
(c) and (d) the phase of the gradient for the test images.
(e) and (f) the direction fields for the test images.

where the maximum deviation in the direction that a pixel can have and still be considered *quite similar* to the reference pixel is defined by D_1. Similarly, D_0 is the minimum deviation that causes a pixel to be considered *quite different* from the reference pixel. Fig. 5.9 shows the fuzzy classifier used to assign a direction similarity measure to each candidate pixel based on the similarity of its direction to that of the reference pixel.

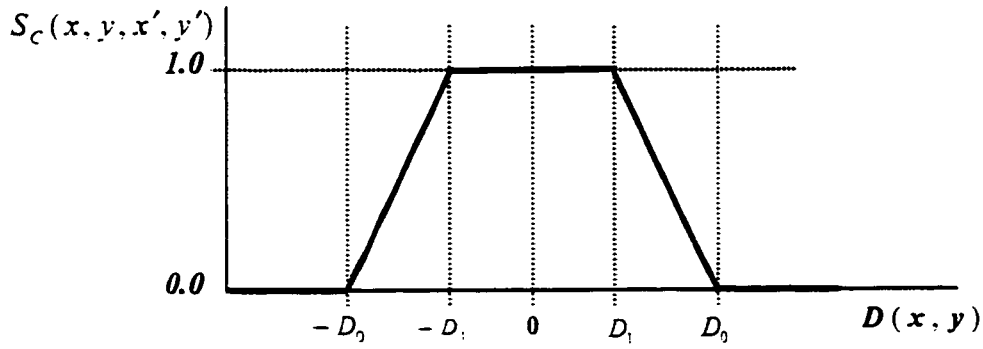


Figure 5.9 *The fuzzy classifier which is used to assign a similarity measure to each candidate pixel based on the similarity of its direction (as defined in the text) to the direction of the reference pixel (x', y').*

Combining the Amplitude, Phase and Direction Similarity Measures

In the previous sections, we assigned three different probability factors to each candidate edge pixel. Each of these numbers shows the chance of a candidate pixel of being on an edge based on a different criterion. In this section we use a probabilistic formulation to combine these factors into a single number. To determine the overall probability factor, both the union and intersection of the partial probability factors must be taken into account:

$$P(x,y) = k_1 \bigcup \{p_A(x,y), p_B(x,y), p_C(x,y)\} + k_2 \bigcap \{p_A(x,y), p_B(x,y), p_C(x,y)\}$$

where $\bigcup$ and $\bigcap$ represent union and intersection operators respectively and *k1* and *k2* must satisfy:

$$k_1 + k_2 = 1$$

By increasing the ratio $k1/k2$, the impact of individual similarity measures on the final decision will be increased. Decreasing the ratio $k1/k2$ puts more emphasis on the coincidence between the measures. Using the fuzzy logic definitions of union and intersection operators [Zimmer80, VonAltrock95], we have:

$$P(x,y) = k_1[\max(p_A(x,y), p_B(x,y), p_C(x,y)] + k_2[\min(p_A(x,y), p_B(x,y), p_C(x,y)]$$

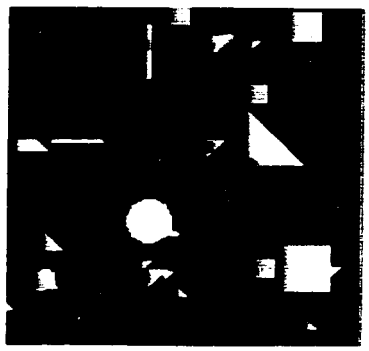
Candidate pixels with the higher $P(x,y)$ are more likely to belong to an edge.

Instead of setting a threshold and treating all the pixels with $P(x,y)$ values above the threshold as edge points, we decided to use a more adaptive approach. We enhance all the pixels in the output of the edge detector. However, the level of the enhancement at each location (x,y), depends on the value of the final probability, $P(x,y)$, at that point. The edge-enhanced image can be represented by:

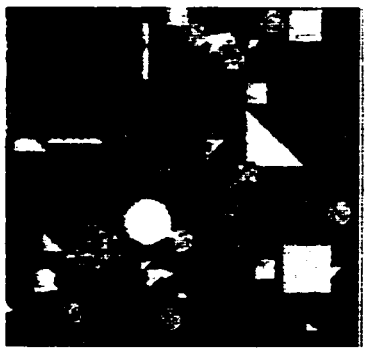
$$S'(x,y) = (1 + k.P(x,y)).S(x,y) \qquad (5.2)$$

where $S(x,y)$ is the gradient image and k is a multiplicative factor that is used to control the level of edge enhancement, or equivalently the level of noise suppression.

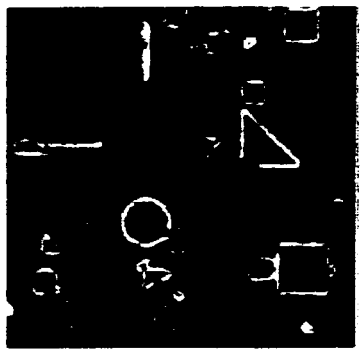
Figs. 5.10 and 5.11 show the edge enhanced images obtained by applying the above method to the synthetic and natural test images, respectively. In the vicinity of some strongly corrupted pixels, short false edges are detected (for example, the false edges detected on the Lenna's face, in Fig. 5.11(e). These false edges can be removed by setting a threshold for the minimum acceptable length of an edge. In the edge-growing



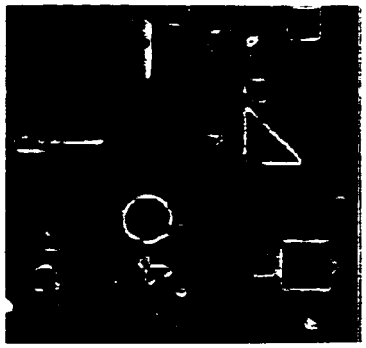
(a)-Original



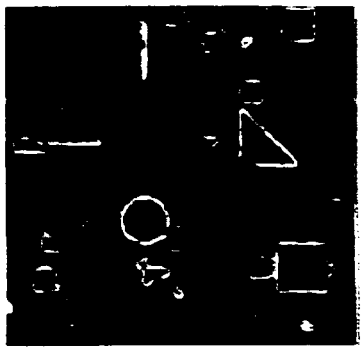
(b)-Noisy



(c)-Sobel gradient



(d)-Edge enhanced gradient image
(k = 1.5 in Eqn. 5.2)



(e)-Edge enhanced gradient image
(k = 3.0 in Eqn. 5.2)

Figure 5.10 *Enhancing the edges in the gradient image of the synthetic test image by using the similarity measures (Eqn. 5.2) (a) Original synthetic 128× 128 image with the range of grey levels between 0 and 255 (b) After adding Gaussian white noise with variance 400 (grey level)^2 . (c) The Sobel gradient image (c) Using Eqn. 5.2 to enhance the edges. k = 1.5 (d) k =3.0. The grey levels are re-scaled to the range 0 to 255.*

(a)-Original

(b)-Noisy

(c)-Sobel gradient

(d)-Edge enhanced gradient image
(k = 1.5 in Eqn. 5.2)

(e)-Edge enhanced gradient image
(k = 3.0 in Eqn. 5.2)

Figure 5.11 *Enhancing the edges in the gradient image by using the similarity measures (using Eqn. 5.4.) (a) Original 256×256 Lenna image with the range of grey levels between 0 and 255 (b) after adding Gaussian white noise with variance 400 (grey level). Results of: (c) Sobel gradient. (c) Using Eqn. 5.2 to enhance the edges, k = 1.5. (d) Using Eqn. 5.2 to enhance the edges, k = 3.0. The grey levels are re-scaled to the range 0 to 255.*

algorithm, all edges with a length shorter than the threshold can be extracted from the image. In practice, however, this can filter out some small features from the image.

In the next section we will introduce another edge detection algorithm which processes the image in the scale-space domain. Although the proposed wavelet-based method is not as accurate as the spatial domain edge detector, it is less sensitive to noise. Most of the false edges, which are wrongly marked by the spatial domain method, are not present in the edge enhanced image obtained using the wavelet-based method. As we will see later, by superimposing the outputs of these two methods, we can suppress the false edges in the final result.

5.3 Detecting Edges in the Wavelet Domain

5.3.1 Introduction

The spatial domain edge detector, which was proposed in the previous section, can detect even weak edges accurately. However, in the presence of strong, spike-like noise, the response of the gradient operator at noisy pixels could be high. This may lead to detection of wrong seed pixels and hence false edges. We shall now show that this is less of a problem in the wavelet domain although other concerns appear.

It has been shown that the wavelet coefficients of noise have a much weaker correlation between scales than the wavelet coefficients of edges [Mallat92, Xu94]. A major edge can generate large wavelet coefficients at several scales, while a noisy pixel can only generate large coefficients at the finest scales. Xu *et al.* used the direct multiplication of coefficients across the scales to highlight edges and suppress noise [Xu94]. Both fine and coarse scales have a crucial role in this direct multiplication

process. The noiseless coarse scales are necessary to suppress noise, while fine scales are essential to localise edges. In practice, however, Xu *et al.* noticed that involving more than two scales in direct multiplication had a negative impact on the result [Xu94].

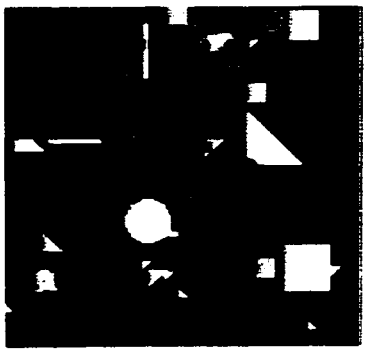
5.3.2 Involving more scales in the direct correlation process

The new wavelet-based edge detection algorithm discussed in this section is a modified version of Xu's approach. The changes we have made to Xu's method allow us to involve more scales in the edge detection process without facing the side effects Xu *et al.* mentioned. As we will see in this section, involving more scales in the correlation process will improve the performance of the method in recognising the edges from the noise.

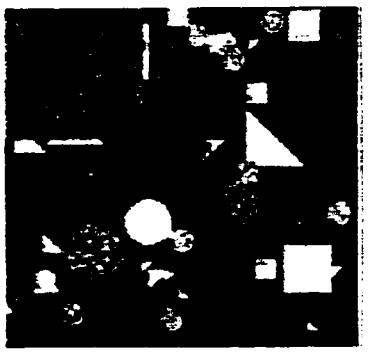
Before we explain our method, it is important to note that the wavelet transform of an image consists of three separate groups of coefficients at each scale. Each group of coefficients, which is called a detail image, is sensitive to the features in a specific direction (Horizontal, Vertical or Diagonal). To compute the correlation of coefficients across the scales, we have to multiply the coefficients in each group of detail images separately.

To compute the wavelet coefficients we used the Biorthogonal quadratic-spline wavelets first introduced by Mallat and Zhong [Mallat89b] to decompose the image. These wavelets have a shape very close to the derivative of Gaussian and hence can act as an optimal edge detector [Canny86].

To locate the edges, we compute the direct correlation of wavelet coefficients across the scales. Figs. 5.12(a) and 5.12(b) show our original and noisy synthetic test images.



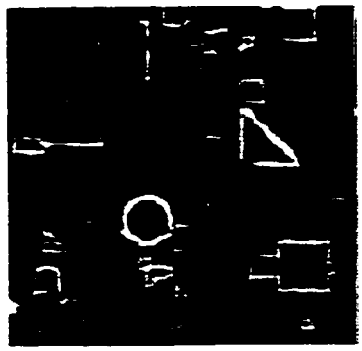
(a)- Original



(b)- Noisy

(c)- Correlation between 2 Scales

(d)- Correlation between 4 Scales



(e) After compressing the dynamic range of Fig. (d).

Figure 5.12 *The direct spatial correlation across the scales for the synthetic noisy image. (a) Synthetic 128× 128 image with grey levels between 0 and 255 (b) After adding Gaussian noise with variance 400 (grey level)*^{2}*. Results of: (c) The direct spatial correlation between two scales. (d) The direct spatial correlation between four scales. (e) The result obtained by compressing the dynamic range of coefficients in Fig. 'd'.*

The results of computing the direct correlation of the wavelet coefficients for this test image are shown in Figs. 5.12(c) and 5.12(d). The number of scales involved in the direct correlation process is 2 for Fig. 5.12(c) while in Fig. 5.12(d) this number is deliberately chosen as too large, 4. To generate the correlation images, the correlation across the scales are first calculated for each group of detail images separately. Then, for a better visualisation of the edges, thethree correlation images in each level are added together. It can be seen from the correlation images in Fig. 5.12 that involving more scales in the edge detection process did not improve the result. Many edges that are successfully detected in Fig. 5.12(c) are missed in Fig. 5.12(d). This is exactly consistent with the results obtained by Xu *et al.* [Xu94].

To fix this problem, we must take into account the fact that involving a larger number of scales in the direct multiplication increases the dynamic range of the resulting correlation image. Strong edges generate large wavelet coefficients at all levels, while the wavelet coefficients are much smaller for weak edges. As a result, multiplying the coefficients across the scales enhances the large coefficients while it weakens the smaller coefficients in the correlation image. We compress the dynamic range of the correlation image. For this purpose, we perform the following transformation:

$$s(x,y) = k \,.\, log(\, 1 + |\, c(x,y)|\,)$$

$$x = 1, 2, \ldots, N_x \quad and \quad y = 1, 2, \ldots, N_y$$

where $c(x,y)$ is the pixel intensity at location (x,y) of the correlation image.

Fig. 5.12(e) shows the result obtained after compressing the dynamic range of the correlation image shown in Fig. 5.12(d). Both strong and weak edges are well localised and highlighted in this image. Moreover the noise is effectively suppressed in this image.

In Fig. 5.13 the effects of compressing the dynamic range of coefficients for our natural test image are presented.

5.3.3 Introducing *a priori* geometrical knowledge

Xu *et al.* used the correlation image directly to classify the pixels as either on the edge or noisy. We can improve the quality of the result by using *a priori* geometrical knowledge about the image. In most practical images, the edge pixels tend to be clustered. Moreover, the strength and direction of edges vary smoothly along their length. Similar to what we did in our spatial domain edge detection algorithm, we apply a double thresholding method to the wavelet correlation coefficients.

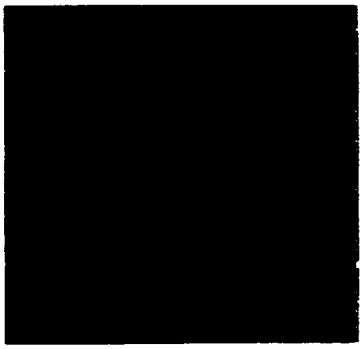
The upper and lower threshold values are chosen as functions of the noise parameters, which are estimated from the signal-free regions of the correlation image. Any pixel of the correlation image with a value above the upper threshold is used as a starting point (seed) for growing the edges. The edge will extend to all 8-connected neighbours of the seed that have a correlation above the lower threshold (candidate edge pixels). Next we measure the similarity between the candidate pixels and their neighbouring seed pixels to assign aprobability factor to each candidate pixel. Finally we use the probability assigned to each pixel to enhance its intensity. As a result, pixels with higher probabilities of being on an edge get greater enhancement. The edge-enhanced image can be represented by:

$$S'(x,y) = (1 + k.P(x,y)).S(x,y)$$

where $P(x,y)$ is the probability assigned to each edge pixel. $S(x,y)$ is the correlation image obtained using the direct multiplication of scales and k is a multiplicative factor used to control the level of edge enhancement.

(a)-Original

(b)- Noisy



(c)- Correlation between 4 Scales

(d)- After compressing the dynamic range of Fig. (c).

Figure 5.13 *Computing the direct spatial correlation across the scales for the natural noisy image. (a) 256×256 test image with grey levels between 0 and 255 (b) After adding Gaussian white noise with variance 400 (grey level)^2 . Results of: (c) The direct spatial correlation between four scales. (d) Compressing the dynamic range of coefficients in Fig. 'c'.*

5.4 Combining the Results

In the previous two sections, we used the spatial and scale-space domain techniques independently to detect the position of edges in a noisy image. The final result in each domain was an image in which the edges are highlighted. Each of these two images has some information about the location and strength of edges in the original noisy image. In this section we mix the information to derive a single edge-enhanced image that shows the position and strength of edges more accurately.

To combine the spatial and wavelet domain edge-enhanced images, we need to re-scale them to the same range of intensities (zero to one). In the following context, the re-scaled spatial and scale-space domain images are represented by $E_S(x,y)$ and $E_W(x,y)$ respectively. The intensity of each pixel in these images shows the chance of that pixel of being selected as an edge point in the final image. Therefore, two different probabilities have been assigned to each pixel that should be mixed into a single probability which shows the overall chance of that pixel of being on an edge.

To derive the final image, the local intensities of each image, as well as their correlation must be taken into account. The final image has the following general form:

$$E_{S-W}(x,y) = k_1 \bigcap \{E_S(x,y), E_W(x,y)\} + k_2 \bigcup \{E_S(x,y), E_W(x,y)\} \qquad (5.2)$$

$$k_1 + k_2 = 1$$

The first component is a function of the correlation between two images, while the second component can be affected by each image separately. By choosing a large k_1 / k_2 ratio, we can put more emphasis on the agreement between our spatial and wavelet domain edge

detectors in locating the edges. Using the fuzzy logic definitions for intersection and union operators [Zimmer80, VonAltrock95], we can re-write Eqn. 5.2 as:

$$E_{S-W}(x,y) = k_1.[\min\{E_S(x,y), E_W(x,y)\}] + k_2.[\max\{E_S(x,y), E_W(x,y)\}]$$

The probability image $E_{S-W}(x,y)$ is our final basis for detection of edges. For simplicity in the rest of this paper we will call the above image the "Edge Probability" image.

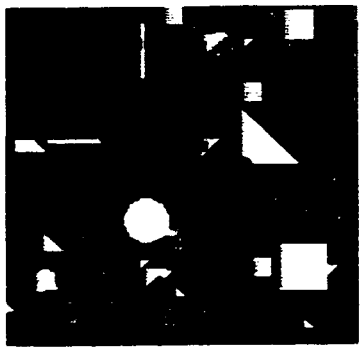
The "Edge Probability" images associated with our synthetic and natural test images are shown in Fig. 5.14 and 5.15 respectively ($k_1 = 0.9$, and $k_2 = 0.1$). As we can see in these figures, even in the presence of strong noise, the new method is quite successful in detecting the edges.

5.5 Filtering the Noise

In this section we use our probabilistic knowledge about edges to preserve them during noise filtering. As with all wavelet-based noise filtering techniques, our proposed method consists of three steps: *Decomposing* the image into its wavelet coefficients, *Modifying* the wavelet coefficients, and *Reconstructing* the cleaned image from the modified coefficients. To decompose the image into its multi-scale wavelet representation, we need to choose a proper wavelet. As we will see later, the proposed method is insensitive to the selected wavelet's shape. However, the wavelet function is preferably be smooth to

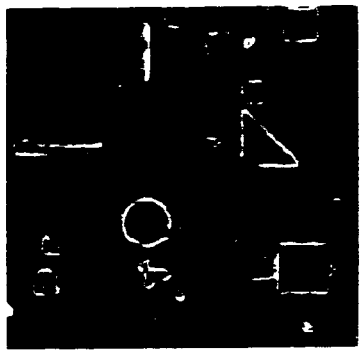
minimise the visual distortion from incorrect noise suppression.

After decomposing the image into its wavelet coefficients, we need to preserve the fine scale (high frequency) wavelet coefficients associated with edges while removing the

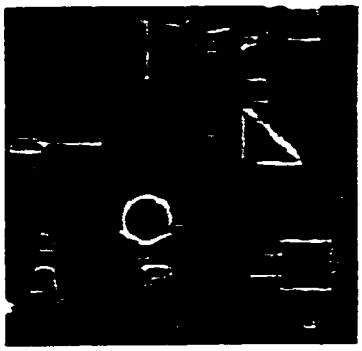


(a)-Original

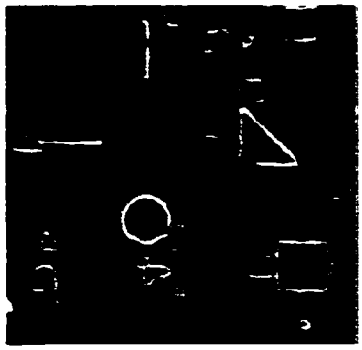
(b)-Noisy



(c)-Using the spatial domain edge detection method



(c)-Using the scale-space domain Edge detection method



(e)- Combining the information in Figs. (c) and (d)

Figure 5.14 *Detecting the edges of the noisy synthetic image (a) Synthetic 128×128 test image with grey levels between 0 and 255 (b) After adding Gaussian noise with variance 400 (grey level)*^{2}*. Results of: (c) The spatial domain edge detector. (d) Wavelet based edge detector. (e) Using a probabilistic formulation to combine the previous two images.*

(a)-Original

(b)-Noisy

(c)-Using the Spatial domain Edge detection method

(d)-Using the Scale-space domain Edge detection method

(e)- Combining the information in Figs. (c) and (d)

Figure 5.15 *Detecting the edges of a noisy natural image (a) Natural 256x256 test image with grey levels between 0 to 255 (b) After adding Gaussian noise with variance 400 (grey level)*^{2} *. Results of: (c) The spatial domain edge detector. (d) Wavelet-based edge detector. (e) Using a probabilistic formulation to combine the previous two images.*

high frequency components from other areas. For this purpose, we define a neighbourhood around each edge and keep the wavelet coefficients in this neighbourhood while suppressing the coefficients from the rest of the image. The shape and size of these neighbourhoods must be defined separately for each detail image. In general the energy of edges is spread over a broader area for detail images from coarser scales. To preserve the energy of edges at coarser scales, we need a larger neighbourhood around each edge.

The proper neighbourhoods for preserving the energy of edges are selected automatically in the proposed method. For this purpose we decompose the "Edge Probability" image (the result obtained in the edge detection stage of the algorithm) into a hierarchy of detail images (see Fig. 5.16). By using the same wavelet to decompose both the original and "Edge Probability" images, the energy of edges of both these images will spread in the same way among their detail images. As a result, the wavelet coefficients associated with the edges of the "Edge Probability" image can be used to detect the wavelet coefficients associated with edges of the original noisy image. Each "Edge Probability" detail image is used as a mask to filter the corresponding component (detail image) of the noisy image.

After removing the noisy coefficients from the detail images, we reconstruct the filtered image from the clean coefficients.

5.6 Results

We applied our method to both synthesised and natural noisy images. The sharpness of edges and the amount of noise remaining in the output were among the qualitative

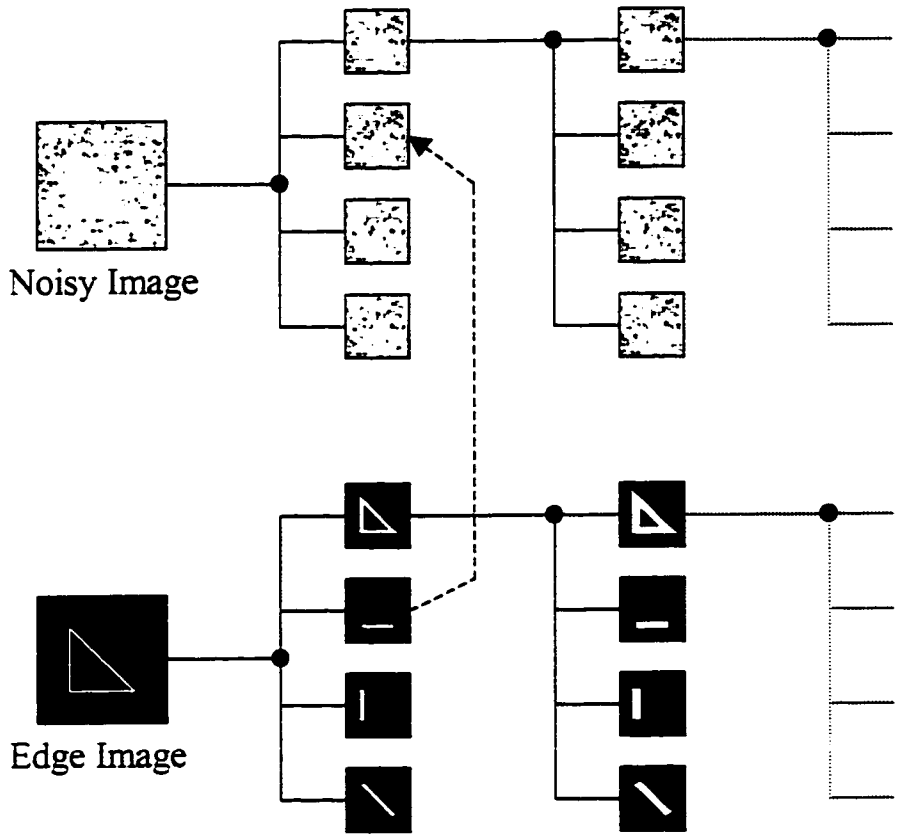
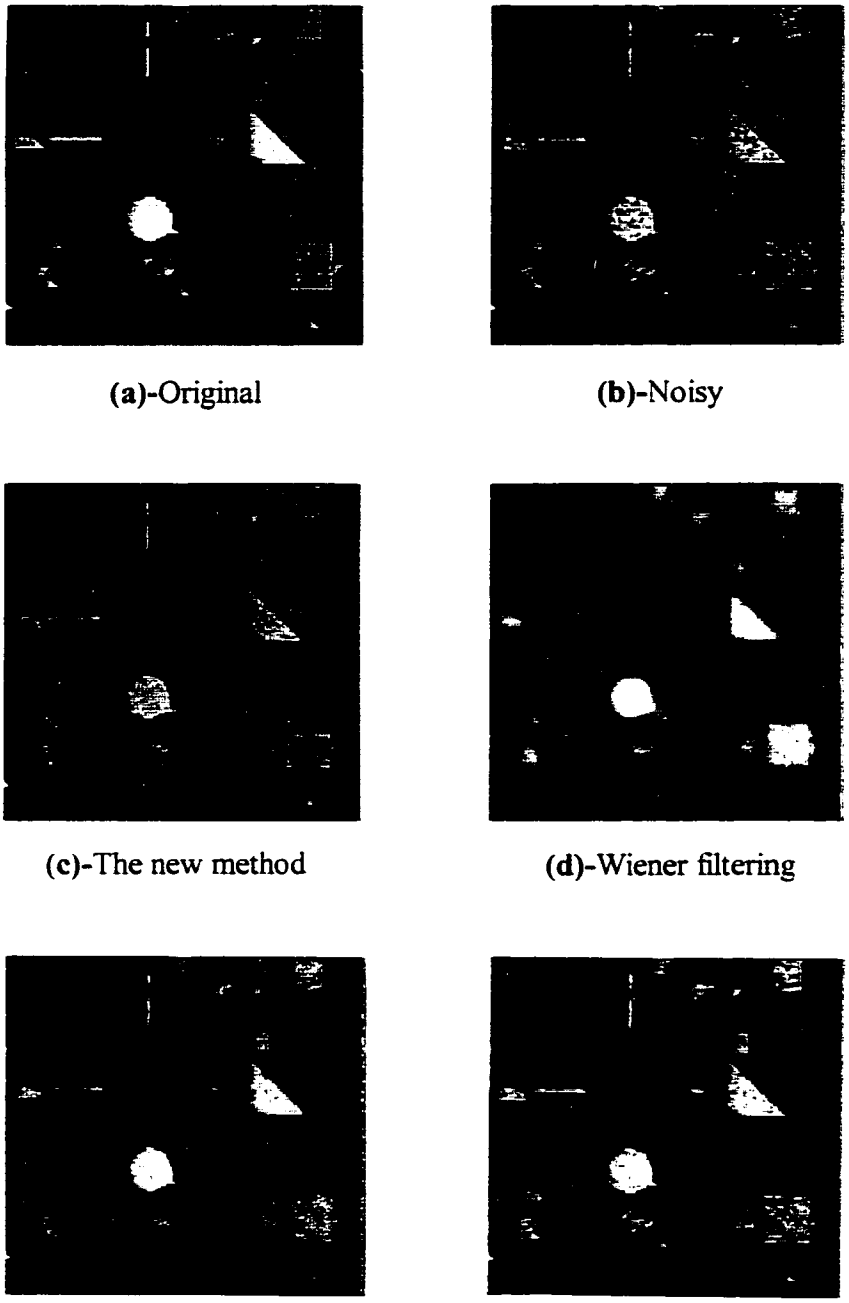


Figure 5.16 *Using the Edge information across the scales to preserve the edges during noise filtering.*

criteria used to compare the images. As global measures for quantitative improvement of the results were MSE and ISNR (section 2.1).

As an initial presentation of the performance of our algorithm we applied it to the synthesised image. In Fig. 5.17(a) the original 128×128 synthetic image is presented. In Fig. 5.17(b) white Gaussian noise with variance 400 has been added to the image. The result of filtering the noisy image with the new method is shown in Fig. 5.17(c). Image features are well retained while the noise is removed quite effectively from this image.



(a)-Original **(b)**-Noisy

(c)-The new method **(d)**-Wiener filtering

(e)- Soft thresholding **(f)**- Xu's method

Figure 5.17 *Comparing the performance of some noise filtering methods in removing the noise from a synthetic test image. (a) Synthetic 256x256 image with grey levels between 0 and 255 (b) After adding Gaussian noise with variance 400 (grey level)^2 . The result obtained using: (c) The propsed wavelet-based method. (d) Wiener filtering (e) Soft Thresholding technique. (f)Xu's method.*

The sharpness of most edges is also preserved. However some noisy spots are left around the sharp edges. For comparison, the results of applying the Wiener, soft thresholding and Xu's methods to the same noisy image are shown in Figs. 5.17(d), (e) and (f) respectively. Table 5.1 gives the quantitative measures for the filtered images. A comparison of quantitative and qualitative results for these filtered images demonstrates the improved performance of the new method.

Figure	*Description*	*MSE (Grey level)²*	*ISNR (dB)*
5.17(b)	Additive noise , $\sigma^2 = 400$ (Grey level)^2	393.0 ± 1.7	
5.17(c)	The proposed method	71.5 ± 0.3	7.40 ± 0.02
2.1(f)	Adaptive neighbourhood method	74.7 ± 0.4	7.21 ± 0.02
2.1(e)	Lee's refined method	77.3 ± 0.7	7.06 ± 0.04
5.17(d)	Wiener filter	94.5 ± 0.7	6.19 ± 0.02
2.1(c)	Lee's method	105.2 ± 0.6	5.73 ± 0.03
2.1(d)	Kuan's method	107.0 ± 0.5	5.66 ± 0.03
4.1(d)	Hard thresholding method	112.4 ± 0.9	5.43 ± 0.04
5.17(f)	Xu's method	120.6 ± 0.7	5.13 ± 0.03
5.17(e)	Soft thresholding method	128.6 ± 0.57	4.85 ± 0.02

Table 5.1 *Quantitative measures (MSE and ISNR) for the results obtained by applying different noise filtering techniques to the noisy synthetic image shown in Fig. 5.17(b).*

Fig. 5.18 presents results of application of the filters to the natural image. The result of filtering the noisy image in Fig. 5.18(b) with the new method as well as the Wiener, soft thresholding and Xu's methods are shown in Fig. 5.18(c) through 5.18(f). Table 5.2 gives the quantitative measures for the filtered images. Again, from a visual inspection of the filtered images, the superior performance of the new method in removing the noise without blurring the edges is evident.

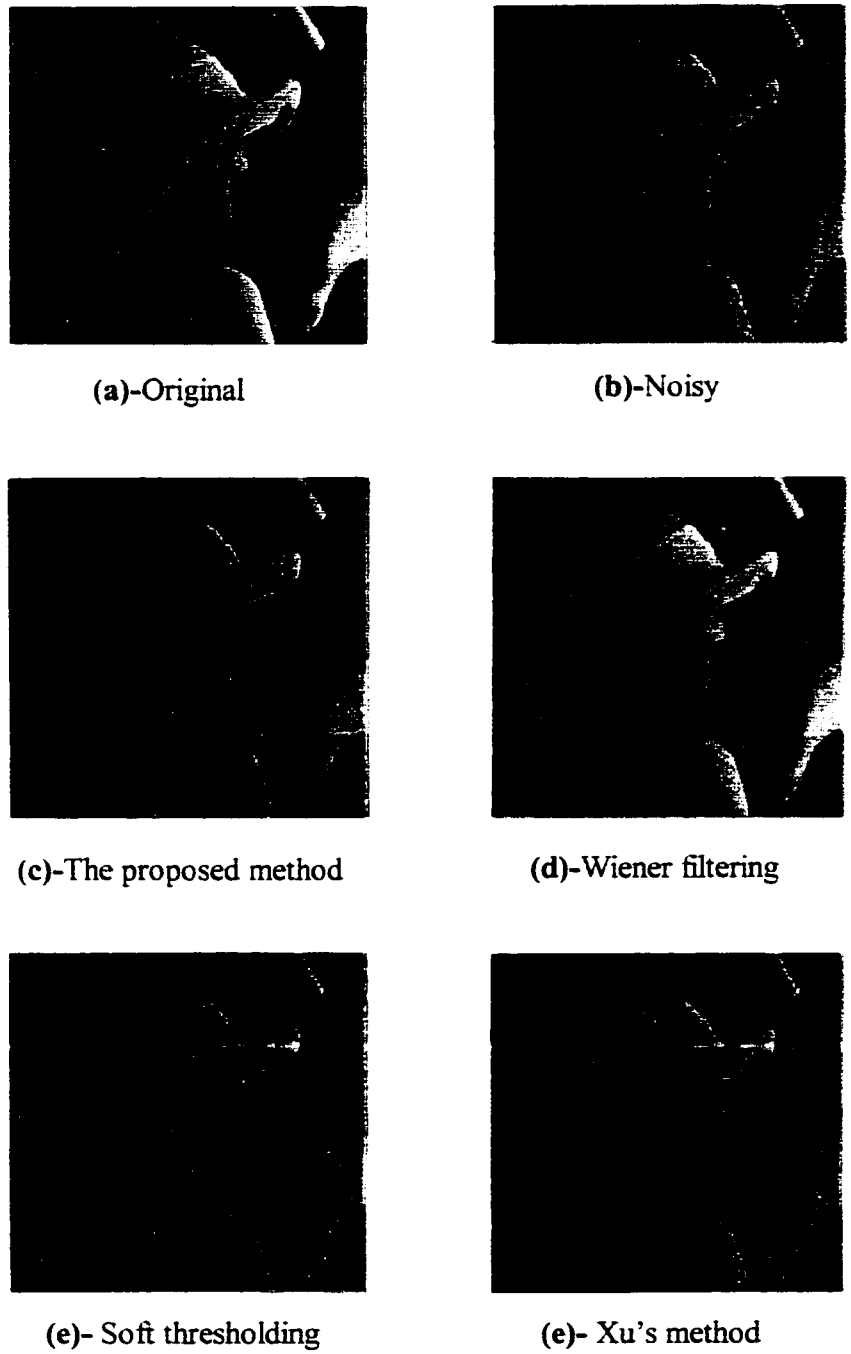


Figure 5.18 *Comparing the performance of some noise filtering methods in removing the noise from a natural test image. (a) 256x256 test image with grey levels between 0 and 255. (b) After adding Gaussian noise with variance 400 (grey level)^2 . The result obtained using: (c) The propsed wavelet-based method (d) Wiener filtering. (e) Soft thresholding technique. (f) Xu's method.*

Figure	*Description*	*MSE* *(Grey level)²*	*ISNR* *(dB)*
5.18(b)	Additive noise , σ^2 = 400 (Grey level)²	384.0 ± 1.7	0.00 ± 0.00
5.18(c)	The proposed method	77.6 ± 0.4	6.94 ± 0.02
2.2(f)	Adaptive neighbourhood method	79.4 ± 0.3	6.85 ± 0.03
5.18(f)	Xu's method	97.2 ± 0.5	5.97 ± 0.02
5.18(d)	Wiener filter	98.3 ± 0.6	5.92 ± 0.03
2.2(e)	Lee's refined method	99.6 ± 0.4	5.86 ± 0.02
4.2(c)	Hard thresholding method	101.6 ± 0.8	5.77 ± 0.04
2.2(d)	Kuan's method	105.3 ± 0.4	5.62 ± 0.02
5.18(e)	Soft thresholding method	105.5 ± 0.8	5.61 ± 0.03
2.2(c)	Lee's method	112.2 ± 0.5	5.34 ± 0.02

***Table 5.2** Quantitative measures (MSE and ISNR) for the results obtained by applying different noise filtering techniques to the noisy natural image shown in Fig. 5.18(b).*

5.7 Discussion and Conclusions

As we saw in the previous section, the Wiener filter was not very successful in filtering the noise from our low-SNR test images. Wiener filtering, as a frequency domain technique, is based on a stationary model for images. In other words, the Wiener filter assumes all pixels in an image to have similar statistics. Almost all practical images violate this basic assumption of data stationarity [Chan85]. As a result the Wiener filter can not effectively adjust itself to the local features of an image and generally smoothes the image.

In the soft thresholding method [Weaver91] we process the image in a global sense using the wavelet transform to decompose the signal into a hierarchy of detail images. Each detail image corresponds to a specific, non-overlapping frequency band. Based on the energy of noise in each frequency band a threshold is set for each detail image and used to modify the wavelet coefficients in that detail image. In Weaver's method [Weaver91], no attempt was made to adapt the filter to the local features of the image. As

a result, as with the Wiener filter, this method also tends to blur the edges. However, because of the small support of wavelets (in comparison to sinusoids) the artifacts caused by wrong modification of coefficients are less noticeable.

In Xu's method [Xu94], the wavelet transform is used to perform a local analysis on the signal. The resulting algorithm is a modified low-pass filter. The high frequency components are mostly suppressed except where an edge is detected. The criterion to detect an edge is the observation that edges have a larger correlation across the scales than the noise. In practice however, Xu noticed that involving more than three scales in calculating the correlation has poor results.

In our noise-filtering algorithm we introduced several major modifications to Xu's method. First of all, we process the image in both the spatial and wavelet domains to detect edges more accurately. We proposed some modifications to Xu's method which allowed us to calculate the correlation across more scales. As we saw in Section 5.3.2, this leads to a better distinction of edges from noise. In addition we used our *a priori* knowledge about edges to distinguish noise. Moreover, unlike Xu's method which considers the pixels as either noisy or clean (a binary decision), we took a more flexible approach in which all pixels can be noisy to some extent (fuzzy classification). As we saw in the previous section, these modifications lead to better noise filtering results.

Chapter 6

Practical Considerations

6.1 Introduction

In a large variety of image-processing techniques, the image is transformed from the spatial domain to a different space for processing. The extra effort required to transform the image to a different domain is often justified by the fact that some of the image features become more apparent and distinguishable in a non-spatial domain. As an example, there may be a better separation between the signal and noise power in the frequency domain, and hence a simple filter in the frequency domain can be used to improve the SNR of a noisy image. However, frequency domain techniques process an image in a global sense and hence they can not be easily adjusted to the local features of the image. As a result, reducing noise in an image using the frequency domain techniques involves a trade off between reducing the spatial resolution in the image (blurring) and noise reduction.

In contrast to applying direct frequency domain transforms to an image and losing the spatial information of the signal, application of the wavelet transform provides an interesting compromise between the spatial and frequency domain information. This transform maps the signal to a spatial-frequency space in which there is simultaneous access to both spatial and frequency domain information.

Many of the wavelet-based algorithms demonstrated in the literature require estimation of the image noise power prior to performing noise filtering. In this chapter we

shall show the effect of noise estimation bias on the performance of a number of wavelet-based noise filtering algorithms. This bias is typically ignored in the literature [Weaver91, Malfait97] or compensated for in an *ad hoc* manner [Xu94]. We shall demonstrate a straightforward approach to avoid the noise estimation bias. The technique is based on compensation methods that are used to generate unbiased SNR in MRI [Henkelman85, McGibney93]. We will also demonstrate how an adaptive technique may be used to compensate for the effect of noise biases on the filtering of noise from low-intensity regions of the image. Although noise bias effects are demonstrated in this chapter in the context of wavelet-based noise filtering, they are present in all image modalities.

In addition to the noise estimation bias, we will show how other problems associated with noise can affect the stability and validity of wavelet-based noise filtering algorithms. We present results showing the effect of amplitude quantisation on the performance of these algorithms. Finally, we examine a noise bias that can be introduced through the application of the FWT itself.

6.2 Demonstrating the presence of noise bias in the performance of wavelet algorithms

In Fig. 6.1(a) we show a test image with object intensities ranging from 0 to 255 grey levels. Superimposed in the four corners are uniform intensity areas. Areas of intensity 0 and 15 represent typical background or low-intensity image areas commonly used for noise power estimation. The other blocks of intensities 30 and 60 are meant to represent uniform, moderate-intensity regions that might be present in an image.

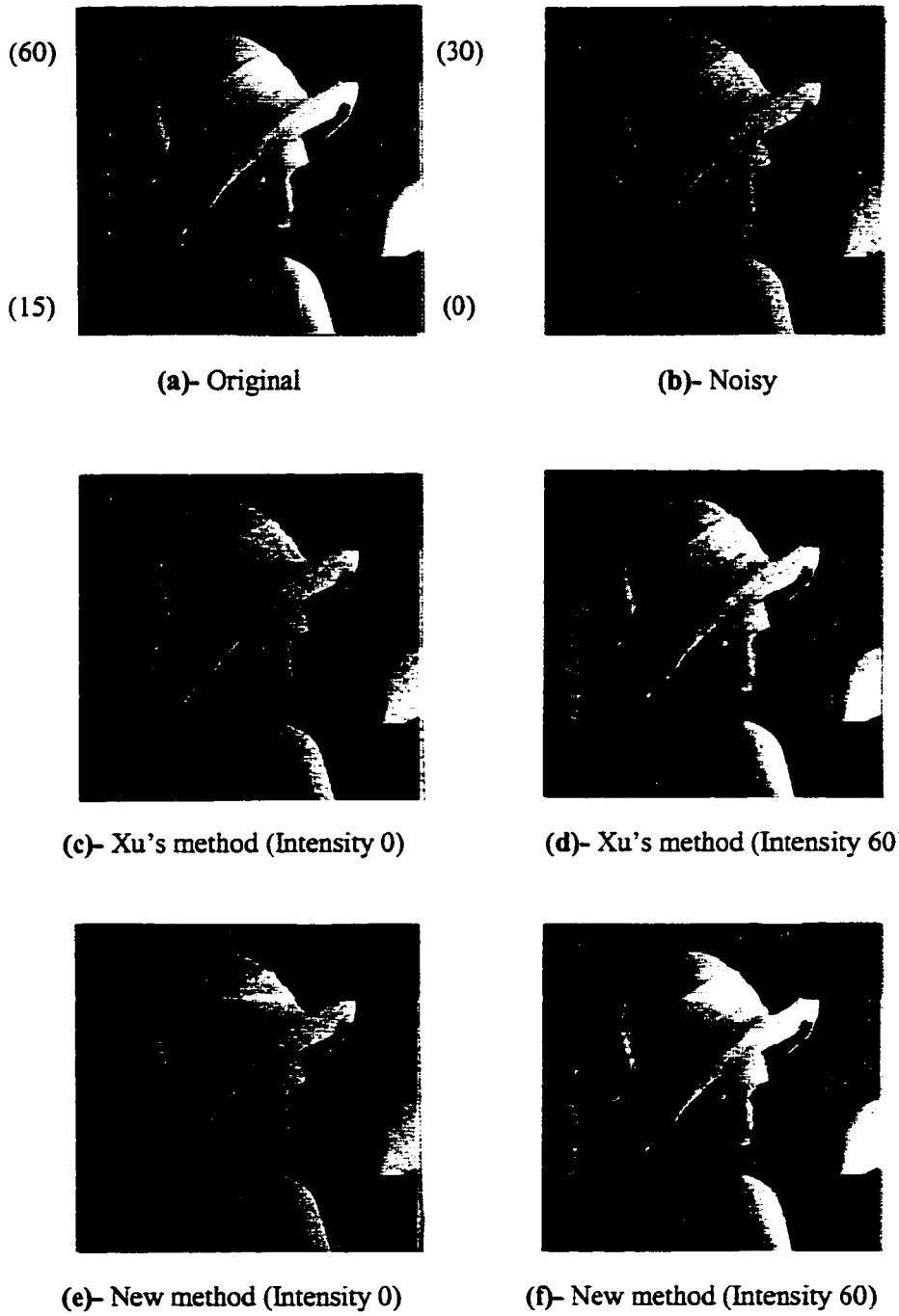


Figure 6.1 *Estimating the noise parameters from uniform regions with different intensities. (a) The original image with maximum grey-level of 255 after adding smooth regions of various intensities (60-30-15-0) to each corner of it. (b) After adding Gaussian noise with variance 400 (grey-scale)^2. Using the Xu's method to filter the image when noise parameters are estimated from: (c) a zero-intensity region (d) a moderate-intensity (with grey level 60) region. Using the new method to filter the image when noise parameters are estimated from: (e) a zero-intensity region (f) a moderate intensity (with grey level 60) region.*

Fig. 6.1(b) shows a typical noisy image generated by adding zero-mean white Gaussian noise of variance 400 to the original image prior to processing. Figs. 6.1(c) and 6.1(d) have been filtered with Xu's wavelet approach [Xu94] using two different noise estimation approaches. The noise level associated with Fig. 6.1(c) was estimated from the zero-intensity region as suggested by Xu *et al.* but without using their *ad hoc* compensation factor. This image shows less noise smoothing but sharper edges than that in Fig. 6.1(d), where the noise was estimated from a moderate-intensity, uniform region (amplitude 60).

In Chapter 5, we introduced a new wavelet-based algorithm that uses matched edge filters generated from both spatial and wavelet domain information. One advantage of this new approach over that suggested by Xu *et al.* [Xu94] is more successful preservation of the edges against blurring during noise filtering. The performance of this new method can also be modified by changing the image region used to determine noise power. However, the resulting changes are different than those for Xu's approach.

Fig. 6.1(e) and 6.1(f) show the filtered images with noise estimated from background (zero-intensity) and moderate-intensity (60) regions respectively. As with Xu's method, estimating the noise from the low-intensity region again produced less noise smoothing (Fig. 6.1(e)) than when estimating the noise from the moderate-intensity region. However, unlike Xu's approach, estimating the noise from the moderate-intensity region produced a noise-reduced image without edge blurring (Fig. 6.1(f)).

The previous discussion indicates that wavelet algorithms are sensitive to the manner by which the noise level is determined. We shall now show that this sensitivity is

associated with introducing a noise level bias. A technique for correcting this bias is illustrated using the wavelet-based algorithms discussed earlier.

6.3 How is the noise bias introduced

MRI is an interesting imaging modality. Its major advantages are associated with the ease of changing image view and image contrast by manipulation of the intensity and timing of the applied RF (Radio Frequency) pulses. MRI experimenters have been long aware of the effect of noise biases, especially when attempting to optimize imaging of low-intensity or low-contrast objects [Henkelman85, McGibney93].

The presence of the noise bias can be simply demonstrated by measuring the noise variance at different locations across the image. For this purpose, a synthetic image with uniform regions of various intensities (zero to 255) was generated and degraded by Gaussian noise. Next the noise variance and the apparent signal intensity at different locations were calculated. In Fig. 6.2(a) the apparent standard deviation of noise is shown as a function of the apparent signal intensity. It can be seen that measuring the noise upon a uniform, low-intensity region leads to incorrect noise parameters including under-estimation of the noise variance. The reason for this change in the noise lies directly in the image gathering procedures. If the zero-mean Gaussian noise present in the original scene was passed through to the display, the final image would contain both positive and negative pixel values. With few exceptions, such values are not found as most imaging modalities involve an implicit magnitude operation. The image gathering procedure modifies the noise characteristics especially in the low-intensity regions of the image. For example, the background noise is no longer zero-mean nor is it Gaussian.

The inaccurate estimation of the noise power has been discussed in the MRI context by Henkelman [Henkelman85] and McGibney and Smith [McGibney93]. This effect have also been mentioned by Xu *et al.* and compensated for in an *ad hoc* manner in their noise-filtering algorithm [Xu94].

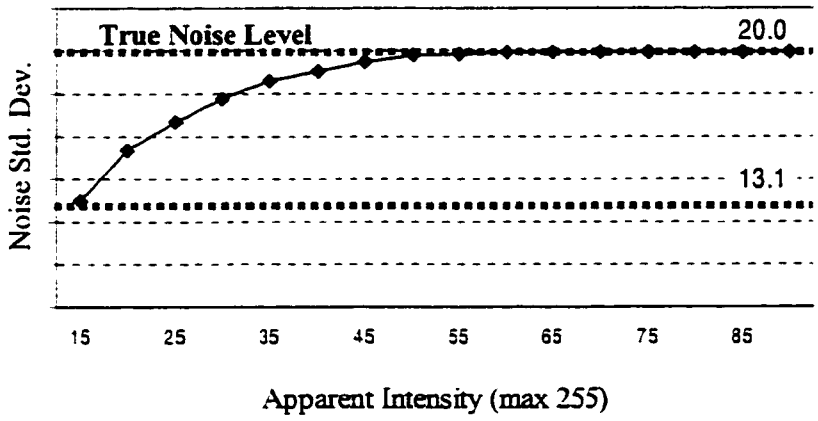


Fig. 6.2 *Apparent standard deviation of noise as a function of apparent image intensity. The original image had a maximum grey level of 255 and noise variance was 400 (grey-level)*^{2}.

In Fig. 6.3, we represent the apparent noise variance as a function of the apparent signal to noise ratio, SNR_{APP}, in order to make the graph independent of the specific maximum intensity of a particular image. The apparent *SNR* is defined by:

$$SNR_{APP} = (\textit{Apparent Signal Average}) / (\textit{Apparent Standard Deviation of Noise})$$

Note that because of the noise rectification that occurs in image regions of low intensity, it is not possible to have an SNR_{APP} below 1.3 measured directly from the standard (magnitude) image.

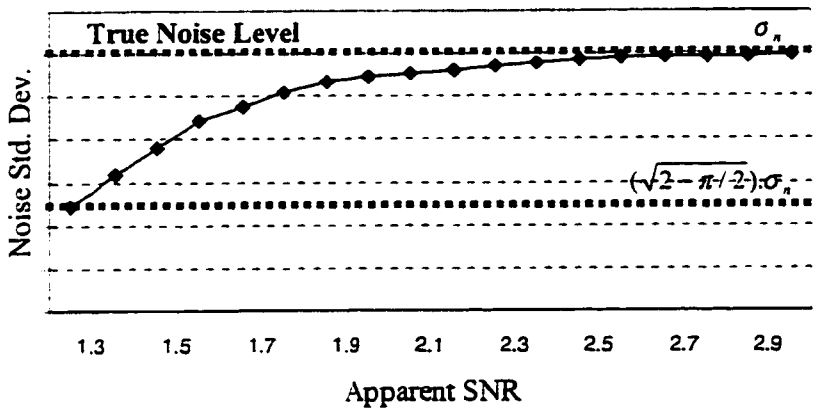


Fig. 6.3 Apparent standard deviation of the noise as a function of apparent SNR

From figures 6.2 and 6.3 it can be seen that computing the standard deviation of noise in a signal-free or background region, as suggested for many wavelet-based algorithms, provides an estimate that is approximately $\sqrt{2-\pi/2}$ times less than the actual standard deviation of the noise. This factor explains the necessity for Xu *et al.* to compensate noise measurements obtained using a background region. Xu *et al.* used an *ad hoc* "adjusting multiplicative factor" of 1.3 to compensate the noise bias. The true compensation factor should be equal to $1/(\sqrt{2-\pi/2})$ ($\equiv 1.53$) from Fig. 6.3. However, Xu *et al.* needed to deliberately under-compensate the effect of the noise bias in order to reduce edge blurring. The necessity of this trade off between noise filtering and edge blurring was demonstrated in Fig. 6.1.

In many images a suitable background area cannot be found and the noise power must be determined from an area of non-zero uniform intensity. Applying Xu's fixed, *ad hoc*, compensating factor to such an image will result in an over-estimation of the noise and increased blurring. In the next section we shall introduce a systematic approach to determine the noise level correction factor for any uniform image area.

6.4 Systematic Correction of Noise Bias

When the signal level in a uniform region is much larger than the standard deviation of noise, the noise statistics remain Gaussian and hence the standard deviation of noise estimated in this region is not biased. In most practical cases, however, it is usually impossible to find a uniform high-intensity region in the image. As the intensity of the signal in the uniform region approaches the magnitude of the noise, the noise will be changed to a Rayleigh distribution [Henkelman85]. The partial rectification of noise will lead to an under-estimation of true noise power. In the extreme case when the intensity of the uniform region is zero, before adding the noise, a correction factor of 1.53 ($=1/\sqrt{2-\pi/2}$), can be used to convert the apparent noise level to the true noise level. If such a zero-intensity region is not available, the noise level can be obtained from any uniform region using the relationship:

$$\sigma_{TRUE} = k.\sigma_{APP}$$

where the correction factor, k, is determined from Fig. 6.4 using the apparent SNR of the uniform region.

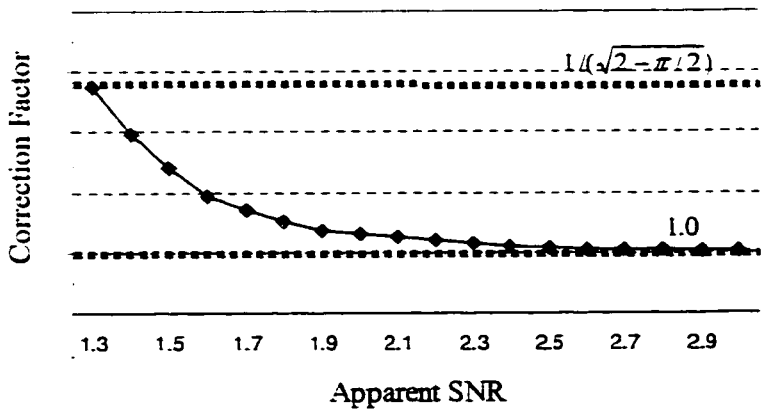


Fig. 6.4 The noise correction factor as a function of apparent SNR

The correction factor is only valid if both the apparent signal and the apparent noise level are measured in a uniform region where the signal intensity deviation is significantly smaller than the noise level. If there is no such uniform region in the image, the noise level will always be over-estimated, especially when the apparent SNR is small.

In wavelet-based methods we must adjust the estimated noise power for all detail images. The apparent noise associated with the background regions in each detail image will have lower levels than the noise in other regions. Therefore, using the lower biased noise level determined from a background area will not introduce sufficient noise smoothing for the remainder of the image.

In Fig. 6.5 we have compensated for the noise bias for both Xu's and our new wavelet denoising algorithms. The results are compared with non-compensated filtered images. As we can see from these figures, noise has been removed more effectively from the images after adjusting the noise level (with a factor of 1.53). However, adjusting for the true noise level has led to some edge blurring in Xu's method (Fig. 6.5(d)). This is mainly because there is a strong trade-off between noise filtering and edge blurring in Xu's method. To keep the edges sharp we must allow more noise in the output images. The sensitivity of edge blurring to the choice of noise power is not as evident for our algorithm (Fig. 6.5(f)).

6.5 Filtering Noise from Low-Intensity Portions of the Image in the Presence of the Noise Bias

As we saw in the previous sections, the noise statistics are distorted in the low-intensity regions of a magnitude image. The standard deviation of the partially-rectified

(a)- Original

(b)- Noisy

(c)- Xu's method

(d)- Xu's method (Compensated)

(e)- New method

(f)- New method (Compensated)

Figure 6.5 *Compensating the noise bias in Xu's and the new method (a) The test image with the range of grey-levels between 0 and 255. (b) after adding a Gaussian white noise with variance 400 (grey scale)^2 to the test image. Results of: (c) Xu's method (d) Xu's method with noise bias compensation. (e) The new method. (f) The new method with noise bias compensation.*

noise in these regions is smaller than the standard deviation of the true undistorted Gaussian noise. Moreover, the partial rectification of noise in low-intensity areas increases the average intensity of the image in these regions.

In the previous section we have shown how to obtain an unbiased estimate of the true noise level found in the high-intensity regions of the image. However, use of this or any noise estimate for noise filtering can introduce image distortion in unexpected ways. For instance, using the unbiased estimate of the noise implies that the noise power in the low-intensity regions (where the noise is distorted) will be over-estimated, leading to over-filtering and hence possible blurring of these regions.

We have shown that to avoid introducing systematic errors when performing noise-filtering, the strength of the filter must be adapted to correspond to changes in signal amplitude across the image. We have not yet developed an adaptive algorithm to compensate for such changes. However, by referring to one of the characteristics associated with MRI reconstruction, we can show that there will be practical advantages to using such an algorithm.

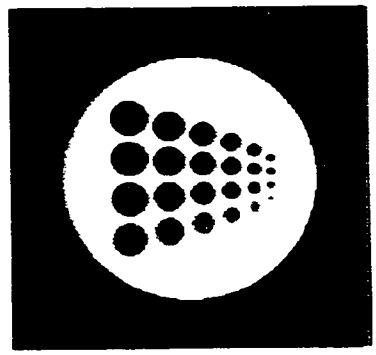
The raw data set associated with an MR image is gathered directly in the frequency domain (k-space) as complex data having both amplitude and phase. The k-space data are transformed using a 2-D DFT [Liang92] or a constrained modeling algorithm [Smith86, Liang92]. The resulting complex image is then displayed as a standard magnitude image. Normally, wavelet denoising or other signal processing techniques will be applied to the standard MR image with its associated noise bias. However, it is possible to process the real component of the complex MR image prior to its display. Since the real component image has pixels that can take both positive and negative values, it does not suffer from

any noise bias effects. By comparing the results of wavelet denoising of real component and standard images, it will be possible to determine the practical usefulness of an adaptive denoising algorithm applied to standard images.

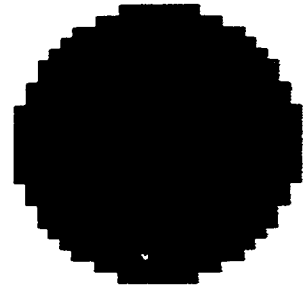
Fig. 6.6(a) shows an original synthetic test image, maximum intensity 140 grey levels, which approximates the phantoms used by Xu *et al.* [Xu94] and Liang *et al.* [Liang92]. The circles in the object contain small inserts (detail) whose intensity increases monotonically from the top of the image to the bottom. Fig. 6.6(b) shows a zoomed view of one detail. Noise of variance 400 was added to the phantom to produce a noisy standard image (Fig. 6.6(c)) and a noisy real component image (Fig. 6.6(d)).

There is no difference between Figs. 6.6(c) and 6.6(d) as the display operation converts the real component image with its negative and positive pixel values, to a standard image containing only positive pixel values. However, the denoised images do show significant differences.

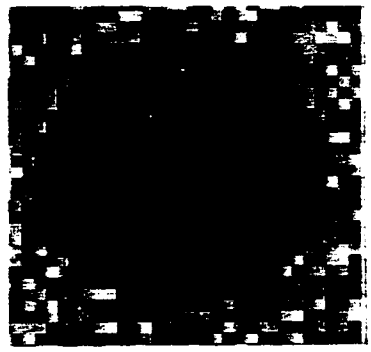
Denoising the magnitude image using the true noise power (Fig. 6.7(b)) suppresses both the noise and the detail since this approach does not compensate for the over-estimation of the noise power in the detail. Denoising the magnitude image using a local noise estimate associated with the background zero intensity reduces the noise in the background, reintroduces the detail, but leaves considerable noise in both the low- and medium-intensity regions (Fig. 6.7(c)). By contrast, denoising the real component image (Fig. 6.7(d)) shows reduction of noise in all portions and retains the detail present in the original image (Fig. 6.7(a)). This simplistic test indicates that denoising should be performed on either the real component image or in an adaptive fashion on magnitude images taking into account the local SNR.



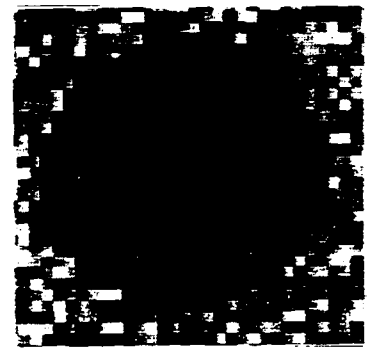
(a)- Original



(b)- Zoomed view

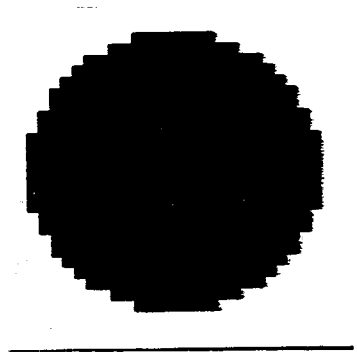


(c)- Noisy Standard

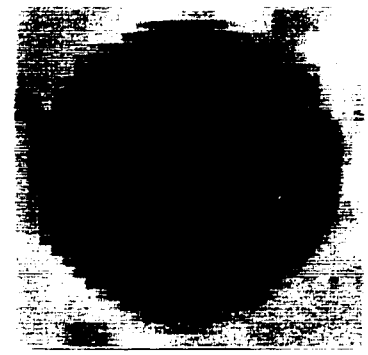


(c)- Noisy "Real"

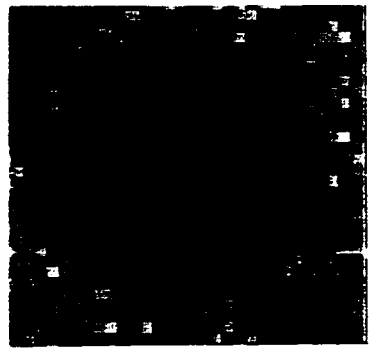
Figure 6.6 *Filtering noise from "Real" and Standard (Magnitude) images. (a) The synthetic test image with the range of grey levels between zero and 140. (b) A zoomed view of the original image (second circle in the first column). Adding Gaussian white noise with variance of 400 (grey level)^2 to the image: (c) Noisy Standard (Magnitude) image. (d) Noisy "Real" image.*



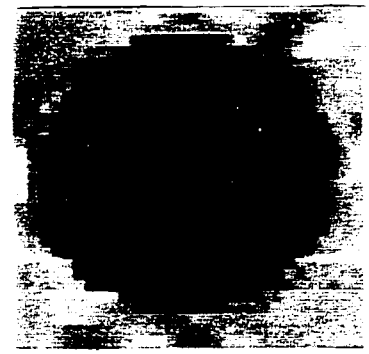
(a)- Original



(b)- Filtering the standard image using the true noise power



(c)- Filtering the standard image using a local noise estimate



(d)- Filtering the "real" image

Figure 6.7 *Filtering noise from "real" and standard (magnitude) images. (a) A zoomed view of the synthetic test image given in Fig. 6.6(a). (b) Filtering noise from the noisy standard image (given in Fig. 6.6(c)) using the true noise power. (c) Filtering noise from the noisy standard image using a local noise estimate associated with the zero intensity region. (d) Filtering noise from the noisy "real" image.*

Working with real component images, with their negative and positive pixel values and increased contrast, is possible in a number of imaging modalities which reconstruct the final image from an intermediate data set. However this approach is not always practical. For example, complex MR images typically show data in both real and imaginary components because of experimental phase "roll-off". This roll-off complicates the detection of image edges. Various image phase correction algorithms have been suggested, but are not always effective [Liang92]. The wavelet domain analysis provides all the necessary tools to process the magnitude image, and adapt the wavelet-based noise filter to the local behaviour of the noise, but yet avoid the need for image phase correction.

Measuring the local noise power for use in an adaptive wavelet denoising algorithm is not straightforward, as the local neighbourhood pixel intensities may not be uniform. In this situation, the best estimate of the local noise level is obtained by reducing the true noise level by a factor, k', determined by the local SNR:

$$\sigma_{LOCAL} = \sigma_{TRUE}/k'$$

When the local neighbourhood has uniform intensity then the factor k can be determined from Fig. 6.4. However, a more general approach must be used in most situations. Fig. 6.5 shows the necessary correction factor k' as a function of an SNR, $SNR_{UNBIASED}$, defined to take account of both noise rectification and non-uniformity of the neighbourhood intensities [McGibney93]:

$$SNR_{UNBIASED} = \sqrt{\frac{S_{Power}}{N_{Power}}}$$

where:

$$S_{Power} = \frac{1}{N_R}\sum_{j \in R} M_j^2 - \sigma_{TRUE}^2 \qquad \text{and} \qquad N_{Power} = \sigma_{TRUE}^2$$

where N_R is the number of pixels in the region of measurement, R, M_j is the apparent intensity of signal at point $j \in R$, and σ_{TRUE}^2 is the undistorted noise variance which is estimated from a uniform region (background) of the image. This calculation approach to determine an unbiased SNR value is more reliable than using a lookup table [Henkelman85, Henkelman86] for non-uniform intensity regions because no one point on the lookup table is valid in a non-uniform region [McGibney93].

After estimating the $SNR_{UNBIASED}$ in each neighbourhood, we use a curve similar to the one shown in Fig. 6.5 to estimate the apparent noise level in that neighbourhood. Finally we use this estimate of the apparent noise level to adapt the denoising filter to the local characteristics of the noise.

6.6 The Effect of Grey-Level Quantisation

In general, image acquisition methods always lead to an image function $f(x,y)$, which is digitised both spatially and in amplitude. Digitisation of the spatial coordinates (x,y) is called image sampling, and amplitude digitisation is called grey-level quantisation.

Grey-level quantisation introduces some error in the intensity of pixels: The pixel values will always be rounded up or down to the closest quantisation level. We usually refer to this type of error as the quantisation error. The size of the quantisation error for each pixel is between zero and $\Delta L/2$ where ΔL is the difference between the intensities of adjacent quantisation levels. Therefore, to decrease the quantisation error, we should

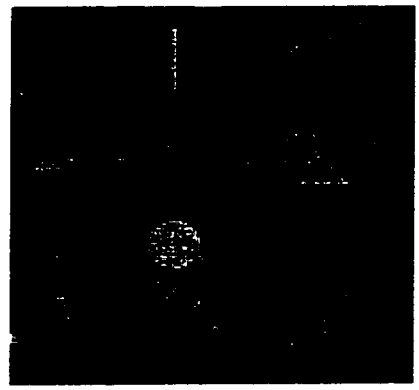
decrease the difference between the intensities of adjacent levels or equivalently increase the number of quantisation levels (grey levels).

Grey-level quantisation may also affect the performance of our noise filtering technique. Because of the quantisation error, the distribution of noise that is originally Gaussian would take a different shape. This is especially true in high-SNR images, where the standard deviation of noise is small and comparable in size with the maximum quantisation error.

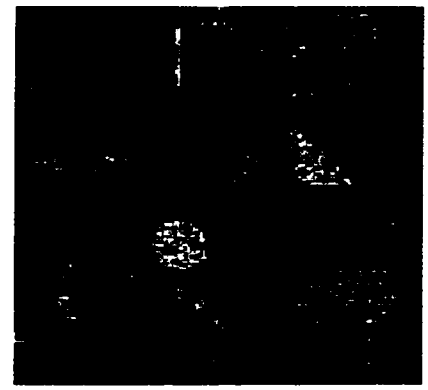
In this section we will examine the effect of the quantisation error on the performance of our noise filtering technique and extend the conclusions in a general fashion to the wavelet algorithms. For this purpose we will quantise our test image using two different numbers of quantisation levels, 256 and 1024. Then we will apply our method to both images and compare the results. We will perform this test for both low and high noise levels.

Figs. 6.8(a) shows the test image after adding a Gaussian white noise with a variance of 25. We used two different number of grey-levels (256 and 1024) to quantise this test image and then applied our noise filtering method to each quantised image separately. Figs. 6.8(b) and 6.8(c) show the results after filtering the noise from the quantised test images (for 256 and 1024 grey levels respectively). The visual quality of the filtered images is almost the same. However, the MSE measure (Table 6.1) shows that the noise is removed more effectively from the image with the larger number of grey-levels.

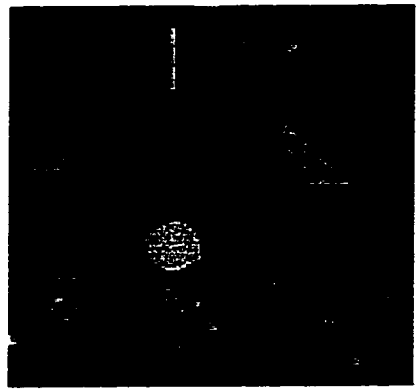
Next we degrade the test image by adding much stronger levels of noise ($\sigma_n^2 = 400$). Again we use two different numbers of levels to quantise the results (256 and 1024 levels). Figs. 6.8(e) and 6.8(f) show the results after applying the new filter to the 256 and



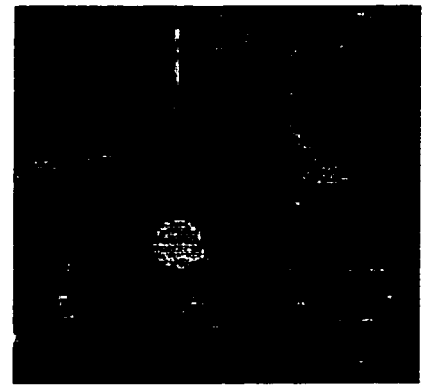
(a)- High-SNR test image



(d)- Low-SNR test image

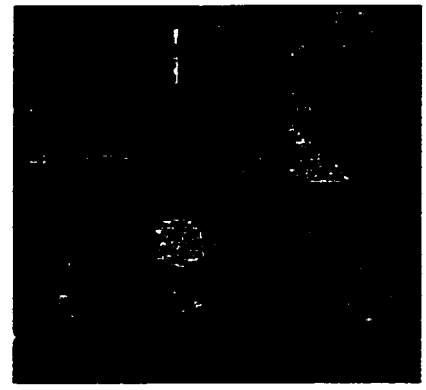


(b)- Applying the new method to the high-SNR image (256 quantisation levels)



(e)- Applying the new method to the low-SNR image (256 quantisation levels)

(c)- Applying the new method to the high-SNR image (1024 quantisation levels)



(f)- Applying the new method to the low-SNR image (1024 quantisation levels)

Figure 6.8 *The effect of quantisation error on high- and low-SNR test images (a) The high-SNR test image with noise variance of 25 (grey level)^2. (b) Filtering the high-SNR image using 256 quantisation levels. (c) Filtering the high-SNR image using 1024 quantisation levels. (d) The low-SNR test image with noise variance of 400 (grey level)^2. (e) Filtering the low-SNR image using 256 quantisation levels. (f) Filtering the low-SNR image using 1024 quantisation levels.*

Figure	*Description*	*MSE* *(Grey level)*^{2}
6.8(a)	Additive noise , σ^2 = 25 (grey level)^2	24.3 ± 0.4
6.8(b)	Filtering the noise from 6.8(a) with 256 quantisation levels	10.3 ± 0.1
6.8(c)	Filtering the noise from 6.8(a) with 1024 quantisation levels	10.1 ± 0.1
6.8(d)	Additive noise , σ^2 = 400 (grey level)^2	393.0 ± 1.7
6.8(e)	Filtering noise from 6.8(d) with 256 quantisation levels	71.5 ± 0.3
6.8(f)	Filtering noise from 6.8(d) with 1024 quantisation levels	71.2 ± 0.3

Table 6.1 *The effect of grey-level quantisation in removing the noise from the noisy test image.*

1024 grey-level noisy images. Neither qualitative analysis nor quantitative measures (Table 6.1) can show a major difference between these filtered images.

In summary, the quantisation error may have some effect on the quality of the filtered image only if the standard deviation of noise is small and comparable to the size of the quantisation step. In fact, for images corrupted by low levels of noise, changing the number of quantisation levels can cause obvious changes in the quantitative measures of the final filtered image. However, for the low-noise images, the visual quality of the filtered images, regardless of how we quantise the noise, is quite acceptable for almost all practical applications. Therefore, in general, changing the number of quantisation levels does not have a noticeable impact on the quality of the filtered images.

6.7 Choosing the Size and Position of the Signal-Free Region

As we saw in the previous chapter, we use the signal-free regions near boundaries of the image to estimate the noise parameters. This approach is based on the assumption that we can always find dark regions near the corners of the image. In the following paragraphs, we refer to this signal-free region as the "reference region".

Our method currently uses a fixed-size rectangular region near one of the image corners as the reference region to determine the local noise level and then correct this level for noise bias. However, it is possible to use more sophisticated techniques that manually or automatically adapt the shape and size of this region to the characteristics of the image under consideration.

To choose a proper size and shape for the reference region, we must consider several parameters. First of all, to have a good estimation of the noise variance the size of the region must be large. However, as we saw in Chapter 3, the computation of the wavelet coefficients are based on successive convolution of the image with the impulse response of the high-pass and low-pass filters associated with the wavelet. Because of these successive convolutions, the discontinuities in the boundaries of the image can spread into the reference region. This spreading will result in a biased estimation of the noise parameters.

As we saw in Chapter 3, calculation of the wavelet coefficients at each level requires application of a one-dimensional convolution to each row and each column of the image. Each convolution will extend the size of a row (or column) by F-1, where F is the length of the high-pass or low-pass filter associated with a wavelet. Therefore, by using small support wavelets, which have small F values, we can limit the spread of the borders into the reference region. Moreover, for an N-level decomposition of the signal, the reference region must be $N \times (F-1)$ pixels apart from the image boundaries. In practice, however, we noticed that the reference region could be chosen closer to the image borders. In fact most of the energy of noise is concentrated in low-scale detail images (the wavelet

coefficients in low levels of the decomposed signal). For the detail images the spread of the borders into the image is much less than $N \times (F-1)$.

In some special cases, such as when the background region is too small, we are forced to estimate the noise from regions that are quite close to image boundaries. In these cases, we can extend the image from each side to smooth the sharp transitions in the boundaries of the image. For this purpose, we can either duplicate the border lines of the image or estimate the average background intensity (by choosing some sample regions in the background) and use this average intensity to extend the image.

6.8 Conclusion

In this chapter we have seen that measuring the noise from low-intensity regions of the image would result in an under-estimation of the noise level. This under-estimation has been ignored in the literature. The effect of noise estimation bias on the performance of a number of wavelet-based noise filtering algorithms was investigated. Finally a systematic approach for compensating the noise bias in the wavelet-based noise filtering methods was presented. It was shown, using "Real" images from MRI, that an adaptive filtering method based on local SNR would be useful.

We also have presented results showing the effect of amplitude quantisation on the performance of the filter. It was shown that in general, changing the number of quantisation levels does not have a noticeable impact on the quality of the filtered images. We also examined the practical considerations that should be taken into account when we choose the size and the position of the noise estimation region because

successive convolutions involved in the FWT algorithm spread the discontinuities in the boundaries of the image into the noise estimation region.

Chapter 7

Conclusions and Future Considerations

7.1 Conclusions

The purpose of this thesis was to find a proper method for filtering the noise from images without blurring their edges. Several existing techniques for removing the noise were investigated and a number of new approaches introduced.

Noise filtering is an important step in many image-processing applications. The key function of noise filtering techniques is to improve the image in ways that facilitate extracting information from the image. We have shown that noise smoothing and edge-enhancement are inherently conflicting processes. Smoothing a region may destroy edges which carry much of the image information. However, sharpening edges might lead to unnecessary noise being retained or enhanced.

In this research, we concentrated on spatial and scale-space domain noise filters. We have shown that these classes of filters have characteristics that can be adapted to local features of the given image. Such filters can be used to process edges and other sharp features of the image in a different way than the rest of the image.

The spatial domain filters were the first group of filters to be investigated. We concentrated on a specific group of spatial domain filters which was based on LLMMSE estimation of signals. The LLMMSE estimator was shown to have the property that noise

in flat regions of the image are smoothed while observations are left unchanged in the vicinity of edges.

To remove the noise more effectively from the vicinity of edges, several modifications to the basic LLMMSE method have been proposed in recent years [Lee80, Lee81, Rajala81, Kuan85, Jiang86, Paranjape94, Das97, Rangayyan98b]. We classified these LLMMSE based filters in two main groups: fixed and advanced neighbourhood filters. Fixed neighbourhood techniques such as Lee's and Kuan's methods were demonstrated to be not very successful in removing the noise from the edge areas. As examples of the advanced neighbourhood method, Lee's refined neighbourhood method [Lee81] and the adaptive neighbourhood method [Rangayyan98b], were discussed. The advanced neighbourhood methods were more successful in removing the noise from the vicinity of edges in comparison to fixed neighbourhood methods.

We presented a new LLMMSE based method for filtering the noise from low-contrast or low-SNR images. The proposed method is a modification of the adaptive neighbourhood method of Rangayyan [Rangayyan98b]. The modification was to add a pre-processing step to improve the accuracy of estimating the local statistics of the image. We compared our approach with other LLMMSE methods, by applying them to some low-SNR images. In the presence of strong noise, the results of the new method have proven to be better in terms of visual quality and MSE measures.

The wavelet-based noise filters were the second group of filters that were discussed in this thesis. As was shown in Chapter 4, wavelet analysis is a powerful tool that can provide an interesting compromise between the spatial and frequency domain information of signals. The wavelet transform maps the signal to a scale-space domain in which there is

simultaneous access to both spatial and frequency domain information. The ability of the wavelet transform to preserve the spatial domain information makes it a very attractive tool for analysing non-stationary signals.

Several wavelet-based noise filtering techniques that have been proposed during recent years [Weaver91, Mallat92, Xu94, Malfait97] were investigated and compared. A brief review of wavelet theory was given to provide the required background for these methods.

The general idea behind almost all the wavelet-based methods is to identify edges and other sharp features from the noise and to preserve them during low-pass filtering. Two different approaches for detecting the edges of an image from its wavelet coefficients were discussed. The first technique is based on examining the behaviour of the modulus maxima of the coefficients across the scales; a complicated and time-consuming approach [Mallat92]. In the second technique, the correlation of the wavelet coefficients across the scales was used to detect the edges [Xu94].

We proposed a new method for suppressing the noise in images. This new method has some major differences from existing wavelet approaches. The new technique processes the image in both scale-space and spatial domains. Moreover, it does not classify the wavelet coefficients into noisy and noiseless coefficients but considers each coefficient to be partially noisy. A major step in the proposed method is to determine to what extent each coefficient has been affected by noise.

In the first phase of the new approach, the images are processed to locate their edges, and next, the information about the edges is used to smooth out the noise without blurring the image. A new edge detection method was introduced to detect the edges more

effectively. The proposed method can detect most of the image edges successfully even in the presence of strong noise as it employs spatial and scale-space domain techniques as well as the *a priori* information about the geometrical features of the edges.

A new edge-growing algorithm was introduced to permit detection of edges in the spatial domain. We applied a double thresholding method to the gradient of the image to locate the seed pixels for growing the edges as well as all other pixels that have a chance for being on an edge (candidate edge pixels). A new method was used to assign a probability measure to each candidate pixel to show its chance for being on an edge. For this purpose we measured the similarity between each candidate pixel and the seed pixels, introducing new measures for calculating the similarity of the pixels. To translate the computed similarities into probabilities, fuzzy classifiers were introduced. Finally a probabilistic formulation was used to assign an overall probability factor to each candidate pixel. This multiple probability approach results in "Edge Probability" images in which the intensity of each pixel represents the chance of that pixel of being on an edge. This is different from more common approaches in which pixels are considered to be either on edges or not (binary decision).

To detect the edges in the scale-space domain, we introduced some modifications which led to a great improvement in the performance of Xu's method [Xu94]. First, to highlight the weak edges, we compressed the dynamic range of the correlation images. Moreover, we added post-processing steps to Xu's method in which *a priori* knowledge about the geometrical characteristics of the image was used to facilitate the recognition of edges from the noise.

It was shown that our spatial domain edge detector is more successful in detecting weak edges than our wavelet-based edge detection method. However, the wavelet-based edge detector is more sensitive to noise. To take benefit of the positive features of both techniques, i.e. accuracy and insensitivity to noise, a probabilistic formulation was used to mix the edge information obtained in the spatial and scale-space domains.

In our new noise filtering technique, we used the proposed edge detection algorithm as a pre-processing step. After detecting the edges, we mapped it to the scale-space domain using the same transform as the one that we used to map the noisy image itself. The result was a hierarchy of detail images in which the distribution of the wavelet coefficients shows how the energy of edges would spread in the scale-space domain. As a new approach, we used these detail images as a bank of masks for filtering the corresponding components, i.e. detail images, of the noisy image. The preliminary results were given to compare this approach with other methods in terms of preserving image sharpness during noise filtering.

Finally the effect of noise estimation bias on the performance of a number of wavelet-based noise filtering techniques was discussed. Many of the noise filtering algorithms demonstrated in the literature require estimation of the image noise power prior to performing noise filtering. We have shown that measuring the noise from low-intensity regions of the image would result in an under-estimation of the noise level which is usually ignored in the literature. A systematic approach for compensating the noise bias in the wavelet-based noise filtering methods was presented. The technique was based on compensation methods that are used to generate unbiased SNR in MRI. An adaptive filtering method based on local SNR was shown to be useful by modeling the effect of the new adaptive algorithm on standard images by processing "Real" images from MRI.

We also have presented results showing the effect of amplitude quantization on the performance of the filter. It was shown that in general, changing the number of quantization levels does not have a noticeable impact on the quality of the filtered images. We also examined the practical considerations that should be taken into account when we choose the size and the position of the noise estimation region because successive convolutions involved in the FWT algorithm spread the discontinuities in the boundaries of the image into the noise estimation region.

7.2 Future Considerations

Throughout this thesis, several noise filtering methods were discussed. In almost all these methods, we tried to remove the noise from images without blurring the edges. The basic idea behind all these methods was to design spatially-selective, low-pass filters which remove the high-frequency components from the smooth areas but leave them unchanged in the vicinity of edges.

In practice however, it is often possible to find textured regions in images that cannot be classified as either edge or smooth area. The methods discussed in this thesis cannot distinguish texture from noise and hence have the tendency to smooth textured regions.

We need to extend our technique to make it sensitive to textured regions. One approach might be to find a method to distinguish textured regions from noise-degraded smooth areas. For this purpose, the wavelet coefficients associated with each region can be examined. Without noise, smooth regions have strong components only in the low frequencies (large scales). However, textured regions have substantial components in a wide frequency (scale) spectrum. The high-frequency components associated with noise

can be characterized by examining a signal-free region of the image. By comparing the frequency (scale) spectrum of different regions with that of a signal-free region, we can recognise textured regions. Moreover, if the texture does not have a random distribution, for example if it has a dominant orientation, the high-frequency components would be larger in a specific group of detail images. This characteristic can be used to distinguish a textured region.

After locating textured regions, we have to examine the possible methods for removing the noise from textured regions without smoothing them. The simplest solution might be subtracting the noise spectrum from the spectrum of the textured region.

We see other possible improvements of the method proposed in this thesis. It was shown that distortion of noise in the low-intensity regions of a magnitude image would result in bias in the estimation of the noise level. We presented a systematic approach for compensating this noise bias; however, the adjusted level of noise is not valid for low-intensity regions of the image where the noise is distorted. To avoid introducing systematic errors when performing noise filtering, the strength of the filter must be adapted to correspond to changes in signal amplitude across the image. In the next step, we will try to develop an adaptive algorithm to compensate for such changes.

In general it can be said that the new wavelet-based method works well in removing noise from images. The quality of the result, however, is ultimately based on our specific image processing application. Only with an extensive study of many images which are judged by specialists in each field, the true value of the suggested filtering technique can be determined.

Bibliography

[Allen77] J. B. Allen and L. R. Rabiner, "A unified approach to short time Fourier analysis and synthesis," *Proc. IEEE*, vol. 65, no. 11, pp. 1558-1564, 1977.

[Andrews 77] H. C. Andrews and B. R. Hunt, *Digital Image Restoration*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[Banham96] M. R. Banham and A. K. Katsaggelos, "Spatially adaptive wavelet-base multiscale image restoration," *IEEE Transactions on Image Processing*, vol. 5, pp. 619-634, April 1996.

[Bracewell 95] R. Bracewell, *Two-Dimensional Imaging*. Englewood Cliffs, NJ: Prentice-Hall, pp. 560-619, 1995.

[Castleman 79] K. R. Castleman, *Digital image processing*. Englewood Cliffs, NJ: Prentice Hall, 1979.

[Cannay 86] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-643, Nov. 1986.

[Chan 85] P. Chan and J. S. Lim, "One dimensional processing for adaptive image restoration," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-33, pp. 117-126, 1985.

[Clark 89] J. J. Clark, "Authenticating edges produced by zero-crossing algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 8, pp. 830-831, 1989.

[Cohen 92] A. Cohen, I. Daubechies and J. C. Feauveau, "Biorthogonal basis of compactly supported wavelets," *Comm. Pure Appli. Math.*, vol. 45, pp. 485-560, 1992.

[Croisier 76] A. Croisier, D. Esteban and C. Galand, "Perfect channel splitting by use of interpolation, decimation, tree decomposition techniques," *Int. Conf. On Information Sciences/Systems, Patras*, pp. 443-446, Aug. 1976.

[Das 97] A. Das and R. M. Rangayyan, "Adaptive region-based filtering of multiplicative noise," *Proc. SPIE 3026, on non-linear image processing* VIII, pp. 338-348, Feb. 1997.

[Daubechies 90] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. on Info. Theory*, vol. 36, no. 5,

pp. 961-1005, Sept. 1990.

[Daubechies 92] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.

[Donoho 95a] D. L. Donoho, "Denoising by soft thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, March 1995.

[Donoho 95b] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian and D. Picard, "Wavelet shrinkage: Asymptopia," *Jour. Roy. Stat. Soc., Series B*, vol. 57, no. 2, pp. 301-369, 1995.

[Edelstein 84] W. A. Edelstein, P. A. Bottomley and L. M. Pfeifer, "A signal to noise calibration procedure for NMR imaging systems," *Med. Phys.*, vol. 11, pp. 180-185, 1984.

[Fram 75] J. R. Fram and E. S. Deutsch, "On the quantitative evaluation of edge detection schemesand their comparison with human performance," *IEEE Transactions on Computers*, vol. C-24, no. 6, pp. 616-628, 1975.

[Gabor 46] D. Gabor, "Theory of communication," *Journal of the IEE*, vol. 93, pp. 429-457, 1946.

[Gonzalez 92] R. C. Gonzalez and R.E.Woods, *Digital Image Processing*. Reading, Mass.: Addison-Welsey, 1992.

[Grossman 84] A. Grossman and J. Morlet, "Decomposition of hardy functions into square integrable wavelets of constant shape," *SIAM. J. Math*, vol. 15, pp. 723-736, 1984.

[Harris 78] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51-84, 1978.

[Henkelman 85] R. M. Henkelman, "Measurement of signal intensities in the presence of noise in MR images," *Med. Phys.*, vol. 12, pp. 232-233, 1985.

[Henkelman 86] R. M. Henkelman, "Erratum: Measurement of signal intensities in the presence of noise in MR images," *Med. Phys.* vol. 13, pp. 544-545, 1986.

[Henkelman 87] R. M. Henkleman and M. J. Bronskill, "Artifacts in Magnetic Resonance Imaging," *Rev. Magn. Reson. Med.* vol. 2, pp. 1-126, 1987.

[Hunt 76] B. R. Hunt and T. M. Cannon, "Non-stationary assumptions for Gaussian models of images," *IEEE Trans. Sys., Man, Cybern.*, vol. SMC-6, pp. 876-881, Dec.1976.

[Jiang 86] S. S. Jiang and A. A. Sawchuk, "Noise updating repeated Wiener filter and other adaptive noise smoothing filters using local image statistics," *Applied Optics*, vol. 25, no. 14, pp. 2326-2337, July 1986.

[Kosko 92] B. Kosko, *Neural Networks and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[Kuan 82] D. T. Kuan, "Nonstationary 2-D recursive restoration of images with signal dependent noise with application to Speckle Reduction," Ph.D. thesis, Univ. of Southern California, Los Angeles, 1982.

[Kuan 85] D. T. Kuan, A. A. Sawchuck, T. C. Strand and P. Chavel, "Adaptive noise filtering for images with signal-dependent noise," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 2, pp. 165-177, March 1985.

[Lee 80] J. S. Lee, "Digital image enhancement and noise filtering by use of local statistics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, no. 2, pp. 165-168, March 1980.

[Lee 81] J. S. Lee, "Refined filtering of image noise using local statistics," *Computer Graphics and Image Processing*, vol. 15, pp. 380-389, 1981.

[Lemarie 86] P. G. Lemarie and Y. Meyer, "Lecture notes in mathematics," *Rev. Mater. Ibero Amer.* 2, 1986.

[Lev 77] A. Lev, S. W. Zucker and A. Rosenfeld, "Iterative enhancement of noisy images," *IEEE Trans. On Systems, Man, and Cybernetics*, vol. SMC-7, no. 6, pp. 435-442, June 1977.

[Levine 85] M. D. Levine, *Vision in Man and Machine*. New York, NY: McGraw-Hill, 1985.

[Liang 92] Z. P. Liang, F. E. Boada, R. T. Constable, E. M. Haacke, P. C. Lauterbur and M. R. Smith, "Constrained reconstruction methods in MR Imaging," *Rev. Magn. Reson. Med.*, vol. 4, pp. 67-185, 1992.

[Lim 90] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1990.

[Mahesh 90] B. Mahesh, W. J. Song and W. A. Pearlman, "Adaptive estimators for filtering noisy images," *Opt. Eng.*, vol. 29, pp. 489-494, 1990.

[Malfait 97] M. Malfait and D. Roose, "Wavelet-based image denoising using a Markov random field a priori model," *IEEE Transactions on Image Processing,* vol. 6, no. 4, pp. 549-565, April 1997.

[Mallat 89a] S. G. Mallat, "A theory for multi-resolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp. 674-693, July 1989.

[Mallat 89b] S. G. Mallat and S. Zhong, "Complete signal representation with multi-scale edges," *New York University Technical Report*, no. 483, December 1989.

[Mallat 91] S. G. Mallat and S. Zhong, "Characterization of signals from multi-scale edges," *New York University Computer Science Technical Report*, November 1991.

[Mallat 92] S. G. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 617-637, March 1992.

[Mallat 98] S. G. Mallat, *A Wavelet Tour of Signal Processing*. New York, NY: Academic Press, 1998.

[Marr 80] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. R. Soc. Lond.*, vol. B207, pp. 187-217, 1980.

[Matlab 96] *Wavelet Toolbox User's Guide*. Cochituate place, Mass.: The MathWorks Inc.1996.

[McGibney 93] G. McGibney and M.R. Smith, "An unbiased signal to signal to noise measure for magnetic resonance images," *Med. Phys.*, vol. 20, no. 4, pp. 1077-1078, Jul/Aug 1993.

[Meyer 91] Y. Meyer, *Un Contre-Exemple a la Conjecture de Marr et a Celle de S. Mallat,* Paris: Hermann, preprint 1991.

[Papoulis 65] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, New York, NY: McGraw-Hill, 1965.

[Paranjape 94a] R. B. Paranjape, T. F. Rabie, R. M. Rangayyan, "Image restoration by adaptive neighborhood noise subtraction." *Applied Optics*, vol. 33, no. 14, pp. 2861-2869, May 1994.

[Paranjape 94b] R. B. Paranjape, R. M. Rangayyan, W. M. Morrow, "Adaptive neighborhood mean and median filtering," *Journal of Electronic Imaging*, vol. 3, no. 4, pp. 360-367, Oct.1994.

[Pentland 82] A. P. Pentland, "Visual inference of shape: Computation from local features," Ph.D. Dissertation, Dep. Psychol., Massachusetts Institute of Technology, Cambridge, MA, 1982.

[Pitas 92] I. Pitas and A. N. Venetsanopoulos, "Order statistics in digital image processing," *Proc. IEEE*, vol. 80, pp. 1893-1923, 1992.

[Portnoff 80] M. R. Portnoff, "Time frequency representation of digital signals and systems based on short time Fourier analysis," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, pp. 55-69, Feb.1980.

[Pratt 91] W.K. Pratt, *Digital Image Processing*, 2nd edition, New York, NY: John Wiley & Sons, 1991.

[Prewitt 70] J. M. S. Prewitt, *Object Enhancement and Extraction, in Picture Processing and Psychopictorics*. New York, NY: Academic Press, 1970.

[Rajala 81] S. A. Rajala, R. J. P. de Figueiredo, "Adaptive nonlinear image restoration by a modified Kalman filtering approach," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 1033-1042, Oct. 1981.

[Rangayyan98a] R. M. Rangayyan, A. Das, "Filtering multiplicative noise in images using adaptive region-based statistics," *Journal of Electronic Imaging*, vol. 7, no. 1, pp. 222-230, 1998.

[Rangayyan98b] R. M. Rangayyan, M. Ciuc and F. Faghih, "Adaptive neighborhood filtering of images corrupted by signal-dependent noise," *Applied Optics*, vol. 37, no. 20, pp. 4477-4487, July 1998.

[Rioul 91] O. Rioul and M.Vetterli, "Wavelets and signal processing," *IEEE SP Magazine*, pp. 14-38, Oct. 1991.

[Rosenfeld 70] A. Rosenfeld, "A nonlinear edge detection technique," *Proc. IEEE (Lett.)*, vol. 58, pp. 814-816, May 1970.

[Rosenfeld 71] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Transactions Comput.*, vol. C-20, no. 5, pp. 562-569, May 1971.

[Sage71] A. P. Sage, J. L. Melsa, *Estimation Theory with Applications to Communications and Control*, NewYork, NY: McGraw-Hill, 1971.

[Smith 86] M. R. Smith, S. T. Nichols, R. M. Henkelman and M. L. Wood, "Application of autoregressive moving average parametric modeling in magnetic resonance image reconstruction," *IEEE Trans. Med. Imaging*, vol. MI-5, pp. 132-139, 1986.

[Smith 98] M. R. Smith, F. Faghih, "Avoiding noise biases in wavelet-based denoising algorithms," under review, *IEEE Trans. Med. Imaging*, Dec. 1998.

[SmithM 86] M. J. Smith and T. P. Barnwell, "Exact reconstruction for tree-structured sub-band coders," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-28, pp. 434-441, June 1986.

[Strickland 97] R. N. Strickland and H. I. Hahn, "Wavelet transform methods for object detection and recovery," *IEEE Transactions on Image Processing*, vol. 6, no. 5, pp. 724-735, May 1997.

[VonAltrock95] C. VonAltrock, *Fuzzy Logic and Neuro-Fuzzy Applications Explained*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[Wahl 87] F. M. Wahl, *Digital Image Processing*. Norwood, MA: Artech house, 1987.

[Weaver 91] J. B. Weaver, Y. Xu, D. M. Healy and L. D. Cromwell, "Filtering noise from images with wavelet transforms," *Magnetic Resonance in Medicine*, vol. 21, pp. 288-295, 1991.

[Wehrli 88] F. W. Wehrli, J. R. McFall and J. H. Prost, "Impact of the choice of operating parameters on MR images," *in Magnetic Resonance Imaging*, edited by C. L. Partain, R. R. Price, J. A. Patton, M. V. Kulkarni, A. E. James, Jr., Saunders, Philadelphia, 1988.

[Witkin 83] A. Witkin, "Scale space filtering," *in Proc. 8th Int. Joint Conference Artificial Intelligence*, 1983.

[Xu 94] Y. Xu, J. B. Weaver, D. M. Healy Jr. and J. Lu, "Wavelet transform domain filters: A spatially selective noise filteration technique," *IEEE Transactions on Image Processing*, vol. 3, no. 6, pp. 747-757, Nov. 1994.

[Yang 96] J. Yang and M. R. Smith, "Constrained and daptive ARMA modeling as an alternative to the DFT, with application to MRI," *Control and Dynamic Systems*, vol.77, pp. 225-299, New York, NY: Academic Press Inc., 1996.

[Yung96] N. H.Yung, A.H. Lai and K.M. Poon, "Modified CPI filter algorithm for removing salt-and-pepper noise in digital images," *SPIE Proc. Series*, vol. 2727, pp. 1439-1448, 1996.

[Zimmer 80] H. J. Zimmermann and U. Thole, "On the suitability of minimum and product operators for the intersection of Fuzzy sets," *Fuzzy Sets and Systems*, vol. 2, pp. 173-186, 1980.