MULTIRESOLUTION DENOISING FOR ARBITRARILY SPACED DATA CONTAMINATED WITH ARBITRARY NOISE

BY

MEHDI SHAHBAZIAN

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

January 2005
To Jila, Saied and Sina
Acknowledgment

I would like to express my highest appreciation to my supervisor Dr. Farhad Ali Farhadpour. I am deeply indebted to him, for his supervision with keen interest, invaluable advice, support and encouragement throughout this research work. I would also like to express my gratitude to my supervisors Professor Ugur Tüzün, Dr. Hooshang Jazayeri-rad and Professor Mohammad Jami-al-ahmadi, for their guidance, encouragement, cooperation and support. Thanks are also extended to Professor Hans Müller-Steinhagen, ex-head of school of Engineering for his support during the course of this study.

I would like to thank the National Iranian Oil Company and the Petroleum University of Technology for providing me with the financial support. I acknowledge with thanks the generous support from the School of Engineering of the University of Surrey and the Research Centre of the Petroleum University of Technology throughout this research project.

I am especially grateful to my mother and late father for their emotional support and continued encouragement over the years that helped me going to the end. And last but not least, my special thanks and gratitude to my wife Mrs Jila Motashafie and my sons Saeid and Sina for their patience, encouragement and unfailing support during this period.
Nomenclature

A Design matrix
A_{i,j} (i,j)^{th} element of matrix A
\hat{a} Vector of parameters
\hat{a} Vector of estimated parameters
B(j, k) A block at dilation j and translation k
B(j, k, l) A two dimensional block at dilation j and translations k and l
BR(j, k) Balance ratio in block B(j, k)
C(\hat{a}) Covariance matrix of estimated parameters
C(j, k) Column of blocks associated with block B(j, k)
C(j, k, l) Two dimensional column of blocks associated with block B(j, k, l)
CS(\chi^2, \nu) Chi square distribution with \nu degrees of freedom
c Scaling coefficients vector (vector of weights in chapter 2)
c_{j,k} Scaling function coefficient at dilation j and translation k
c_j Vector of scaling coefficients at resolution j
\tilde{c}_{j+1} Even indexed samples (or coarse set) at resolution j
center_{j,k} Center of block B(j, k)
center_{j,k,l} Center of two dimensional block B(j, k, l)
d_{j,k} Wavelet coefficient at dilation j and translation k
d Vector of all wavelet coefficients at all resolutions
d_j Vector of wavelet coefficients at resolution j
d_{j,k} Selected wavelet coefficient at dilation j and translation k
\tilde{d}_{j+1} Odd indexed samples (or detail set) at resolution j
d True wavelet coefficients vector
d_{j,k} True wavelet coefficient at dilation j and translation k
E(.) Expected value
e_N The N-term approximation error
F(x, \hat{a}) Approximating function
F(x, \hat{a}^*) Best approximating function
F(x, \hat{\hat{a}}) Estimated function
F_j(x) Approximated function at resolution j
F(o) The Fourier transform of f(x)
f Vector of samples of the true function
f(x) The true function
f_j(x) Detail function at resolution j
f^*(x) Best estimate of f(x)
\( \hat{f}(x) \)  
Estimation of \( f(x) \)

\( f(x_1, x_2) \)  
A two dimensional true function

\( G_j \)  
Design matrix for scaling functions at resolution \( J \)

\( \mathbf{G}_j \)  
Matrix of the band pass filter at resolution \( j \)

\( g(n) \)  
Wavelet filter coefficients

\( H_j \)  
Design matrix for wavelets at resolution \( j \)

\( \mathbf{H}_j \)  
Matrix of the low pass filter at resolution \( j \)

\( h(n) \)  
Scaling filter coefficients

\( \text{interp}(\cdot) \)  
One dimensional data interpolation function

\( \text{interp2D}(\cdot) \)  
Two dimensional data interpolation function

\( J \)  
The coarsest resolution

\( j_{\max} \)  
Maximum resolution supported by data

\( L \)  
The finest resolution

\( \mathcal{L}^p \)  
Norm of degree \( p \)

\( \mathcal{L}^p(R) \)  
Space of real square integrable functions

\( \mathcal{L}^p(R^2) \)  
Space of two dimensional real square integrable functions

\( \ell(\cdot) \)  
Loss function

\( M \)  
Number of data points

\( M^u \)  
Number of projected regular data

\( M_n(j, k) \)  
Number of points above the estimate in block \( B(j, k) \)

\( M_{\min}(j, k) \)  
Minimum number of points on one side of the estimate in block \( B(j, k) \)

\( M(j, k) \)  
Number of data points in block \( B(j, k) \)

\( M_L(j, k) \)  
Number of data within block \( B(j, k) \)

\( \text{MABR}(j) \)  
Minimum acceptable balance ratio at resolution \( j \)

\( \text{MABR}(j, k) \)  
Minimum acceptable balance ratio at block \( B(j, k) \)

\( \text{MBR}(j) \)  
Minimum balance ratio at resolution \( j \)

\( M_{\min}^q(j, k) \)  
Minimum acceptable number of points on one side of the estimate in block \( B(j, k) \) computed from \( q_{low} \)

\( \varphi_{j,k}^q(x) \)  
The \( q^\text{th} \) order moment of the scaling function \( \varphi_{j,k}(x) \)

\( m_{j,k} \)  
Median of data within block \( B(j, k) \)

\( \hat{m}_{j,k} \)  
Predicted value of the median within block \( B(j, k) \)

\( N \)  
Number of basis functions

\( N_j \)  
Number of basis functions at resolution \( j \)

\( N_L(j, k) \)  
Effective number of basis functions within block \( B(j, k) \)

\( P(x) \)  
Probability distribution

\( P(x, y) \)  
Joint probability distribution

\( P(y|x) \)  
Conditional probability distribution

\( P_{j,k} \)  
Discrete power of wavelet \( \psi_{j,k}(x) \)

\( P(x) \)  
Cumulative probability distribution

\( P_{M,r} \)  
The value of binomial distribution for \( r \) successes in \( M \) trial

\( Q(a,x) \)  
Complement of the incomplete Gamma function
$q_{low}$ Limit of probability $q$
$q_{low}(j)$ The level dependent probability limit $q_{low}$ at resolution $j$
$q_{L}(j, k)$ Local probability within block $B(j, k)$
$R^a$ Interpolation matrix
$R$ Risk functional
$R(a)$ Predicted risk functional
$R_{emp}(a)$ Empirical risk functional
$\hat{r}_j$ Residual values at resolution $j$
$\hat{r}_i$ The $i^{th}$ residual
$S$ Function space
$S_k$ The $k^{th}$ subspace of a structured function space $S$
$SD_j$ Decomposition set at resolution $j$ in the IBMD algorithm
$SR_j$ Reconstruction set at resolution $j$ in the IBMD algorithm
$\text{select}(j, k)$ An indicator for selection of the coefficient corresponding to block $B(j, k)$
$U$ $M \times M$ orthonormal matrix in SVD
$V$ $N \times N$ orthonormal matrix in SVD
$\nu_j$ Subspace of approximating functions at resolution $j$
$\text{var}(.)$ Variance
$W$ $M \times N$ diagonal matrix of singular values in SVD
$W^\mu$ Orthogonal matrix associated with projected regular data
$W$ Orthogonal matrix associated a particular set of wavelets
$w$ Vector of nonlinear parameters
$w_j$ The $j^{th}$ singular value in SVD
$w_j$ Subspace of detail functions at resolution $j$
$x_{j, k}$ Abscissa of data within block $B(j, k)$
$Y$ A random variable
$\hat{Y}_j$ Vector the measured values
$\hat{Y}_{j, k}$ Estimated values at resolution $j$
$y_{j, k}$ Ordinate of data within block $B(j, k)$
$\hat{y}_{j, k}$ Vector of estimates within block $B(j, k)$ at resolution $j$
$y_{med}$ True median
$y_{sm}$ Sample median
$z_{j, k}$ Additive noise of the wavelet coefficients $d_{j, k}$
$(x_j, y_j)$ The $j^{th}$ data point
$\# \{ \}$ Cardinality of a set (number of members of a set)
**Greek's**

- $\alpha$: Probability of success in the binomial distribution
- $\beta$: Probability of failure in the binomial distribution
- $\Gamma(.)$: The Incomplete Gamma function
- $\chi^2$: The Chi-square value
- $\chi^2_{\text{min}}$: The Minimum chi-square
- $\chi^2_{L}(j, k)$: The Local chi-square within block $B(j, k)$
- $\delta_{kl}(x)$: The unit Kronecker delta function
- $\delta_i(a)$: The $i^{\text{th}}$ weighted residual
- $\delta$: The Dirac delta function
- $\epsilon_i$: A noise sample
- $\phi(x)$: A Scaling function
- $\phi_{j,k}(x)$: The $j^{\text{th}}$ dilation and $k^{\text{th}}$ translation of $\phi(x)$
- $\phi(x_1, x_2)$: A two dimensional scaling function
- $\psi(x)$: The mother wavelet
- $\psi_{j,k}(x)$: The $j^{\text{th}}$ dilation and $k^{\text{th}}$ translation of $\psi(x)$
- $\psi(x_1, x_2)$: A two dimensional wavelet
- $\eta^H$: Hard thresholding function
- $\eta^S$: Soft thresholding function
- $\nu$: Degrees of freedom in the chi-square distribution
- $\nu_{L}(j, k)$: Local degrees of freedom within block $B(j, k)$
- $\theta(x, w)$: General basis function
- $\rho(.)$: Probability density function
- $\Sigma^c_j$: Covariance matrix of scaling coefficients at resolution $j$
- $\Sigma^d_j$: Covariance matrix of wavelet coefficients at resolution $j$
- $\Sigma_y$: Covariance matrix of original data
- $\Sigma^{n}$: Covariance matrix of the regular projected data
- $\Sigma^{d}$: Covariance matrix of all wavelet coefficients
- $\sigma$: Standard deviation of noise
- $\hat{\sigma}$: Estimated standard deviation of noise
- $\sigma_{j,k}$: Standard deviation of noise for data within block $B(j, k)$
- $\tau$: Global threshold value
- $\tau_j$: Level dependent threshold value at resolution $j$
- $\tau_{j,k}$: Coefficient dependent threshold value at resolution $j$ and dilation $k$
**Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BR</td>
<td>Balance Ratio</td>
</tr>
<tr>
<td>DCT</td>
<td>The Discrete Cosine Transform</td>
</tr>
<tr>
<td>DWT</td>
<td>The Discrete Wavelet Transform</td>
</tr>
<tr>
<td>FDE</td>
<td>First Difference Error</td>
</tr>
<tr>
<td>FT</td>
<td>The Fourier Transform</td>
</tr>
<tr>
<td>FWT</td>
<td>The Fast Wavelet Transform</td>
</tr>
<tr>
<td>IBMD</td>
<td>Interpolated Block Median Decomposition</td>
</tr>
<tr>
<td>i.i.d.</td>
<td>Independent and identically distributed</td>
</tr>
<tr>
<td>LBF</td>
<td>Local Balance of Fit</td>
</tr>
<tr>
<td>LGF</td>
<td>Local Goodness of Fit</td>
</tr>
<tr>
<td>MABR</td>
<td>Minimum Acceptable Balance Ratio</td>
</tr>
<tr>
<td>MBR</td>
<td>Minimum Balance Ratio</td>
</tr>
<tr>
<td>MIPT</td>
<td>Median-Interpolating Pyramidal Transform</td>
</tr>
<tr>
<td>MRA</td>
<td>Multiresolution Analysis</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>SDE</td>
<td>Second Difference Error</td>
</tr>
<tr>
<td>SSE</td>
<td>Sum of Squared Error</td>
</tr>
<tr>
<td>STFT</td>
<td>The Short Time Fourier Transform</td>
</tr>
<tr>
<td>SURE</td>
<td>Stein Unbiased Risk Estimate</td>
</tr>
<tr>
<td>SVD</td>
<td>Singular Value Decomposition</td>
</tr>
<tr>
<td>TMSE</td>
<td>True Mean Square Error</td>
</tr>
<tr>
<td>TMSFDE</td>
<td>True Mean Square First Difference Error</td>
</tr>
<tr>
<td>TMSSDE</td>
<td>True Mean Square Second Difference Error</td>
</tr>
<tr>
<td>WT</td>
<td>The Wavelet Transform</td>
</tr>
<tr>
<td>1-D</td>
<td>One Dimensional</td>
</tr>
<tr>
<td>2-D</td>
<td>Two Dimensional</td>
</tr>
</tbody>
</table>
## Contents

Acknowledgement ii
Nomenclature iii
Contents viii
Abstract xii

### Chapter 1 Introduction
1.1 Denoising 1
1.2 General Framework for Function Estimation 5
1.3 Wavelets and Multiresolution Analysis 7
1.4 Multiresolution Analysis of Irregularly Spaced Data 11
1.5 Multiresolution Denoising of Irregularly Spaced Data 13

### Chapter 2 Function Estimation
2.1 Function Estimation: General Problem Statement 18
2.2 Function Spaces 22
2.2.1 Basis Functions 25
2.2.2 Structure in Function Spaces 28
2.3 Loss Functions: Maximum Likelihood Estimation 30
2.4 Empirical Risk Minimization: Estimating the Parameters 34
2.4.1 Singular Value Decomposition: Solving the Normal Equations 36
2.5 Complexity Control 38
2.5.1 Complexity Control using the Goodness of Fit 38
Chapter 3 Multiresolution Analysis and Wavelets

3.1 Multiresolution Analysis
   3.1.1 General Idea of Multiresolution Analysis
   3.1.2 Mallat's Multiresolution Analysis
   3.1.3 Scaling Functions and Wavelets

3.2 The Wavelet Transform
   3.2.1 Time-Frequency Analysis
   3.2.2 The Fast Wavelet Transform (FWT)

3.3 Important Properties of Wavelet Systems

3.4 The Multidimensional Wavelet Transform

3.5 Wavelets and Irregular Data

Chapter 4 Multiresolution Function Approximation for Irregularly Sampled Data

4.1 Introduction
   4.1.1 Indirect Methods
   4.1.2 Direct Methods

4.2 The Linear Projection Method

4.3 The Lifting Scheme
   4.3.1 Second Generation Wavelets

4.4 The Least Squares Wavelet Decomposition
   4.4.1 Selection of Basis Functions
   4.4.2 Computing the Wavelet Coefficients: Formulating the Least Squares Decomposition

4.5 Results and Discussion
   4.5.1 The Level by Level (Sequential) Least Squares Wavelet Decomposition
   4.5.2 The Simultaneous Least Squares Wavelet Decomposition
   4.5.3 Influence of Basis Selection Procedure on the Least Squares Reconstruction

4.6 The Multidimensional Least Squares Wavelet Decomposition

4.7 Conclusions
Chapter 5 Denoising Based on Local Goodness of Fit (LGF):
A Method for Irregular Data with Gaussian Noise

5.1 Introduction 113
5.2 Wavelet Domain Denoising of Regular Data: A Review 115
  5.2.1 Denoising of Regular Data by Wavelet Thresholding 120
  5.2.2 Global Thresholding 122
  5.2.3 Level Dependent Thresholding 125
  5.2.4 Coefficient Dependent Thresholding 129
5.3 Wavelet Domain Denoising of Irregular Data 131
  5.3.1 Projection and Wavelet Thresholding 132
  5.3.2 Least Squares Decomposition and Wavelet Thresholding 135
5.4 Data Domain Denoising: A New Method Based on
  the Local Goodness of Fit (LGF) 137
  5.4.1 The Local Goodness of Fit (LGF) 138
  5.4.2 The LGF algorithm 140
5.5 Results and Discussion 143
  5.5.1 Selection of the Probability Limit $q_{low}$ 143
  5.5.2 The Comparison of the LGF Method with Conventional Wavelet
        Thresholding 149
  5.5.3 Denoising of Irregular Data: the LGF Algorithm with the Least Squares
        Wavelet Decomposition 156
  5.5.4 Extension to Higher Dimensions 164
5.6 Conclusions 171

Chapter 6 Multiresolution Denoising with non-Gaussian Noise:
Local Balance of Fit (LBF) Technique 174

6.1 Introduction 174
6.2 Failure of Conventional Decompositions in the Presence of Outliers 176
6.3 Robust Estimation: Median versus Mean Estimation 179
6.4 Review of Robust Multiresolution Estimation Methods 184
6.5 Interpolated Block Median Decomposition (IBMD) 188
6.6 Data Domain Denoising: A New Method Based on the Local Balance of Fit (LBF) 193
   6.6.1 The Local Balance Criterion 195
   6.6.2 Selecting the Minimum Acceptable Balance Ratio (MABR) 199
   6.6.3 The LBF Algorithm 205
   6.6.4 Selection of the Probability Limit \( g_{low} \) 209
6.7 Comparison of the IBMD-LBF Denoising Method with other Techniques 217
   6.7.1 Regularly Spaced Data Contaminated with Gaussian Noise 217
   6.7.2 Irregular Data Contaminated with Gaussian Noise 219
   6.7.3 Regular Data Contaminated with Long Tail Noise 221
   6.7.4 Robustness against Outlier Patches 224
   6.7.5 Arbitrarily Spaced Data Contaminated with Arbitrary Noise 225
   6.7.6 Extension of the IBMD-LBF Denoising Method to Higher Dimensions 228
6.8 Conclusions 234

Chapter 7 Conclusions and Suggestions for Future Work 236
7.1 Conclusions 236
7.2 Suggestions for Future Work 244

Appendix 247

References 249
Abstract

Denoising is an essential ingredient of any data processing task because real data are usually contaminated by some amount of uncertainty, error or noise. The ultimate objective in this study is to handle the multiresolution denoising of an arbitrarily spaced multidimensional data set contaminated with arbitrary noise.

Denoising is closely related to function estimation from noisy samples, which is best achieved by complexity control in a structured function space. Multiresolution analysis and wavelets provide a suitable structured space for function estimation. However, conventional wavelet decompositions, such as the fast wavelet transform, are designed for regularly spaced data. Furthermore, the projection and lifting scheme approaches for dealing with irregular data cannot be easily extended to higher dimensions and their application to denoising is not straightforward. In contrast, the least squares wavelet decomposition offers a method for direct decomposition and denoising of multidimensional irregularly spaced data. We show that the frequently applied level by level multiresolution least squares wavelet decomposition suffers from gross interpolation error in the case of irregularly spaced data. The simultaneous least squares wavelet decomposition, with careful wavelet selection, is proposed to overcome this problem.

Conventional wavelet domain denoising techniques, such as global and level dependent thresholding, work well for regularly spaced data but more sophisticated coefficient dependent thresholding is required for irregularly spaced data. We propose a new data domain denoising method for Gaussian noise, referred to as the Local Goodness of Fit (LGF) algorithm, which is based on the local application of the conventional goodness of fit measure in a multiresolution structure. We show that the combination of the simultaneous least squares wavelet decomposition and the LGF denoising algorithm is superior to the projection and coefficient dependent thresholding and can handle arbitrarily spaced multidimensional data contaminated with independent, but not necessarily identically distributed, Gaussian noise.

For denoising of data contaminated with outliers and/or non-Gaussian long tail noise, the decomposition methods based on mean estimation are not robust. We develop a new robust multiresolution decomposition, based on median estimation in a dyadic multiresolution structure, referred to as the Interpolated Block Median Decomposition (IBMD). The IBMD method overcomes the limitations of existing median preserving transforms and can handle multidimensional irregularly spaced data of arbitrary size.

Thresholding methods for the coefficients of robust median preserving decompositions are currently limited to regular data contaminated with noise drawn independently and identically from a known symmetric distribution. To overcome these serious limitations, we develop a fundamentally new data domain robust multiresolution denoising procedure, called the Local Balance of Fit (LBF) algorithm, which is based on local balancing of the data points above and below the denoised function in a dyadic multiresolution structure. The LBF algorithm, which was inspired by the intuitive denoising style carried out by a human operator, is a distribution free method that can handle any arbitrary noise without a priori knowledge or estimation of the noise distribution. The combination of the robust IBMD decomposition and the LBF denoising algorithm can effectively handle a wide spectrum of denoising applications involving multidimensional arbitrarily spaced data contaminated with arbitrary and unknown noise. The only limitation is that the noise samples must be independent or uncorrelated.
Chapter 1

Introduction

"Few subjects have attracted as much attention in engineering and mathematics as wavelets have in recent years."
Jelena Kovacevic and Ingrid Daubechies

1.1 Denoising

Noise reduction is an essential part of any data processing task because real data are usually contaminated by some amount of uncertainty, error or noise. It is customary to assume that the noise is additive and any measured value is the sum of a true value and an added random noise value. Since both the true value and the noise value are unknown, appropriate statistical methods are required to remove the noise from the measured data for finding the true values. Denoising is closely related to function estimation from noisy samples, which is a main task in signal processing, image processing, modelling, identification and control. It is also addressed in nonlinear regression in statistics and in the learning theories of neural networks. In order to estimate the unknown true underlying function from noisy samples, it is necessary to remove or reduce the noise effects. In general, noise has a random nature and high frequency contents and causes the estimated function to be non-smooth and oscillatory. Consequently, noise reduction leads to smoothness and vice versa. Estimating the true function from noisy samples, regardless of the method of estimation, involve a trade-off between two important measures:

1- Agreement between the available data and the estimated function (sharpness).
2- Smoothness or stability of the estimated function
In the statistical literature, the compromise between sharpness and smoothness is referred to as the \textit{trade-off between bias and variance} of the estimation. A sharp estimator has a small bias but a large variance, while a smooth estimator gives a large bias but a small variance.

The development of denoising techniques has been pursued in diverse fields, using similar procedures embodied in different languages and notations. The problem of constructing an approximation of an unknown function from its empirical samples is a central problem in statistics known as \textit{function estimation} or \textit{regression}. Denoising is an essential part of any function estimation procedure dealing with noisy empirical samples. Signals and images are stored and processed in discrete form and a basic task is the reconstruction of a signal from noisy discrete samples (Vaseghi, 1996). Signal reconstruction and denoising have a close relationship with function estimation for one and two dimensional data sets. Another important subject in science and engineering is system modelling from empirical observations. Modelling can be considered as the process of finding mathematical correlations from examples, which obeys the same principles and faces the same difficulties encountered in a function estimation problem. It has also been demonstrated (Poggio and Girosi, 1989) that there is an intimate relationship between feed-forward neural networks and multidimensional function approximation. In particular, the development of the learning procedures in neural networks can be represented through functional relationships that involve the solution of a regression problem. The merger between the statistical, artificial intelligence, and signal processing points of view of the denoising problem has resulted in significant progress in all disciplines. Statistical function estimation techniques have been enriched with the addition of new learning techniques, and learning methods have benefited from a sound theoretical framework in statistics. The rich literature, strong theory and vast applications of signal and image processing have strongly supported (and have been strongly supported by) the other view points of denoising.

In this study we focus on denoising as a function estimation problem, which embodies almost all of the other specific view points. A very general formulation of
empirical function estimation from discrete noisy samples is via the so-called statistical learning theory. This theory, which originated in the late 1960's (see Vapnik and Chervonenkis 1968) and was further developed in the 1980's (Vapnik 1982, 1995, 1998, 1999) is usually referred to as the Vapnik-Chervonenkis or VC theory. The VC theory emphasizes function estimation in a structured space and provides a general mathematical framework for non-parametric (distribution-free) function (dependency) estimation from finite empirical data. This formulation of the learning problem is rather general, and incorporates the differing viewpoints adopted in pattern recognition, regression estimation, and density estimation. In particular, multiresolution analysis using wavelets, which forms the backbone of many of the effective modern denoising algorithms, falls naturally within the framework of the VC theory.

In general, noise can be considered as an unknown random variable drawn independently from a particular probability density function. Separating the noise from the data requires a priori knowledge or an assumption about the form of the noise distribution and its descriptive statistics, for example its mean and variance. For an a priori known noise distribution, it is possible to develop specific procedures that can provide the maximum likelihood that the data at hand could have occurred from a given model of the data. A clear and concise introduction to maximum likelihood estimation is given by Press et al, 1992. According to the central limit theorem of statistics, the sum of a very large number of very small independent random deviations almost always converges to a normal (or Gaussian) distribution. It has therefore become customary to assume that the noise is drawn independently and identically from a zero mean normal distribution. The variance of the assumed normal noise distribution is then either given a priori or may be estimated from the data using some suitable statistical method. In the case of normal noise, the maximum likelihood estimation procedure reduces to minimizing the familiar sum of squared errors. The least squares estimate preserves the mean of the data and is therefore referred to as a mean estimator. Many of the existing function estimation methods are mean estimators, which simply compute the estimate of the true value at each point as the average (mean) of its adjacent data values.
The Gaussian distribution of errors is often poorly realized in practice. The normal distribution decays sharply (as \(e^{-\left(y - y_{\text{mean}}\right)^2/\sigma^2}\)) when a particular measured value \(y_i\) deviates from the mean of the data \(y_{\text{mean}}\). This means that a particular data point deviating by \(\pm 10\sigma\) from the mean value is expected to occur only once in every \(5.1 \times 10^{21}\) measurements. Off points or "outliers" occur much more frequently in practical measurements due to non-predictable faults, for example power flickers, sensor failures and operator errors. The compact Gaussian distribution predicts that "tail" events or "off" points are much less likely than they actually are. For instance, radar glint data, astronomical data, infrared imaging and radiation measurement of weather balloons, often contain large noisy spikes and glitches and the associated noise distribution is strongly non-Gaussian.

Outliers cause serious problems in data analysis applications, for example function estimation, parameter estimation, model identification and time series analysis, if the noise distribution is taken as Gaussian and a mean estimator is employed. This is because outliers have a significant effect on the mean of the data and mean estimators are severely distorted in the presence of outliers. In particular, the majority of existing denoising techniques rely on mean estimators and perform poorly in the presence of outliers. To deal with outliers it is necessary to adopt a distribution for noise that has a longer tail than the compact normal distribution, the "double sided exponential" and "Cauchy" distributions are often used. The maximum likelihood estimators for such distributions are "robust" against outliers and do not attempt to preserve the mean. For example, in the case of the double sided exponential distribution the sum of absolute deviations, rather than the sum of squared deviations, is minimized. This corresponds to preserving the median rather the mean of the data and yields a median estimator which is robust against outliers. In other cases, the noise distribution is inherently non-Gaussian. For example, the noise associated with measurements based on counting exhibits a Poisson rather than normal distribution. The Poisson distribution tends to a normal distribution for large count numbers but has a broader tail than the normal distribution for small count numbers. The development of robust denoising techniques capable of dealing directly with outliers and long-tail noise is of particular interest in this study.
A large majority of the reported denoising algorithms, particularly those based on multiresolution wavelet analyses, are limited to uniformly spaced data contaminated with Gaussian noise. Practical measurements often suffer from data dropout, sampling jitter, multi-rate sampling or may be arbitrarily spaced. Another important objective of this study is the development of a denoising technique capable of direct handling of arbitrarily spaced multidimensional data sets. The scope of this study is therefore much wider than previous works on denoising. The ultimate aim is an effective denoising technique for an arbitrarily spaced multidimensional data set contaminated with noise drawn from an arbitrary and unknown distribution. Such a procedure would find application in a wide class of practical denoising applications.

1.2 General Framework for Function Estimation

In this study we shall view the denoising problem within the general framework of a function estimation problem. The goal of function estimation is to predict the value of the unknown underlying function in novel situations and the main requirement is the accuracy of the estimated function. It is helpful, therefore, to set the ground on which the accuracy of different methods can be analyzed and compared and discuss the inherent characteristics and limitations of the general function estimation problem. We shall briefly point to some of the issues involved, such as extrapolation, model validation, the ill-posed nature of the problem, and finally smoothness of the solution (Koulouris, A. 1995). A more detailed view of the function estimation problem in a structured space will be presented in Chapter 2.

The available data are always finite and limited to a bounded region of the input space. In non-parametric function estimation, the range of validity of the estimated function is limited to this bounded region. In other words, extrapolation of any approximating function beyond the bounded region is meaningless; it is equivalent to postulating a hypothesis that can not be supported by evidence. Another intrinsic characteristic of empirical function estimation is its inductiveness, finding the estimated function is actually equivalent to generalizing discrete and usually uncertain information to a continuous function. In fact, there are no rigorous mathematical proofs for the validity of any model constructed by any method. In other
words, no solution can be guaranteed and there is no way to exactly determine the
degree of accuracy of a developed model. *Testing* the accuracy of a model using
additional unseen data can not *prove* a model’s reliability; it can only *disprove* it if the
prediction accuracy is not satisfactory. This is the general curse of statistics, it can
never prove things, only disprove them! Mathematical justification of an estimation
mechanism can at best be provided as a proof of its *asymptotic convergence*
properties. This means that as the number of data points grows to infinity, the derived
model converges either to the true but unknown function or to an approximation with
known accuracy.

The fact that function estimation from discrete samples is an inherently
difficult and *ill-posed* problem is another complication (Poggio and Girosi, 1989).
This is essentially due to a general lack of knowledge about the true underlying
function and the finiteness of the available data. For every given set of data points,
there exist an infinite number of functions which can approximate the data arbitrarily
well but may be very different from the true function. Regardless of the method used
for data regression, all these approximating functions are equally acceptable solutions
to the function estimation problem. The question that naturally arises is whether there
exists some criterion based on which potential solutions can be screened out.
Certainly, the requirement for accuracy of data approximation alone is not a sufficient
criterion for selecting a *unique solution* amongst all potential solutions to the problem.
Furthermore, closeness to the available data is not an appropriate basis for comparing
the performance of different regression methods. Screening of the potential solution
must be based on *a priori* information about the properties of the true function or the
noise contaminating the data.

Without *a priori* information, intuition takes over. Intuitively, *smooth*
approximating functions are more plausible solutions to the function estimation
problem. It is natural to choose the simplest smooth-looking function that
approximates well the available data as the “best” solution. This agrees with the well-
known Occam’s razor philosophical principle (Kearns and Vazirani, 1994), which in
simple terms prefers the *shortest hypothesis* which can explain the observations. In
this context, shortness is equivalent to smoothness, which is a desirable property for
estimated function.
In the final analysis, any function estimation problem is reduced to a risk minimization problem. The best estimated function is the one that best resembles the probable set of data. In other words, ideally the predicted risk between the estimated function and all probable data should be minimized. In practice, however, the set of all probable data is not available and, therefore, the predicted risk is not accessible. Since only a finite set of noisy data is given, we can only compute and minimize the empirical risk which is based on the discrepancy between the estimated function and given data. For computing the empirical risk, it is necessary to select an appropriate function space for the estimated function by selecting a suitable set of basis function. In addition, an appropriate loss function must be defined to serve as a measure of the discrepancy between the available data and the estimated function. Empirical risk minimization alone cannot guarantee the minimization of the inaccessible prediction risk. It can, however, be coupled to a means for complexity control of the estimated function to provide an approximate minimization of the prediction risk. The use of a structured function space facilitates the solution of the function estimation problem and the minimization of the prediction risk. The selection of appropriately structured function spaces, the definition of suitable loss functions and methods for complexity control are considered in general terms in Chapter 2.

1.3 Wavelets and Multiresolution Analysis

In recent years, the theory and applications of wavelets has attracted considerable attention in various scientific disciplines and a substantial literature is now available on the subject. Many excellent references are now available on the fundamentals of the wavelet theory, for example Daubechies (1992), Meyer (1993), Chan (1995), Strang et al (1997), Burruse et al (1998), and Mallat (1998). Wavelets possess valuable mathematical properties which make them an ideal analysis tool in diverse areas; ideas from many different fields have been combined or merged to speed up the development of wavelet analyses (see Cohen and Kovacevic 1996). Applications exploiting the desirable properties of wavelets have been circulating in different areas such as sub band coding, filter bank analysis and pyramid algorithms.
in image processing, multigrid techniques in numerical analysis, the advanced theories in harmonic analysis, fractals, splines, and sampling theory in function approximation. Wavelets can be traced back to 1910, when Alfred Haar introduced a set of discontinuous basis functions for linear expansion of a signal (Haar, 1910). These basis functions, which are actually the simplest wavelets, are now referred to as Haar wavelets. The major advancement in wavelet theory was achieved in the 1980’s following the fundamental studies of pioneers such as Stromberg (1982), Morlet et al (1982), Morlet (1983), Grossman and Morlet (1984), Meyer (1985), Battle (1987), Lemarié (1988), and Daubechies (1988, 1990). A readable historical review of the development of wavelet theory from its different origins has been reported by Daubechies (1996). Our interest in wavelets stems from their ability to provide a very useful structured function space for function estimation and denoising. Here, we shall introduce the inherent suitability of wavelets for *time-frequency analysis* and their natural inclusion within a *multiresolution analysis* (MRA). A more detailed account of wavelets and multiresolution analysis will be given in Chapter3.

Let's start by recalling the well known Fourier analysis, where the basis functions are an infinite set of orthogonal sinusoids with infinite duration. Fourier expansion can identify and separate the frequency components of a signal. It cannot, however, specify the temporal location at which a particular frequency component exists. In other words, the classical Fourier expansion cannot reveal any information about a subinterval of a given signal. The Fourier expansion is, therefore, a powerful tool for the analysis of a stationary signal, whose power spectrum is constant over time. Characterization of a non-stationary signal, as well as distinguishing the local features of a signal, needs a combined *“time-frequency analysis”*.  

Conventionally, time-frequency analysis is performed via a windowed Fourier transform. This can be easily done by multiplying a moving window function (for example a rectangular or a Gaussian window) and the signal. This method is referred to as the *Short Time Fourier Transform* (STFT) and amounts to calculating the Fourier transform of a short time interval of a signal, over which the
power spectrum of the signal may be assumed constant. STFT maps a time function into a two dimensional space of frequency (ω) and time (t). The most severe drawback of STFT is the fixed time-frequency tiling employed. In practical transient signals, high frequency components occur over a shorter time span than low frequency components. This calls for the use of wide windows at low frequencies and progressively narrower windows at the higher frequencies, which is not offered by the fixed window size of STFT. The “Wavelet Transform” is an alternative to STFT with more desirable time-frequency resolution; it provides a powerful tool for the analysis of non-stationary or time varying signals (see Nielsen and Wickerhauser 1996).

In the wavelet transform, the basis functions are “Wavelets”. A wavelet ψ(x) is a small wave with its energy concentrated in both a time interval and a frequency range. In the time domain, wavelets either have compact support or decay rapidly toward zero. In the frequency domain, wavelets are band pass signals and have zero dc components. A complete set of wavelet basis functions can be simply generated via dilations (compression) and translations (shift) of a single wavelet function. The characteristic shape of a wavelet does not change by dilation but its time duration and frequency content change in opposite directions. Low frequency wavelet basis functions have a wide duration in time and a narrow band in the frequency domain, while high frequency wavelets have a short duration in time and a wide frequency band. In addition, wavelet basis functions can be designed such that they are orthogonal to each other. This makes the wavelet transform much better suited for time-frequency analysis of non-stationary signals than STFT. More significantly, wavelet systems are intrinsically connected to the notion of multiresolution analysis (MRA).

The notion of scale (resolution) plays an important role in the development of mathematical models in science and engineering. For example, a model of the earth must be considered at radically different scales in different applications such as solar system modelling, weather forecasting, surveying, and construction planning. The development of multiresolution (multiscale) techniques for data analysis and
modelling has attracted considerable interest in recent years. The basic aim in multiresolution analysis (MRA) is the ability to represent a function at different levels of detail, where important features of the function become more apparent. The lower resolution representation contains the global trend (low frequency contents) and the higher resolutions progressively capture the finer details (high frequency components) of the function. The general idea of MRA is similar to analyzing an object under a microscope, where by increasing the magnification more and more details of the object can be observed. MRA also appears to have a root in natural human behaviour, for example in human vision. In the process of memorizing or recognizing an image, the human eye-brain system performs MRA by focusing alternatively on the global and local features of the image. MRA has become an important and powerful tool in empirical function estimation and denoising and its characteristics will be covered in more detail in Chapter 3.

Within a multiresolutional structure, a function \( f(x) \) is expressed by its components at different scales (resolutions), that is:

\[
F_L(x) = F_J(x) + \sum_{j=J}^{L-1} F_j(x)
\]

where \( F_J(x) \) is the coarsest resolution (scale) representation and \( F_L(x) \) is the finest resolution representation of the function \( f(x) \). A coarse resolution represents the global trend (low frequency components) and a finer resolution contains local features (high frequency components) of the function. The basis functions used to construct the multiresolutional structure should have different measures of locality (effective support) at different resolutions. A basis function with a wider support can better approximate the global trend of a function at a coarse resolution while a narrow basis function is better suited for approximating the local features at a finer resolution. As we go up the resolution levels in a multiresolutional structure, the number of the basis functions increases but their effective support decreases. Wavelets provide a highly flexible set of basis functions that are both local and orthogonal and allow efficient multiresolution analysis.

The formal definition of MRA, which was first proposed by Mallat (1989) based on wavelet theory, will be discussed in Chapter 3. In particular, it will be
shown that, in the case of uniformly spaced data, the coefficients of the wavelet expansion (i.e. the wavelet transform) can be obtained through a highly efficient pyramidal algorithm. This so called Fast Wavelet Transform (FWT) requires O(M) operations (Burruse et al, 1998) where M is the number of data samples, it is even more efficient than the familiar Fast Fourier Transform (FFT) which requires O(MlogM) operations. The desirable mathematical properties of wavelets and the highly efficient FWT algorithm have led to an explosion in the application of MRA for function approximation, data compression and denoising. Such techniques are, however, only directly applicable to regularly spaced data sets.

1.4 Multiresolution Analysis of Irregularly Spaced Data

Recovery of a continuous function (or signal) from its discrete samples is a recurrent activity in practical applications. Conventional signal reconstruction methods are, however, largely based on uniform spacing of the samples. Real data are not always uniformly spaced and may suffer from different types of irregularities, such as data dropout, multirate sampling, sampling jitter or arbitrary spacing. The problem of signal reconstruction from non-uniformly spaced data arises in diverse areas of science and engineering such as signal and image processing, computer graphics, communication systems, geo-science and identification and control.

For uniformly sampled data, calculation of the coefficients in the multiresolution wavelet expansion can take advantage of the uniform grid and is straightforward. In particular, we can employ the highly efficient FWT algorithm. In the case of non-uniformly sampled data, the calculation of the wavelet expansion coefficients must be performed by a method capable of dealing with a non-uniform grid. Different approaches have been proposed for the recovery of a function from irregular samples, which include statistical methods, modified Fourier techniques, and basis fitting methods. In this study we focus on the basis fitting methods, which approximate the function as the linear sum of a set of basis functions. In particular, we consider the wavelet basis functions in a multiresolution structure. In the case of non-uniformly sampled data, the FWT algorithm cannot be used directly and the
estimation of the wavelet coefficients becomes more complex. Chapter 4 provides a detailed discussion of the techniques proposed for handling irregularly sampled data within a multiresolutional wavelet structure.

A variety of direct and indirect approaches have been suggested for handling irregularly spaced data within a multiresolution structure. Indirect methods involve the projection of the irregular data onto a regular grid using a suitable projection technique; the projected data can then take full advantage of the machinery of the efficient FWT algorithm. This procedure has been particularly well developed by Kovac and Silverman (2000) for one-dimensional irregularly spaced data sets. The projection methods, however, do not deal with the available data directly and their performance is inevitably sensitive to the type of projection employed. In addition, the extension of indirect methods to two and higher dimensional data sets is complicated by the complexities inherent in multidimensional projection of arbitrarily spaced scattered data. To our knowledge, the reported applications of indirect projection methods have been confined to one-dimensional data sets.

Two general schemes have been proposed for dealing directly with the available irregularly spaced data within a multiresolution structure. The \textit{irregular lifting scheme} developed by Wim Sweldens and his co-workers (Sweldens 1994, 1995-a, 1995-b, 1996, Sweldens and Schröder 1995) employs data adapted second generation wavelets. The lifting scheme is computationally highly efficient and can deliver a perfect reconstruction of arbitrarily spaced data and is fully discussed in Chapter 4. However, the second generation data adapted wavelets employed are no longer the translated and dilated versions of a single mother wavelet. This complicates the extension of the irregular lifting scheme to two and higher dimensions. In this study we favour another optimization (\textit{least squares}) based decomposition, which deals with the irregular data directly, employs regular wavelets that are the translations and dilations of a single mother wavelet and is easily extended to any dimension. The least squares wavelet decomposition has been used by a number of other authors (e.g. Bakshi and Stephanopoulos 1993, Safavi, 1995, Ford and Etter, 1998) but its characteristics have not been fully recognized. A full analysis of the least squares wavelet decomposition will be undertaken in Chapter 4. In particular, it will
be demonstrated that the currently used level by level least squares wavelet decomposition suffers from gross interpolation error in the case of strongly irregular data sets. Such errors can lead to very unreliable multiresolution denoising performance, a point that to our knowledge has not been previously addressed. An alternative, simultaneous least squares wavelet decomposition will be suggested in Chapter 4, which does not suffer from gross interpolation error and allows the development of reliable multiresolution denoising techniques.

1.5 Multiresolution Denoising of Irregularly Spaced Data

Wavelets provide a sparse representation of practical signals. That is, the signal energy is mainly concentrated in a small number of the wavelet coefficients and a large fraction of the wavelet coefficients are essentially zero. Many powerful and fast wavelet domain multiresolution denoising methods have been developed to exploit this property of the wavelet transform. A simple but powerful method for denoising in the wavelet domain is to shrink the small wavelet coefficients (smaller than a specified threshold) to zero before reconstruction. This wavelet shrinkage or wavelet thresholding approach was pioneered by Donoho and Johnstone (1992, 1994, 1995) and forms the backbone of many of the classical wavelet denoising methods now available. Selecting the threshold value is a fundamental issue in wavelet-based denoising methods and the various procedures reported will be critically reviewed and compared in Chapter 5.

The classical wavelet thresholding techniques have almost invariably been developed on the assumption that the data is uniformly spaced and contaminated with noise drawn independently and identically from a Gaussian distribution. It is easy to show that for such cases the wavelet coefficients are also independent and identically normally distributed, which paves the way for developing simple but effective thresholding procedures. As we shall clearly demonstrate in Chapter 5, classical thresholding techniques do not perform well when the noise is not Gaussian and cannot be used directly if the data is arbitrarily spaced. An attempt to extend the scope of wavelet thresholding for handling irregularly spaced noisy one-dimensional
data sets has been reported by Kovac and Silverman (2000). This method involves the projection of the irregular data onto a regular grid and denoising the projected data using a sophisticated coefficient dependent thresholding procedure to account for the induced correlation between the wavelet coefficients. Simple examples will be used in Chapter 5 to demonstrate the unreliable denoising performance of this procedure.

In this study, we develop an alternative approach for denoising of arbitrarily spaced data by performing the denoising in the original data domain rather than in the wavelet domain. In Chapter 5, we propose a new denoising method for irregular data contaminated with Gaussian noise based on a Local Goodness of Fit (LGF) criterion. The LGF denoising procedure is carried out entirely in the data domain but takes full advantage of the locality of the multiresolution basis functions in both space and frequency. The fact that the multiresolution basis functions have local support is sufficient to allow for complexity control by checking a local measure of the traditional goodness of fit. Furthermore, the LGF denoising procedure is readily combined with the simultaneous least squares wavelet decomposition method developed in Chapter 4 to form a computationally efficient algorithm. The combination of the LGF criterion and the least squares wavelet decomposition is readily extended to higher dimensions and is also capable of handling spatially variable Gaussian noise. Detailed comparison of the proposed LGF denoising algorithm and the available wavelet domain denoising methods will also be undertaken in Chapter 5 through a variety of illustrative examples and simulations.

The highly efficient classical wavelet thresholding techniques are all developed on the understanding that the noise is independently and identically drawn (i.i.d.) from a Gaussian distribution. The new data domain LGF denoising method presented in Chapter 5 retains the Gaussian assumption but can be used for non-identically distributed or spatially variable noise. In many practical measurements, however, the data is contaminated by off points or outliers caused by instrument failure or human error. Outliers cannot be handled using the compact Gaussian distribution and force the use of a distribution with a much wider tail. We start Chapter 6 with a few illustrative examples to clearly demonstrate the influence of outliers and long-tail noise and the serious failure of denoising procedures developed on the assumption of Gaussian noise. Wavelet decomposition is a linear
transformation of the data and the outliers exert an unbounded influence on the wavelet coefficients. Consequently, wavelet shrinkage denoising methods or the data domain LGF denoising technique presented in Chapter 5 do not perform adequately in the presence of outliers. Direct handling of outliers or data contaminated with non-Gaussian noise has critical importance in non-stationary applications like tracking, signal and image processing, navigation and fault detection. Our aim in Chapter 6 is to develop a data domain denoising method that is effectively distribution free and can handle arbitrarily spaced data contaminated with independent but arbitrarily distributed noise.

Multiresolution denoising is developed on the premise that the essential features of the signal are captured by the lower resolution coefficients and the noise is captured by the higher resolution coefficients. Both the classical discrete wavelet decomposition and the least squares wavelet decomposition are designed to preserve the local mean of the signal across the resolutions. Outliers have a strong influence on the mean and can therefore distort the lower resolution coefficients obtained through a mean-preserving decomposition. Consequently, the denoised signals recovered can become severely distorted in the presence of outliers. Much better denoising performance can be achieved through a multiresolution decomposition aimed at preserving the local median rather than the local mean of the signal. This is simply because the median is far less affected by outliers; a median preserving decomposition is “robust” in the sense that it limits the influence of outliers to the higher resolutions and prevents its leakage to the lower resolution.

Median preserving decompositions have been examined by a number of authors, in particular by Donoho and Yu (1997) for regularly-spaced data under fairly restrictive conditions. In Chapter 6 we develop a median preserving decomposition, named Interpolated Block Median Decomposition (IBMD), which removes such restrictions and can be used with irregularly spaced data. The robust IBMD decomposition opens the way for the development of a completely new robust multiresolution data domain denoising algorithm, named the Local Balance of Fit (LBF) algorithm, which is based on local balancing of the data points above and below the denoised function in a dyadic multiresolution structure. The LBF algorithm is inspired by the seemingly effortless yet effective way a human operator is able to
draw a smooth curve through a noisy data set. An individual drawing such a curve
does not care about data spacing, handles the outliers automatically, performs no
discernable calculations for noise estimation and needs no information about the noise
distribution.

A comprehensive comparison of the IBMD-LBF denoising procedure will be
carried out in Chapter 6 for several examples which can also be handled using other
denoising techniques. Such examples serve to confirm the robustness and excellent
denoising performance of the proposed IBMD-LBF denoising method. A number of
complex examples involving arbitrary spaced data contaminated with arbitrary noise
will also be presented that cannot be handled by other denoising methods but are dealt
with seamlessly and effectively by the proposed IBMD-LBF method. There is
nothing in either the IBMD algorithm or the LBF algorithm to prevent their extension
to multidimensional denoising applications. Illustrative two-dimensional denoising
examples will also be presented in Chapter 6 to confirm the effectiveness of the
proposed IBMD-LBF denoising method. The examples and simulations presented in
Chapter 6 serve to confirm the IBMD-LBF combination as a distribution free method
that can be confidently used for all types of denoising applications.

The only requirement of the data domain LGF denoising algorithm presented
in Chapter 5 is that the noise is independent and Gaussian. On the other hand, the
IBMD-LBF algorithm developed in Chapter 6 only requires that the noise is
independent and uncorrelated. Suggestions for future work, in particular the
extension of the data domain LGF and LBF denoising algorithms to correlated noise,
are given in Chapter 7. Finally, we wish to remark that all the calculations performed
in this thesis were performed using specific procedures developed by the author
within the MATLAB computing environment. Copies of the relevant programs and
procedures are available from the author on request. Some important samples of the
MATLAB codes are included in the enclosed CD.
Chapter 2

Function Estimation

This study concerns the estimation of a function from a limited set of noisy data. This introductory chapter provides a brief background on important concepts of function estimation. The function estimation problem has been investigated in different disciplines under different names but following the same spirit. For example, regression in statistics, denoising in signal processing, empirical modelling in science and engineering, and learning in Neural Networks (NN) can generally be considered in a function estimation framework.

The problem of constructing an approximation of an unknown function from its empirical samples is one of the main problems in statistics and is known as regression. Empirical data are usually subject to some amount of uncertainty or noise. To estimate the unknown true function from empirical data it is necessary to remove or reduce the effect of noise. In other words, denoising is embedded in any function estimation procedure dealing with noisy samples.

An essential subject in science and engineering is system modelling from empirical observations. Modelling can be considered as the process of finding mathematical correlations from examples. There are two model development approaches: parametric and non-parametric. In parametric modelling, the unknown function is constrained to a specific form based on the fundamental laws of physics and chemistry and the task is estimating the values of a fixed number of unknown but physically meaningful parameters. In contrast, non-parametric models implement the function structure without making strong assumptions about the form of the unknown function and involve a set of parameters (coefficients) with little or no physical meaning.
The non-parametric modeling problem obeys the same principles and encounters the same challenges faced in the function estimation problem. For example, there is a close relation between feed-forward neural networks and multidimensional function estimation. The learning in neural networks can be represented through functional relationships that involve the solution of a function estimation problem. Another way of looking at the problem of function estimation is via the so-called statistical learning theory, which is a general mathematical framework for non-parametric (distribution-free) function (dependency) estimation from finite empirical data. A useful product of this viewpoint is the unification of the diverse languages used in various disciplines.

In this chapter, the general mathematical description of function estimation is presented first and important elements in function estimation procedures including structured function spaces, loss functions, empirical risk minimization and complexity control are introduced in turn. The importance of selecting a proper function space, suitable basis functions, and the structure in function spaces are discussed in the second section. In the third section, the loss function and empirical risk are defined based on the maximum likelihood estimation principle for different noise distributions. This is followed by a fairly detailed look at empirical risk minimization for the case of Gaussian noise. This reduces to a least square problem with its robust solution obtained through singular value decomposition (SVD). Section 2.5 is devoted to a review of the available methods for complexity control (model selection) and the identification of the classical goodness of fit criterion as the method of choice for complexity control.

2.1 Function Estimation: General Problem Statement

Function estimation is the problem of estimating an unknown mapping from available limited input-output samples in order to predict the output for any input value. The main goal of estimation is prediction accuracy which is also referred to as generalization. We start the description of the function estimation problem from the general frame work of statistical learning theory introduced by Vapnik and Chervonenkis in the late 1960's (see Vapnik and Chervonenkis, 1968) and developed
further in the 1990's (see Vapnik, 1982, 1995, 1998, 1999). This theory, which is often referred to as the VC theory, is rather general and covers various estimation and learning problems such as classification, regression and density estimation.

The generic function estimation (learning) system is shown schematically in Figure 2.1 and consists of three main parts:

![Figure 2.1 Components of a generic function estimation system](image)

1) A generator of random input vectors $x$, which are drawn independently from a fixed but unknown probability distribution $P(x)$;

2) A system which returns an output value $y$ for every input vector $x$ according to a fixed but unknown conditional probability distribution $P(y|x)$;

3) An estimation mechanism, which is capable of estimating the unknown mapping between the inputs and the output of the system from a finite set of available data $(x_i, y_i), (i = 1, \cdots, M)$.

The solution to the estimation problem is a function of the form $F(x, a)$, where $a \in A$ and $A$ is a set of arbitrary parameters. This solution belongs to a set of parameterized approximating functions space $S$ and the purpose of estimation is to select a function $F(x, a^*)$ from the space $S$ that best approximates the system’s response. This selection is based on the knowledge of a finite number $M$ of available samples $(x_i, y_i), (i = 1, \cdots, M)$ generated according to an unknown joint probability distribution $P(x, y) = P(x)P(y|x)$. For a given input $x$, the quality of an
approximation produced by the estimation mechanism is measured by a discrepancy or
*loss function* \( \ell(y, F(x, \theta)) \) between the actual output generated by the system and
the estimate produced by the estimation mechanism. The choice of the loss function \( \ell \)
is an essential decision that characterizes different estimation problems such as
classification, regression and density estimation (See Vapnik, 1998).

The expected value of the loss function is called the *predicted (expected) risk
functional* and is defined as

\[
\mathcal{R}(\theta) = \int \ell(y - F(x, \theta)) \, dP(x, y)
\]  

(2.1)

Based on the above description, the general function estimation problem can be
mathematically stated as follows:

"Given a set of data points \((x_i, y_i) \in \mathbb{R}^{d \times 1}, (i = 1, \cdots, M)\) drawn from some
unknown joint probability distribution \(P(x, y)\), find a function
\(F(x, \theta^*)\) belonging to a class of functions \(S\) that minimizes the predicted risk
functional \(\mathcal{R}(\theta)\)."

Without loss of generality, we assume that the input space is restricted to the \(d\)-
dimensional hypercube \(I = [0, 1]^d\). The only restriction placed on the true function \(f(x)\)
in this study is that it is *bounded and square integrable* within the hypercube \(I\),

\[
|f(x)| < \infty
\]  

(2.2)

\[
\int |f(x)|^2 \, dx < \infty
\]  

(2.3)

Ideally, the space of approximating functions \(S\) should also possess such properties.

The specific problem addressed in this thesis is the estimation of an unknown
function from a set of noisy samples. This problem is known in statistics as
(nonlinear) regression and can be stated as follows:
Given a finite set of measured values \((x_i, y_i) \in \mathbb{R}^d, (i = 1, \ldots, M)\) of an unknown bounded and square integrable function \(f(x)\) corrupted by some random noise \(\varepsilon\) with an unknown probability distribution,

\[
y_i = f(x_i) + \varepsilon_i, \quad i = 1, \ldots, M
\]

(2.4)

determine an estimation of the true function \(f^*(x) = F(x, \theta^*)\) belonging to a class of functions \(S\) that minimize the predicted risk functional \(\mathfrak{R}(\theta)\).

Here, \(x\) is a \(d\)-dimensional vector and \(y\) is a scalar output. The random noise \(\varepsilon\) has zero mean and the data samples are usually supposed to be independent and identically distributed (i.i.d.) according to some (possibly unknown) joint probability density function \(P(x, y)\). An important contribution of this thesis is the development of a robust technique for estimation of the true function when the noisy samples are not identically distributed.

For any approximating function in space \(S\), the predicted risk \(\mathfrak{R}(\theta)\) is a measure of the approximation error with respect to the data over the entire input space \(I\). The function \(F(x, \theta^*)\) that minimizes \(\mathfrak{R}(\theta)\) is the closest function in \(S\) to the true function \(f(x)\) with respect to the loss metric \(\ell\) weighted by the probability \(P(x, y)\), see equation (2.1). This measure is usually taken as the \(L^p\) norm of the difference \(y - F(x, \theta)\):

\[
L^p[y-F(x, \theta)] = \left[\int |y-F(x, \theta)|^p \, dx\right]^{1/p}
\]

(2.5)

The prediction risk \(\mathfrak{R}(\theta)\) can not be directly quantified for two reasons:

1) the joint probability distribution \(P(x, y)\) is usually unknown

2) data is not available over the entire input space

The estimation procedure is therefore usually based on the average of the loss metric \(\ell\) over the available but limited data points,
\[ R_{\text{emp}}(\hat{\theta}) = \frac{1}{M} \sum_{k=1}^{M} \ell(y_k - F(x_k, \hat{\theta})) \]  

(2.6)

The directly quantifiable functional in equation 2.6 is referred to as the **empirical risk** and its minimization is termed **empirical risk minimization**. The problem is therefore reduced to finding an estimate of the true function \( F(x, \hat{\theta}) \) within the function space \( S \) by minimizing the computable empirical risk rather than the desired but inaccessible prediction risk.

It is clear that by increasing the size of the parameter vector \( \hat{\theta} \), we can reduce the empirical risk further and further. A large parameter vector \( \hat{\theta} \) will inevitably result in a complex approximating function \( F(x, \hat{\theta}) \). This may, however, be very different from the best approximating function \( F(x, \hat{\theta}^*) \) that minimizes the prediction rather than the empirical risk, especially in the presence of noise. **Complexity control** (i.e. reducing the size of the parameter vector \( \hat{\theta} \)) is an essential and important tool for avoiding such difficulties (over-fitting) and assuring that \( F(x, \hat{\theta}) \) is an acceptable approximation to \( F(x, \hat{\theta}^*) \). **Empirical risk minimization** and complexity control are the two essential ingredients of any function estimation procedure which involves:

1) Selection of the approximating functions space \( S \),

2) Definition of the loss function \( \ell \),

3) Development of a method for empirical risk minimization, and

4) Model selection based on a suitable complexity control mechanism.

The issues of space selection may be considered within a general framework and will be considered next. The definition of specific loss functions, methods for minimizing the empirical risk, and available methods for complexity control will be covered in sections 2.3, 2.4, and 2.5 respectively.

### 2.2 Function Spaces

The critical importance of choosing an appropriate function space is brought out by considering potential sources of error in a function estimation procedure. This
is perhaps best appreciated by resorting to the language of approximation theory, which deals with the problem of deriving (best) approximations of a given function in normed functional spaces $S$. By definition, a best approximation of $f(x)$ in $S$ is a function $f^*(x) = F(x, a^*)$ that has the minimum distance from $f(x)$, defined in terms of a chosen functional norm $\|\|$:

$$\|f - f^*\| \leq \|f - \tilde{f}\| \quad \text{for all } \tilde{f} \in S$$  (2.7)

It is evident that unless $S$ is properly selected to include the given function $f(x)$, the best approximation $f^*(x)$ in $S$ will be different from $f(x)$. This difference is referred to as the approximation error (Girosi and Anzellotti, 1993, Cheney, 1966 and Rice, 1964, 1969). Obviously, the estimation task becomes meaningless for an inappropriate choice of $S$ that causes a large approximation error. For a class of functions with given properties, for example boundedness and square integrability, the function space $S$ must be chosen such that it is possible to approximate any function (belonging to that class) with arbitrary accuracy. This feature, which is known as the universal approximation property, is the minimal requirement for a desirable function space $S$. For example, in Fourier analysis, any bounded and square-integrable function is exactly represented by an infinite number of trigonometric functions. For computations to be feasible, however, the space $S$ can only be characterized by a finite number of adjustable parameters (i.e. a finite set of basis functions). Consequently, another reason for the existence of approximation error is actually the finiteness of the basis functions in the function space $S$.

In the case of empirical function estimation, the true function is unknown and only a finite number of noisy samples are available. The finiteness and uncertainty of the data are two additional sources of error in function estimation. In general, the function estimated from finite noisy samples ($\hat{f}$) is different from the best approximation $f^*$, regardless of the estimation algorithm. This difference, $\|\hat{f} - f^*\|$, is referred to as the empirical error (Girosi and Anzellotti, 1993). The combination of the approximation and empirical errors is referred to as the generalization (or prediction) error, which measures the predictive accuracy of the estimated function. The inevitability of generalization error means that in practice some estimation error has to
be tolerated. The magnitude of this error depends in part on the selection of the function space $S$ and the procedure used to control the empirical error.

One of the critical decisions in the function estimation problem is the choice of the function space $S$. In nonparametric modelling, the space of approximating functions $S$ is usually defined as the linear combination of a finite number $N$ of basis functions $\theta(x, w)$ with a set of (possibly unknown) parameters $w$:

$$S = \left\{ F(x, a) | F(x, a) = \sum_{k=1}^{N} c_k \theta_k(x, w), \quad a = (c, w) \right\}$$

(2.8)

We note here that in the above equation the parameter vector $a$ in $F(x, a)$ is divided into separate vectors $c$ and $w$, $a = (c, w)$. The linear coefficients (or weights) are denoted by $c$, and $w$ are the (possibly nonlinear) parameters of the chosen basis function $\theta(x, w)$. The number of basis functions that construct $S$ is referred to as the dimension of $S$. This definition of the space $S$ covers a large class of approximating functions including those used in standard types of statistical regression (e.g. polynomial approximation) and many types of Neural Networks (e.g. Radial Basis Function Neural Networks).

In this study, we are concerned with a subspace of the above function space, where the basis functions are fixed a priori. The vector $w$ is then empty and the basis functions have no adjustable parameters so that $a = c$ and,

$$F(x, a) = \sum_{k=1}^{N} a_k \theta_k(x)$$

(2.9)

This class of approximating functions is linear with respect to the parameters $a$, which has distinct advantages in both empirical risk minimization and complexity control. For the above linear model, the choice of the function space, $S$, requires two related decisions:

1) the type of the basis functions, $\theta_k(x)$, and

2) the number of basis functions $N$ (and so the dimension of $S$).
2.2.1 Basis Functions

Linear approximating functions are used in many practically important estimation methods such as multiple linear regression, linear additive models, polynomial approximation, Fourier expansion and finally wavelet and multiresolution approximation. The subspace constructed by a set of multiresolution basis functions (for example wavelets) is of particular interest in this study and has the desired universal approximation property for bounded and square integrable functions.

a) Key Properties of Basis Functions

The key properties of basis functions are locality and orthogonality, which can influence the efficiency of the approximation. One of the most important properties of basis functions is orthogonality. A set of basis functions is called orthogonal if the inner product of any two members \( \theta_k(x) \) and \( \theta_l(x) \) is:

\[
\langle \theta_k(x), \theta_l(x) \rangle = \int \theta_k(x) \theta_l(x) dx = E \delta_{kl}(x)
\]

where \( \delta_{kl}(x) \) is the unit Kronecker delta function and \( E \neq 0 \). If the functions are normalized such that \( E=1 \), then the set is called an orthonormal set. We shall assume that an infinite set of orthonormal basis functions \( \{ \theta_k(x), k=1,2,\ldots,\infty \} \) exists and spans the continuous function space. It can then be shown that each coefficient \( a_k \) is uniquely obtained by the projection (the inner product) of \( f(x) \) on the basis function \( \theta_k(x) \):

\[
a_k = \langle f(x), \theta_k(x) \rangle = \int f(x) \theta_k(x) dx,
\]

The approximation error incurred on using only an \( N \)-term orthonormal expansion can be represented as:

\[
e_N^2 = \int [f(x) - \sum_{k=1}^{N} a_k \theta_k(x)]^2 dx = \sum_{k=N+1}^{\infty} a_k^2
\]

It follows that the improvement obtained by adding the \( (N+1) \)th term is measured directly by the coefficient \( a_{N+1} \):

25
$e^{2}_{N+1} - e^{2}_{N} = a^{2}_{N+1}$ \hspace{1cm} (2.13)

Consequently, for an orthonormal set the smallest expansion for a given accuracy is formed by selecting those basis functions corresponding to the largest coefficients.

There is a wide variety of orthogonal sets which can serve as basis functions for orthogonal expansion, for example, sinusoids and orthogonal polynomials. It should be noted, however, that such functions have 'global' characteristics and influence the approximation over the whole range of the independent variable. This means that all basis functions will contribute to the approximation at a given point. A local basis function on the other hand, either has compact support or falls rapidly to zero outside a limited range of the independent variable. Therefore, only some of the basis functions can affect the overall approximation at a given point. Polynomials and Gaussians are typical examples of global and local basis function shown in Figure 2.2.

![Global and local basis functions](image)

Figure 2.2 Global and local basis functions (a) Global (polynomial) (b) Local (Gaussian)

Clearly, a global basis function can better approximate global trends while a local basis function is better suited for approximating local features. It is of course possible to use a combination of global and local basis functions. Ideally, we require a set of basis functions which are simultaneously local and orthogonal and allow a measure of control on their locality. Wavelets provide a highly flexible set of basis functions which are both local and orthogonal and also allow efficient multiresolution
approximation. Figure 2.3 shows examples of global and local orthogonal basis functions.

Figure 2.3 Orthogonal basis functions (a) Sinusoidal (global) (b) Cubic B-Spline wavelets (local)

b) Number of Basis Functions

We note that the form of the chosen basis functions can affect the number of basis functions required to achieve a given accuracy. For example, a smooth function can be accurately represented by relatively few smooth basis functions whereas a non-smooth function requires a larger number of basis functions for approximation. Evidently, a priori knowledge on the unknown function, \( f(\mathbf{x}) \), may be used to guide the selection of suitable basis functions.

For selected basis functions and given data, the size of \( S \) determines the trade-off between the smoothness of the estimated function and the precision with which the available data is approximated. This is also referred to as the bias versus variance trade-off in statistics. By making \( S \) sufficiently large, the precision of data approximation is increased at the expense of the smoothness of the estimated function. In other words, the bias in the estimation is decreased while its variance is increased. For example, in the space of polynomial basis functions, a large degree polynomial can closely approximate (even interpolate) the data, but with a highly oscillating function. This problem is referred to as over-fitting, which usually occurs when an unnecessarily large number of basis functions are used for fitting the data. Based on
the above discussion, it can be concluded that for every particular set of basis functions and given data, an appropriate size of \( S \) should be determined for finding a desirable estimation. This is actually the issue of complexity control that will be discussed in section 2.5.

2.2.2 Structure in Function Spaces

Choosing a large size function space (large number of basis functions) may demand a large effort to search for the solution and can also lead to over-fitting problems. On the other hand, it is inappropriate to choose a small size function space because this may lead to a large approximation error regardless of any modelling effort. To avoid such difficulties, it is prudent to search for a solution in a hierarchy of increasingly larger function spaces. Vapnik (Vapnik, 1982, 1995, 1998, 1999) introduced the notion of structure for a function space to address this problem. The space \( S \) of approximating functions \( F(x, a) \) is assumed to consist of an infinite ladder of nested subspaces \( S_k \) such that

\[
S_1 \subset S_2 \subset \cdots \subset S_k \subset \cdots \quad \text{and} \quad \lim_{k \to \infty} S_k = S
\]  
(2.14)

Here, each element of the structure \( S_k \) has a finite dimension \( d_k \). A structure provides ordering of its elements according to their dimensions,

\[
d_1 \leq d_2 \leq \cdots \leq d_k \leq \cdots
\]  
(2.15)

The space of polynomials of increasing degree is an example of a hierarchical structure. Generally, for continuity of the nested subspaces, it is convenient that the basis functions that span the subspace \( S_{j+1} \) be a superset of basis functions forming the subspace \( S_j \). As the index \( j \) of \( S_j \) increases, the number of basis function increases and the subspaces become gradually more complex but can approximate a given data set with increased precision. Starting the search for the solution at the lowest index space causes a clear bias towards simpler (but not necessarily smoother) solutions. To bias the solution to smoother approximating functions, it is also necessary to impose some restriction on the form and smoothness of the basis functions in the hierarchical structure of subspaces. For example, spanning the lower index (dimension) subspaces with smoother basis functions causes a strong bias towards smoother solutions.
Once a set of basis functions of given type is chosen, the definition of the structure amounts to the ordering of the basis functions according to their likelihood of being significant for accurate function estimation. This ordering can proceed in two distinct ways (Donoho, et al, 1998; Cherkassky and Shao, 2001). In the simpler case, the ordering is defined independent of data and is based on the properties of the basis functions alone. For example in the case of the classical Fourier analysis, the sinusoidal basis functions may be ordered based on their frequencies. The subspace $S_k$ is then chosen as the first $k$ harmonic frequencies. This data independent approach is referred to as the linear approach for defining the structure. In the linear approach the nested structure is laid down a priori and we can then perform the parameter estimation and model selection tasks. The major disadvantage of this approach lies in the fact that valuable information which may be contained in the data is not utilized for defining the nested structure and the data is only used in the parameter estimation step.

In the second approach, the ordering of the basis functions is based on the given data. For example, the basis functions can be ordered according the amplitude of their coefficients in the linear expansion. In the Fourier analysis case, the coefficients of expansion can be first estimated based on the entire data set. The ordering of the sinusoidal basis functions is then performed based on the amplitude of the corresponding Fourier coefficients. The subspace $S_k$ then contains the sinusoids with the $k$ largest coefficients. This data dependent approach is referred to as the nonlinear approach for defining the structure.

Both linear and nonlinear orderings are widely used in the case of orthogonal basis functions. The linear scheme seems very natural for approximation of many practical functions using the classical Fourier basis or orthogonal polynomial basis, because the first $k$ terms are also essentially the most important (largest) $k$ terms. Figure 2.4 illustrates two different reconstructions using the first-100 and the largest-100 coefficients of the Discrete Cosine Transform (DCT) of a blocky function. The first-100 approximation is smoother but the largest-100 approximation has a smaller sum of squared error (SSE). As we shall see later, in the presence of noise, the nonlinear approach is significantly better than the linear approach for approximating non-smooth functions with wavelet basis functions. This is because the most
important $k$ coefficients may be very different from the first $k$ coefficients. We shall have more to say on the importance of structure in the function space and the ordering of the function space in Chapters 5 and 6.

![DCT reconstruction using first-100 coefficients](image1)

**Figure 2.4** Reconstructions of a blocky function using the first-100 and the biggest-100 coefficients of the DCT

### 2.3 Loss Functions: Maximum Likelihood Estimation

We now turn to the definition of the loss function for the determination of the empirical risk,

$$ R_{\text{emp}}(\theta) = \frac{1}{M} \sum_{k=1}^{M} \ell(y_k - F(x_k, \theta)) $$

c.f. (2.6)

which defines a measure of the agreement between the available data and the estimated function. In order to select a suitable loss function, we can resort to the concept of *maximum likelihood estimation* (see Press *et al*, 1992). This is based on establishing the probability of a data set occurring from the model with a given set of
parameters. Suppose M data points \((x_i, y_i)\) are given and the goal is to fit them to a model \(F(x, \theta) = \sum_{i=1}^{N} a_i \theta_i(x)\) with adjustable parameters \(\theta\). The parameters \(\hat{\theta}\) can be obtained by minimizing the empirical risk that measures the agreement between the data and the model.

For a given model, some parameter values will be highly unlikely, those for which the model predictions are at odds with the observed data. Others will be more likely and produce model predictions which resemble the data. We wish to select among the likely parameters those which are most likely and reproduce the closest agreement with the true function. This goal can be achieved by finding the probability that the observed data set could have occurred for a given model and a particular set of parameters. In other words, we should compute the probability of a data point occurring within an interval \(F(x_i, \theta) \pm \Delta y\) for a given model with a particular choice of parameters. Suppose that each observation \(y_i\) has an independently random measurement error \(\varepsilon_i\) drawn from a certain distribution \(\rho(\varepsilon_i)\) centred on the true value \(f(x_i)\). The probability of an individual data point \(y_i\) occurring within the interval \(F(x_i, \theta) \pm \Delta y\) may be taken as,

\[
p_i \propto \rho(y_i, F(x_i, \theta)) \Delta y,
\]

where \(\rho\{\cdot\}\) is chosen to correspond to the distribution of the error \(\varepsilon_i\). In general, the probability density function \(\rho(y_i, F(x_i, \theta))\) does not depend on its individual arguments independently and is only a function of their weighted difference defined as,

\[
\delta_i(\theta) = \frac{y_i - F(x_i, \theta)}{\sigma_i}
\]

(2.17)

The probability of the entire data set occurring is the product of the probabilities of the individual data points,

\[
P(\theta) = \prod_{i=1}^{M} \rho(\delta_i(\theta)) \Delta y
\]

(2.18)

The best-fit parameters are obtained by seeking to maximise the probability \(P(\theta)\) of the data given the model. Maximising \(P(\theta)\) is equivalent to minimising the negative of its logarithm, which we may take as the empirical risk,
\[
\min_{\hat{a}} \mathcal{R}_{\text{emp}}(\hat{a}) = -\ln(P(\hat{a})) = -\sum_{i=1}^{M} \ln[p(\delta_i(\hat{a}))] - M \ln(\Delta y) \tag{2.19}
\]

Defining the loss function as \( \ell(\delta_i) = -\ln[p(\delta_i(\hat{a}))] \) and dropping the constant term \( M \ln(\Delta y) \) leads to,

\[
\min_{\hat{a}} \mathcal{R}_{\text{emp}}(\hat{a}) = \sum_{i=1}^{M} \ell(\delta_i(\hat{a})) \tag{2.20}
\]

Using the optimality condition
\[
\frac{\partial \mathcal{R}_{\text{emp}}(\hat{a})}{\partial \hat{a}} = 0, \tag{2.21}
\]

and noting that
\[
\delta_i(\hat{a}) = \left(\frac{y_i - F(x_i, \hat{a})}{\sigma_i}\right), \quad \text{c.f.}(2.17)
\]

and \( F(x, a) = \sum_{i=1}^{N} \theta_j(x) \) leads to a system of nonlinear equations

\[
\sum_{i=1}^{M} \frac{d}{d \delta_i} \ell(\delta_i) \left(\frac{1}{\sigma_i}\right) \theta_j(x_i) = 0, \quad j = 1, 2, ..., N \tag{2.22}
\]

which must hold at the minimum. The best parameters \( \hat{a} \) may be computed either by directly minimising the multivariate objective function (2.20) or by solving the set of non-linear equations (2.22). However, except for the simple case of a normal distribution, either approach involves substantial numerical difficulties and presents a major challenge (Press et al., 1992).

If the distribution of measurement error \( \rho(\varepsilon_i) \) is known a priori and the probability distribution \( \rho(\delta(\hat{a})) \) is chosen accordingly, then we are able to quantify the goodness of fit of the model directly. For a correct model, the calculated distribution of the fitted residuals \( \hat{\varepsilon}_i = y_i - F(x_i, \hat{a}) \) should resemble the known distribution of the measurement error \( \rho(\varepsilon_i) \). In the simplest case of normal measurement errors, the probability \( \rho(\delta_i) \) can be taken as a Gaussian distribution with a zero mean and standard deviation \( \sigma_i \),

\[
\rho(\delta_i(\hat{a})) \propto \exp \left[ -\frac{1}{2} \left( \frac{y_i - F(x_i, \hat{a})}{\sigma_i} \right)^2 \right] = \exp \left( -\frac{1}{2} \delta_i^2(\hat{a}) \right) \tag{2.23}
\]
which implies that the corresponding loss function is:

\[ \ell(\delta_i(a)) = \frac{1}{2} \delta_i^2(a) \]  

(2.24)

For this case, the empirical risk (2.20) reduces to the familiar "chi square" or the "weighted sum of squared residuals",

\[ \min \ \mathbb{R}_{\text{emp}}(a) = \chi^2 = \frac{1}{2} \sum_{i=1}^{M} \delta_i^2(a) = \frac{1}{2} \sum_{i=1}^{M} \left( \frac{y_i - F(x_i, a)}{\sigma_i} \right)^2 \]  

(2.25)

If the variances are assumed identical \( \sigma_i = \sigma \), the objective function reduces further to the familiar least square criterion,

\[ \min \ J(a) = \chi^2 = \frac{1}{2\sigma} \sum_{i=1}^{M} (y_i - F(x_i, a))^2, \]  

(2.26)

The normal (Gaussian) distribution of errors has an unrealistically narrow tail and is often rather poorly realized in practice. A more realistic distribution with a wider tail is the double or two-sided exponential,

\[ \rho(\delta_i(a)) = \exp\left[-|\delta_i(a)|\right] = \exp\left[-\frac{y_i - F(x_i, a)}{\sigma_i}\right] \]  

(2.27)

For this distribution, the loss function \( \ell(\delta_i(a)) \) is given by,

\[ \ell(\delta_i(a)) = |\delta_i(a)| \]  

(2.28)

and the empirical risk reduces to the sum of absolute deviations,

\[ \min \ \mathbb{R}_{\text{emp}}(a) = \sum_{i=1}^{M} |\delta_i(a)| = \sum_{i=1}^{M} \left| \frac{y_i - F(x_i, a)}{\sigma_i} \right| \]  

(2.29)

A distribution with an even more extensive tail is the Cauchy or Lorentzian distribution,

\[ \rho(\delta_i(a)) = \frac{1}{1 + \left( \frac{y_i - F(x_i, a)}{\sigma_i} \right)^2} \]  

(2.30)

The corresponding loss function is,

\[ \ell(\delta_i(a)) = \ln(1 + \frac{1}{2} \delta_i^2(a)) \]  

(2.31)

and the empirical risk for the Cauchy distribution takes the form,

\[ \min \ \mathbb{R}_{\text{emp}}(a) = \sum_{i=1}^{M} \ln(1 + \frac{1}{2} \delta_i^2(a)) = \sum_{i=1}^{M} \left[ \ln(1 + \frac{1}{2} \left( \frac{y_i - F(x_i, a)}{\sigma_i} \right)^2) \right] \]  

(2.32)
The various distributions and their corresponding loss functions are contrasted in Figure 2.5. The double sided exponential and the Cauchy distributions exhibit a much broader tail compared to the normal distribution.

Figure 2.5 Comparison of the Gaussian distribution, double sided exponential and Cauchy (Lorenzian) distributions and their corresponding loss functions

2.4 Empirical Risk Minimization: Estimating the Parameters

In Chapter 6 we shall consider the subject of robust estimation, which deals with outliers and long tail distributions in some detail. For present purposes, we shall focus on the Gaussian distribution in order to introduce some of the machinery required for minimizing the empirical risk. The best-fit parameters $\hat{a}$ are those that minimize the empirical risk based on a suitably defined loss function. For a Gaussian noise distribution, the empirical risk for the linear model
is the chi-square, which in matrix form can be expressed as

\[ R_{\text{emp}}(a) = \chi^2 = \frac{1}{2} (y - Aa)^T (y - Aa) = \frac{1}{2} \| y - Aa \|_2 \]  \hspace{1cm} (2.33)

Here \( a \) is the vector of \( N \) parameters, \( y \) is the vector of \( M \) sample values, and \( A \) is the \( M \times N \) "design matrix" with elements:

\[ A_{i,j} = \theta_j(x_i) \]  \hspace{1cm} (2.34)

Note that without loss of generality, we have taken \( \sigma_i = \sigma = 1 \) to simplify the presentation. The necessary conditions to minimize the empirical risk are therefore,

\[ \frac{\partial R_{\text{emp}}(a)}{\partial a} = \frac{\partial \chi^2(a)}{\partial a} = (A^T A)a - A^T y = 0, \]  \hspace{1cm} (2.35)

which constitutes a set of \( N \) linear equations, often called the normal equations. In theory, the best-fit parameters can be obtained by solving the linear system,

\[ (A^T A)a = A^T y \]  \hspace{1cm} (2.36)

Provided that \( (A^T A) \) is invertible, the unique best-fit parameters \( \hat{a} \) are then given by

\[ \hat{a} = (A^T A)^{-1} A^T y \]  \hspace{1cm} (2.37)

It follows that the minimum empirical risk is

\[ (R_{\text{emp}})_{\text{min}} = \chi^2_{\text{min}} = \frac{1}{2} \| (I_M - (A^T A)^{-1} A)y \|_2 \]  \hspace{1cm} (2.38)

For the special case of \( N = M \), and a non-singular matrix \( A \), the best fit parameters are given by

\[ \hat{a} = A^{-1} y, \]  \hspace{1cm} (2.39)

and it follows that

\[ (R_{\text{emp}})_{\text{min}} = \chi^2_{\text{min}} = 0, \]  \hspace{1cm} (2.40)

This simply states that in this case the noisy data points are interpolated exactly. Evidently, interpolating a noisy data set will inevitably lead to a highly oscillating model and a very serious generalization error. In general, the number of data points is much greater than the number of basis functions \( N \) and the system (2.36) must be solved numerically.
2.4.1 Singular Value Decomposition: Solving the Normal Equations

The normal equations (2.36) should in theory be an over determined system of equations as there are usually many more data points than basis functions, $M >> N$. It is therefore expected that the $N \times N$ matrix $(A^T A)$ is well conditioned, which suggests that the system (2.36) may be solved by a variety of usual techniques such as Gaussian elimination, LU decomposition, QR decomposition, etc (Press et al, 1992). In practice, however, there are many cases where the normal equations are degenerate and the matrix $(A^T A)$ is close to singular. In such cases, the usual techniques for the solution of a system of linear equations encounter severe numerical difficulties. Classical methods either fail altogether, for example by encountering a very small pivot, or the solution vector $\mathbf{a}$ contains very large elements with alternating signs that are delicately balanced to cancel out when the fitted function is evaluated at the data points. Such an oscillating model inevitably leads to gross generalization error.

The reason that $(A^T A)$ is often severely ill-conditioned is easy to explain. The data available do not often clearly distinguish between two or more of the basis functions employed. If two basis functions or two different combinations of basis functions fit the data equally, the matrix $A^T A$ is unable to distinguish between them. This means that two rows or two combinations of rows of this matrix become very nearly linearly dependent. The usual solution techniques are incapable of recognizing such linear dependencies and this often results in catastrophic failure. For a robust solution of the normal equations (2.36), we need a technique which can both recognize and cure the potential linear dependency problem. This is provided by the powerful Singular Value Decomposition (SVD) method.

Singular value decomposition (SVD) is a powerful technique for dealing with sets of linear equations that are either singular or numerically very close to singular. The SVD method is based on the following theorem of linear algebra (Golub and Van Loan, 1996)

"Any matrix $A \in \mathbb{R}^{M \times N}$ with $M \geq N$ can be decomposed into,

$$A = UV^T$$

(2.41)"
where $U$ is an orthonormal $M \times M$ matrix ($U^TU = U^T = I_M$), $V$ is an orthonormal $N \times N$ matrix ($V^T V = V V^T = I_N$), and $W$ is an $M \times N$ diagonal matrix whose non-negative diagonal elements $w_j$, $(j = 1, \ldots, N)$ are the singular values of the matrix $A$.

Using the SVD of $A$ and the orthonormality of $U$ and $V$ matrices, the formal solution of the normal equations (2.36) is:

$$\hat{a} = V(W^TW)^{-1}WU^T \gamma = V \left[ \text{diag}\left(\frac{1}{w_j^2}\right) \right] WU^T \gamma$$

(2.42)

An important issue in minimizing the empirical risk is the confidence that we can place in the best fit parameters $\hat{a}$ obtained. In the least square case, this information is embedded in the covariance matrix of the estimated parameters:

$$C(\hat{a}) = (A^T A)^{-1},$$

(2.43)

with diagonal elements of $C$ representing the variance of the estimated parameters,

$$\sigma^2(a_j) = C_{jj}$$

(2.44)

The SVD solution of the least squares problem also delivers the elements of the covariance matrix $C$ directly,

$$C_{jk} = \sum_{l=1}^{N} \frac{1}{w_l^2} V_j V_k$$

(2.45)

The formal SVD solution (2.42) breaks down altogether if any of the singular values is exactly zero, the diagonal matrix $W^TW$ is then singular and cannot be inverted. Severe numerical difficulties will also be encountered if the ratio of the largest to smallest singular value is large $w_{\text{max}}/w_{\text{min}} >> 1$; this ratio is in effect a measure of the condition number of the matrix $A^T A$ and a large value signifies an ill-conditioned or numerically singular matrix. The small singular values identify those linear combinations which do not contribute much to reducing the $\chi^2$. Including these small singular values leads to a parameter vector with large elements of alternating sign that are delicately balanced to cancel out when the model is evaluated at the given data points. Evaluating the model with such parameters at other locations leads to severe oscillations and gross generalisation error. The simple remedy is to set the inverse of the singular values smaller than a certain threshold to zero, rather than a large number, before inverting $W^TW$. Editing out the small singular values puts aside
the nearly linearly dependent combinations and eliminates the gross generalisation error with only a negligible increase in the minimum of the $\chi^2$. The unique diagnostic properties of SVD make it the method of choice for solving least squares problem. The major difficulty in applying SVD is in specifying the appropriate threshold for insignificant singular values, which is highly problem dependent.

### 2.5 Complexity Control

Any reasonable estimation method needs to have some provision for complexity control to ensure that the selected model is appropriate for fitting the data set at hand. This requires an estimate of the inaccessible prediction risk and a variety of methods have been developed, primarily in connection with parametric modelling. A frequently used approach is based on data re-sampling, for example the popular leave one out cross validation and the generalized cross validation techniques (see for example Hastie and Tibshirani, 1990). A number of procedures have also been proposed where the computable empirical risk is penalised by an analytic function of some measure of complexity. In this approach,

$$\text{Predicted Risk} \quad \text{Penalisation Factor} \times \text{Empirical Risk}$$

$$= \left( \frac{N}{M} \right) \times \left( \frac{1}{M} \sum_{i=1}^{M} (y_i - F(x_i, a))^2 \right)$$ (2.46)

The penalisation factor $r(.)$ is designed to increase as model complexity increases and empirical risk is reduced. Various forms for the analytic penalisation factor have been discussed by Cherkassky et al. (1999) and Cherkassky and Shao (2001). Another frequently employed estimate of the prediction risk is the unbiased risk estimate of Stein (1981). Instead of an explicit estimation of the prediction risk, some similar measure such as an information criterion can be minimized for complexity control; a well known choice is the Aka Information Criterion (Akaike, 1974).

#### 2.5.1 Complexity Control using the Goodness of Fit

In the case of non-parametric function estimation, the form of the model is not known in advance and any of the above complexity control techniques can be used in
a constructive manner for basis selection. Vapnik (1982, 1995, 1998, 1999) has proposed that complexity control is best achieved by procedures that recognise and make use of the structure of the function space chosen. In order to extend the use of conventional model complexity procedures it is necessary to define counterparts which are designed to operate in a structured space. For example, Donoho and Johnstone (1995) succeeded in applying the unbiased prediction risk estimate of Stein (1981) to the multiresolution wavelet space, yielding the highly popular SURE wavelet shrinkage method for denoising. In this study we shall make use of the traditional goodness of fit measure defined locally as a means of complexity control in a structured function space. We therefore close this introductory chapter by providing a brief background to the traditional goodness of fit as a means of complexity control.

In a function estimation procedure dealing with noisy data we need a means to decide whether the estimated function is appropriate for fitting the data set at hand; that is we need to test the goodness of fit against a meaningful statistical standard. In order to provide a statistical measure of goodness of fit, it is essential that the probability distribution of the noise is available. In the case of Gaussian measurement error, the best fit parameters $\hat{\theta}$ are obtained by minimizing the $\chi^2(a)$ criterion:

$$\chi^2_{\text{min}} = \frac{1}{2} \sum_{i=1}^{M} \left( \frac{y_i - F(x_i, \hat{\theta})}{\sigma_i} \right)^2$$

(2.47)

The distribution of $\chi^2$ at its minimum plays a central role in determining the goodness of fit. For an appropriate estimated function, the normalized residuals $\left( (y_i - F(x_i, \hat{\theta})) / \sigma_i \right)$ are normally distributed but not independent and $\chi^2_{\text{min}}$ is therefore the sum of $M$ squares of normally distributed but correlated quantities. For linear models of the type considered in this study and Gaussian measurement noise, the probability distribution of different values of $\chi^2$ at its minimum can be derived analytically, and is given by the “chi-square distribution” for $\nu = M - N$ degrees of freedom (See Meyer, 1975).

$$CS(\chi^2, \nu) = \rho(\chi^2) = \frac{\chi^2^{(\nu-2)/2} e^{-\chi^2/2}}{\Gamma(\nu/2)2^{\nu/2}}$$

(2.48)

where $\Gamma(.)$ is the “incomplete Gamma function”.

39
Figure 2.6 shows the chi-square probability distribution for different degrees of freedom and the mean and variance of $\chi^2$ for $\nu$ independent degrees of freedom are:

$$E(\chi^2) = \nu \quad \text{and} \quad \text{var}(\chi^2) = 2\nu$$

(2.49)

For an appropriate linear model a reasonable value for $\chi^2_{\text{min}}$ is therefore $\chi^2_{\text{min}} = \nu$. If $\chi^2_{\text{min}} << \nu$, the variance of the residuals is much smaller than the noise variance, and the estimated function has captured the noise. If $\chi^2_{\text{min}} >> \nu$, then the variance of residuals is much larger than the noise variance and the estimated function has failed to capture the true function.

A more direct measure of the goodness of fit is given by the probability $q$ for $\chi^2_{\nu}$ to exceed $\chi^2_{\text{min}}$ by chance,

$$q = \Pr[\chi^2_{\nu} \geq \chi^2_{\text{min}}]$$

(2.50)
This probability is readily computed from the chi-square distribution,

\[ q = \int_{\chi_{\text{min}}^2}^{\infty} CS(\chi^2, \nu) d\chi^2 = Q\left(\frac{\nu}{2}, \frac{\chi_{\text{min}}^2}{2}\right) \]  

(2.51)

and may be expressed in terms of the complement of the incomplete Gamma function,

\[ Q(a, x) = \frac{1}{\Gamma(a)} \int_{x}^{\infty} e^{-t} t^{a-1} dt \]  

(2.52)

The values of this function are tabulated or can be computed using standard numerical routines. Figure 2.7 shows \( Q(\nu / 2, \chi_{\text{min}}^2 / 2) \) for different degrees of freedom as a function of \( \chi_{\text{min}}^2 \).

![Figure 2.7 Q(ν/2, χ² min/2) for different degrees of freedom.](image)

The probability \( q \) (0 ≤ q ≤ 1), gives a quantitative measure for goodness of fit of the noisy data by the estimated function. A value of \( q = 1 \) corresponds to a very small \( \chi_{\text{min}}^2 \rightarrow 0 \) and is indicative of serious overfitting. A value of \( q = 0 \) signifies a very large \( \chi_{\text{min}}^2 \rightarrow \infty \) and indicates that the estimated function has failed to capture the
true trend underlying the noisy data. A reasonable value for \( q \) is around 0.5, which corresponds to a \( \chi^2_{\text{min}} \) \( v \). For practical purposes, we should choose a range centred around \( q=0.5 \), say \( q_{\text{low}} < q < 1 - q_{\text{low}} \), as an acceptable range for the goodness of fit. Choosing a \( q_{\text{low}} \) close to 0.5, results in a small acceptable range and increases the probability of rejecting even the true function by chance. On the other hand, choosing \( q_{\text{low}} \) too close to zero, leads to a wide acceptable range and increases the probability of accepting an incorrect estimated function on par with the true function. To compromise between these conflicting criteria, the value of \( q_{\text{low}} \) is usually set in the range between 0.001 and 0.1.

The essential steps in a function estimation procedure were briefly introduced in this chapter. In particular, the need for a structured function space and a method for complexity control were emphasized. In the next chapter we introduce the multiresolution wavelet space as an efficient structured space for function estimation. Special decomposition methods capable of handling irregularly spaced data will be developed in Chapter 4. We shall demonstrate in Chapter 5 of this thesis that, provided the noise is Gaussian, the use of a local measure of the goodness of fit provides a powerful means of complexity control in a multiresolution structure. The proposed local goodness of fit (LGF) measure handles regularly and irregularly spaced data seamlessly. It can also handle non-identically, or spatially variable, Gaussian noise effectively but may break down when the noise is non-Gaussian or significant outliers contaminate the data. An effective means for complexity control for arbitrarily spaced data contaminated with arbitrary noise distribution is developed in Chapter 6 of this thesis based on a local balance of fit (LBF) measure.
Chapter 3

Multiresolution Analysis and Wavelets

The importance of choosing a proper function space for function estimation was discussed and the desirable properties of structured spaces was highlighted in Chapter 2. In addition, the important properties of basis functions such as locality and orthogonality were discussed. In this chapter, we consider multiresolution analysis (MRA) as a means of providing a useful structured space. Wavelets can lend themselves naturally to multiresolution analysis for the construction of a proper structured space for function estimation. It is the combination of properties of wavelets as basis functions and their ability to fit naturally in MRA that has seen an explosion in their applications in diverse areas such as function estimation, denoising, compression, etc. Here we explain the role of wavelets in Mallat’s multiresolution analysis and introduce the capability of wavelets in time frequency analysis. Some important issues such as computing the discrete wavelet transform, the important properties of wavelets and finally multidimensional wavelets will also be discussed.

3.1 Multiresolution Analysis

3.1.1 General Idea of Multiresolution Analysis

The basic aim in multiresolution analysis is the representation of a function at different levels of detail, where important features of the function become more apparent. The fundamental task in multiresolution analysis is to develop a nested ladder of interrelated subspaces, which can progressively capture finer and finer details of a function. Such a structured space is an essential ingredient for an efficient function estimation procedure. Within a multiresolution structure, a function \( f(x) \) may be expressed by its components at different scales (resolutions), that is:
\[ F_L(x) = F_J(x) + f_J(x) + f_{J+1}(x) + \cdots + f_{L-1}(x) \]
\[ = F_J(x) + \sum_{j=J}^{L-1} f_j(x) \]  

(3.1)

Here \( F_J(x) \) is the coarsest resolution (scale) representation and \( F_L(x) \) is the finest resolution representation of the function \( f(x) \). It is also clear that

\[ F_{J+1}(x) = F_J(x) + f_j(x) \]  

(3.2)

where \( f_j(x) \) represents the detail not captured at resolution \( j \). A coarse resolution represents the global trend (low frequency components) and a finer resolution contains more of the local features (high frequency components) of the function. Increasing the scale, the approximation looks finer until the true function is recovered at infinite scale:

\[ f(x) = \lim_{j \to \infty} F_j(x) \]  

(3.3)

The multiresolution decomposition of an arbitrary function is demonstrated by the synthetic construction sketched in Figure 3.1. The left hand panel shows progressively higher resolutions of the same function, and provides a description of the function at different scales. The right hand panel shows the detail added at each stage in going from a coarser to a finer scale. Given the finest resolution of the signal available in practice, the objective is to separate out the detail at each resolution clearly and accurately, enabling a well defined multiresolution approximation of the original signal.
Figure 3.1 Multiresolution decomposition of an arbitrary function. The right panel shows the detail added at each resolution.

3.1.2 Mallat's Multiresolution Analysis

The general idea of multiresolution analysis is not new. Many sets of basis functions can represent a function in a multiresolution manner. For example, we can use polynomials of different degrees to represent the coarse, $F_j(x)$, and the detail, $f_j(x)$, components of a polynomial function:
\[ F_L(x) = F_J(x) + \sum_{j=0}^{J-1} f_j(x) = \sum_{j=0}^{J-1} a_j x^j + \sum_{j=J}^{J-1} \sum_{k=0}^{k-1} a_k x^k \]  

(3.4)

This representation would, however, be inefficient because the polynomial basis functions are not orthogonal. The familiar Fourier expansion can also form a type of multiresolution analysis. In this case, the basis functions are orthogonal sinusoids and perform an efficient multiresolution analysis because there is no redundancy in the representation. However, sinusoidal basis functions are global and cannot efficiently represent functions with sharp local features.

A formal definition of an efficient multiresolution structure was presented by Mallat (1989). Following Mallat (1989), a multiresolution analysis of a function \( f(x) \) is a sequence of successive approximations \( F_j(x) \) resulting from the projection of \( f(x) \) onto a nested ladder of continuously larger subspaces of functions \( v_j \). Formally, the sequence of subspaces \( v_j \) in the multiresolution structure should possess the following properties:

**Nestedness:**
A higher index (finer resolution) subspace contains all the lower index (coarser resolution) subspaces:
\[ \ldots \subset v_{-1} \subset v_0 \subset v_1 \subset v_2 \subset \ldots \]  

(3.5)

**Completeness:**
The subspace \( v_j \) contains all square integrable functions as \( j \) goes to infinity.
Conversely it contains only the zero function when \( j \) tends to -
\[ \{0\} = v_{-\infty} \subset v_{-1} \subset v_0 \subset v_1 \subset v_2 \subset \ldots \subset v_\infty = L^2(R) \]  

(3.6)

**Dilation:**
All subspaces are dyadic scaled versions of just one subspace:
\[ f(x) \in v_j \iff f(2x) \in v_{j+1} \]  

(3.7)

**Translation:**
The space of a function contains all integer translated versions of that function:
\[ f(x) \in v_j \Rightarrow f(x - k) \in v_j \quad \text{for all } k \in \mathbb{Z} \]  

(3.8)
The sequence of subspaces $V_j$ (for $j \in \mathbb{Z}$) that satisfies the above properties form a multiresolution analysis of $L^2(\mathbb{R})$. We note that all the conditions of the structured multiresolution space are stated in the space (time) domain. The nested subspaces $V_j$ are shown schematically below. The efficiency with which the above multiresolution structure can be implemented depends on the properties of the basis functions employed to define the various subspace $V_j$. Scaling functions and wavelets provide a highly efficient set of interrelated basis functions for realising the structured space of multiresolution analysis.

![Schematic representation of nested function subspaces](image)

**Figure 3.2** Schematic representations of nested function subspaces

### 3.1.3 Scaling Functions and Wavelets

Simplistic attempts at multiresolution analysis such as polynomial and Fourier expansion do not possess the desirable properties highlighted by Mallat (1989). Evidently, particular basis functions should be designed deliberately to realise the formal structure of an efficient multiresolution analysis. Mallat (1989) showed that a unique function $\varphi(x)$ with an effective local support, called the *scaling function* can be determined such that its integer translations and dyadic dilations $\varphi_{j,k}(x)$ span the nested subspaces, $V_j$: 
\[ \phi_{j,k}(x) = 2^{j/2} \phi(2^j x - k) \]  
\[ v_j = \text{Span} \{ \phi_{j,k}(t) \} \quad k \in \mathbb{Z} \]

Here \( j \) is the index of dilation (resolution) and \( k \) is the index of translation of \( \phi(x) \).

The approximation to \( f(x) \) at resolution \( j \) may be written as a linear combination of the dilated and translated scaling functions:

\[ F_j(x) = \sum_{k=-\infty}^{\infty} c_{j,k} \phi_{j,k}(x) \]

In practice, we are concerned with a square integrable function with a finite duration, which is zero outside a limited range. Evidently, a scaling function translated outside this finite range will have a zero coefficients. This means that the translation index \( k \) has a limited range, which we can arbitrarily assign as \( 1 \leq k \leq N_j \)

\[ F_j(x) = \sum_{k=1}^{N_j} c_{j,k} \phi_{j,k}(x) \]

It is desirable that the translated versions of the scaling function at a given resolution \( j \) form an orthonormal basis for \( v_j \):

\[ \langle \phi_{j,k}(x), \phi_{j,l}(x) \rangle = \delta_{k,l} \quad \text{for } j,k,l \in \mathbb{Z} \]

In such a case, the coefficients of the expansion \( c_{j,k} \) can be simply obtained from the inner product:

\[ c_{j,k} = \langle f(x), \phi_{j,k}(x) \rangle = \int f(x) \phi_{j,k}(x) \, dx \]

For a non-orthonormal set of scaling functions, we must essentially solve a linear system of equations to obtain the coefficient \( c_{j,k} \).

There is a major drawback to multiresolution analysis with scaling functions alone. This is because the scaling functions at level \( j+1 \) are not orthonormal to the scaling functions at level \( j \). Put another way, the only requirement for subspaces \( v_j \) and
$v_{j+1}$ is that they are nested subspaces. Consequently, on going from a resolution $j$ to resolution $j+1$ it becomes necessary to calculate the coefficient for a totally different set of scaling functions $\varphi_{j+1,k}(x)$ at level $j+1$. It is of great practical importance that there exist a new set of basis functions to approximate the difference $f_j(x)$ between two successive resolutions. The difference between subspaces $v_j$ and $v_{j+1}$ will be denoted by a subspace $w_j$ and is shown in the sketch below. The difference (detail) between approximations at levels $j$ and $j+1$,

$$f_j(x) = F_{j+1}(x) - F_j(x),$$

will be contained in subspace $w_j$. Evidently, a different set of basis functions is required to define a ladder of subspace $w_j$ in order to capture the detail lost between level $j$ and $j+1$.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{nested_subspaces.png}
\caption{Schematic representation of nested function and detail subspaces}
\end{figure}

It is advantageous to define the subspace $w_j$ as the orthogonal complement to the space $v_j$:

$$v_{j+1} = v_j \oplus w_j$$

(3.16)

where $\oplus$ denotes orthogonal sum. This means that all the features captured by subspace $v_{j+1}$ and absent in subspace $v_j$ is contained within the subspace $w_j$. This would ensure that the subspace $w_j$ is orthogonal to both $v_j$ and all the other $w_j$'s, so that

$$v_j = v_j \oplus w_j \oplus w_{j+1} \oplus \cdots \oplus w_j$$

(3.17)

$$L^2(R) = v_j \oplus w_j \oplus w_{j+1} \oplus \cdots \oplus w_{j-1} \oplus w_j \oplus w_{j+1} \oplus \cdots$$

(3.18)
The combination of the nested subspace $v_j$ and the complementary orthogonal subspaces $w_j$ with the above properties ensure an efficient representation with minimal redundancy.

Mallat (1989) also demonstrated that for any admissible orthogonal scaling function there exist a unique function $\psi(x) \in L^2(R)$, whose integer translations and dyadic dilations constitute orthonormal bases for the orthogonal subspaces $w_j$,

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$$  \hspace{1cm} (3.19)

$$w_j = \text{Span}\{\psi_{j,k}(t)\}, \quad k \in \mathbb{Z}$$  \hspace{1cm} (3.20)

The function $\psi(x)$ is usually referred to as the mother wavelet and its dilated and translated versions $\psi_{j,k}(x)$'s are called baby wavelets. There is an intimate link between the scaling function $\varphi(x)$ and the wavelet function $\psi(x)$; such that for any admissible scaling function $\varphi(x)$, a mother wavelet $\psi(x)$ can be found based on the multiresolution and orthogonality properties of the subspaces. In particular, in an orthogonal wavelet system, the scaling functions at level $J$ are orthogonal to the wavelet functions at any resolution higher than $J$,

$$\langle \varphi_{j,k}(x), \psi_{j,l}(x) \rangle = \int \varphi_{j,k}(x)\psi_{j,l}(x) \, dx = 0 \quad \text{for } j = J, J+1, J+2, \ldots$$  \hspace{1cm} (3.21)

This means that we need only define the scaling functions at the coarsest resolution considered. The detail at the higher resolutions can be captured with wavelets without introducing any redundancy in the representation. Figure 3.4 shows typical examples of scaling functions $\varphi(x)$ and their corresponding mother wavelets $\psi(x)$. 
Using wavelets, the detail of approximation at scale \( j \) can be represented as a linear sum of translated and dilated wavelets

\[
f_j(x) = \sum_{k=1}^{N_j} d_{j,k} \psi_{j,k}(x)
\]  

(3.22)

The wavelets at resolution \( J \) capture the detail not covered by the scaling function at level \( J \). The approximation at resolution \( J+1 \) is therefore obtained by adding this detail,

\[
F_{J+1}(x) = F_J(x) + \sum_{k=1}^{N_j} d_{J+1,k} \psi_{J+1,k}(x)
\]

(3.23)

The overall approximation at the finest resolution \( L \) may be expressed as,

\[
F_L(x) = \sum_{k=1}^{N_J} c_{J,k} \phi_{J,k}(x) + \sum_{j=J}^{L-1} \sum_{k=1}^{N_j} d_{j,k} \psi_{j,k}(x)
\]

(3.24)
The scaling function \( \varphi(x) \) is effectively a low frequency function with a non-zero mean, its translations are used to approximate the function at the coarsest resolution \( F_r(x) \). The higher frequency content of the function is captured by the wavelet basis functions. Wavelets are effectively band pass filters with zero dc content and cannot capture the low frequency content effectively. The wavelet and scaling function coefficients, \( d_{j,k} \) and \( c_{j,k} \), are actually the coefficients of the wavelet transform of the approximated function \( F_L(x) \).

### 3.2 The Wavelet Transform

The transformation of a function (or signal) \( f(x) \) is a mathematical operation that results in a different representation which often exposes key features of the signal that are difficult or impossible to observe in the original domain. A transform determines the composition of a signal in terms of building block or “basis functions” of the transformed domain. We start by considering the well known Fourier Transform (FT), where a function is represented as a continuous sum (integral) of sinusoids of different amplitude, frequency and phase:

\[
\text{Forward Transform: } F(\omega) = \int_{-\infty}^{+\infty} f(x)e^{-i\omega x} \, dx \quad (3.25)
\]

\[
\text{Inverse Transform: } f(x) = \int_{-\infty}^{+\infty} F(\omega)e^{i\omega x} \, d\omega \quad (3.26)
\]

In Fourier analysis, the basis functions are orthogonal and global with an infinite duration. Fourier expansion determines the frequency components over the entire domain of a signal but does not specify in which interval of \( x \) which frequency components exist. It cannot reveal any information about the frequency content of a subinterval of the function \( f(x) \). The Fourier expansion is therefore suitable for analyzing stationary signals, with power spectrums that are constant over time.
3.2.1 Time-Frequency Analysis

Characterization of non-stationary signals and distinguishing the local features of a signal needs "time (space) -frequency analysis". The Fourier transform has no partitioning in time (space) and is therefore best suited to stationary signals. Conventionally, time frequency analysis is performed via a windowed Fourier transform. The basic assumption is that the power spectrum may be taken as constant over a short time (space) interval. The so-called Short Time Fourier Transform (STFT) is achieved by multiplying the function $f(x)$ by a sliding window function (e.g., a rectangular or a Gaussian window). This amounts to calculating the Fourier transform of a short time interval of a function. STFT maps the function into a two dimensional space of frequency $\omega$ and $x$ with fixed size time and frequency partitioning (tiling). The equal sized partitioning is perhaps the most severe problem of STFT. In general, high frequency components of a transient signal have a shorter time span than the low frequency components. This calls for wide windows at the low frequencies and continuously shorter windows at the higher frequencies. This is not provided by STFT, which complicates its application to non-stationary signals. Multiresolution analysis using wavelets provides a time (space)-frequency partitioning better suited for a non-stationary signal.

The complete set of wavelet basis functions are generated via dyadic dilation ($2^j$) and integer translations ($k$) of a mother wavelet function, $\psi(x)$. The shape of the wavelet does not change by dilation but its time duration and frequency content will change in the opposite directions. Figure 3.5 shows the cubic B-spline mother wavelet and its translated ($k=4$) and dilated ($j=1$) versions.
The low resolution wavelet basis functions have a wide duration in time (space) and a narrow band in the frequency domain while high resolution wavelets have short time durations and wide frequency bands. This results in a logarithmic partitioning in the frequency domain with the tiles having different dimensions in different frequency ranges. Figure 3.6 compares the time-frequency tiling for Wavelet Transform (WT), Short Time Fourier Transform (STFT), and Fourier Transform (FT).

Figure 3.5 The cubic B-spline mother wavelet and its translation (k=4) and dilation (j=1)

Figure 3.6 Time-frequency partitioning (tiling) of wavelet transform (WT), short time Fourier transform (STFT) and Fourier transform (FT)
3.2.2 The Fast Wavelet Transform (FWT)

Recalling the approximation by wavelets

\[ F_L(x) = \sum_{k=1}^{N} c_{j,k} \phi_{j,k}(x) + \sum_{j=1}^{L-1} \sum_{k=1}^{N} d_{j,k} \psi_{j,k}(x) \]  

we can consider the coefficients \( c_{j,k} \) and \( d_{j,k} \) as the Discrete Wavelet Transform (DWT) of the function \( f(x) \). For a set of orthogonal wavelets, the coefficients for a given function \( f(x) \) can be computed using the inner products:

\[ d_{j,k} = \int_{-\infty}^{\infty} f(x) \psi_{j,k}(x) \, dx \]  

\[ c_{j,k} = \int_{-\infty}^{\infty} f(x) \phi_{j,k}(x) \, dx \]

In practical situations, only discrete samples of the function \( f(x) \) are available, and a discrete form of the above integral must be used. Mallat (1989) has proposed a pyramidal algorithm based on convolutions to compute the coefficients efficiently. Mallat’s pyramidal algorithm is often called the Fast Wavelet Transform (FWT) because its computation requires \( O(M) \) flops (floating-point operations), which is even faster than \( O(M \log M) \) flops required by the FFT algorithm.

The FWT algorithm is derived in terms of the filter bank theory of electrical engineering and makes use of the recursive refinement equations for scaling functions and wavelets. It can be shown (Mallat 1989) that every scaling function can be constructed as a weighted sum of the scaling functions at the next higher resolution,

\[ \phi(t) = \sum_{n} h(n) \sqrt{2} \phi(2t - n) \]  

where \( h(n) \) is the scaling filter coefficients (scaling vector) that may be real or complex. This recursive equation is the fundamental equation for scaling function design and is called the refinement equation for scaling functions. A similar refinement equation can be derived for wavelets:

\[ \psi(t) = \sum_{n} g(n) \sqrt{2} \phi(2t - n) \]
This equation can be used to derive the mother wavelet from the scaling function using a related set of coefficients $g(n)$. The recursive equations (3.29) and (3.30) show how the coefficients $h(n)$ and $g(n)$ wholly determine the scaling function $\varphi(t)$ and the mother wavelet $\psi(t)$.

Multiresolution analysis can also be performed without referring to either the scaling functions or the wavelets explicitly. That is only the coefficients $h(n)$, $g(n)$, $c_{j,k}$, $d_{j,k}$ need be considered. The coefficients $h(n)$ and $g(n)$ can be considered as digital filters and $c_{j,k}$ and $d_{j,k}$ considered as digital signals. Actually, $h(n)$ and $g(n)$ are the so called Quadrature Mirror Filters corresponding to wavelets and scaling functions (Ramchandran et al 1996, Strang 1997). In the FWT algorithm, the refinement equations are used to calculate the scaling and wavelet coefficients of a coarse scale by convolution of the filter coefficients with the scaling coefficients of the finer resolutions:

\[
c_{j,k} = \sum_{m} h(m-2k)c_{j+1,m} \\
d_{j,k} = \sum_{m} g(m-2k)c_{j+1,m}
\]

(3.31)

Each of the above equations is exactly a digital filtering followed by a down sampling. The coefficients at the scale $j+1$ are filtered by two FIR filters $h(-n)$ and $g(-n)$, and then down sampled to give the next coarser scaling and wavelet coefficients. Filtering and down-sampling can be repeated on the scaling coefficients to give a multiscale structure. This is the so called “iterating filter banks” algorithm and is shown schematically below:

Figure 3.7 Iterating filter bank for computing the FWT
The inversion of the FWT is also achieved with great efficiency. The original fine scale coefficients of the signal $c_{j+1}$ can be reconstructed from the combination of the scaling coefficients $c_j$ and the wavelet coefficients $d_j$ at a coarser resolution:

$$c_{j+1,k} = \sum_m c_{j,m} h(m-2k) + \sum_m d_{j,m} g(m-2k)$$

(3.32)

This synthesis can be done by up-sampling $c_j$ and $d_j$, filtering them by $h(n)$ and $g(n)$ respectively and then adding them to each other.

The theoretical discrete wavelet transform algorithm is based on an infinite number of samples on the entire real line. In practical applications, however, data is only available over a finite interval and the wavelet transform should be adapted to it. In practice, $M=2^L$ equally spaced samples on the given interval are used because $M=2^L$ wavelets and scaling functions can form a complete orthonormal system. When the number of available data is not a power of 2 (non-dyadic), it can be extended to the next power of 2 value using a smooth extension method (for example by repeating the first or last values).

The basic FWT algorithm is based on convolution and down sampling, performing a convolution on a finite length signal results in border distortion. Therefore, in the case of a finite set of data, borders should be treated differently from the other parts of signal. There are two general methods to deal with the boundary problem. The first solution involves using specially designed wavelets for intervals (see Cohen, et al, 1993). Boundary wavelets are not entirely satisfactory because they require different wavelet filters are used at the boundaries. The second treatment, which is simpler, involves signal extension on the boundaries; to get perfect reconstruction, a few extra coefficients must then be computed at each stage of the decomposition (see Strang, 1996). Available signal extension methods include zero padding, symmetric extension, smooth padding and periodic padding.

A major advantage of the wavelet decomposition is the separation of the local and global features of a function in different resolutions. The following example is chosen to illustrate the multiresolution property of the wavelet transform and its ability to separate out localized high frequency features. The top left picture of Figure
3.8 shows a signal built by the addition of a low frequency sinusoid, $\sin(20^*x)$, and two local sinusoids with higher frequencies $0.2*\sin(320^*x)$ for $0.1<x<0.3$ and $0.2*\sin(180^*x)$ for $0.4<x<0.6$. The discrete signal was taken as 300 uniform samples of the combined functions. The approximations recovered at three successively finer resolutions are shown on Figure 3.8. The approximation at the coarsest resolution $F_c(x)$ represents the global trend of the original signal $\sin(20^*x)$ and is constructed using only the scaling functions. Increasing the resolution by one scale recovers the details in the range $0.4<x<0.6$ corresponding to $0.2*\sin(180^*x)$ and this localised high frequency feature is clearly captured by the algorithm. Increasing the resolution further, does not alter the function between $0.4<x<1.0$ but recovers the highest frequency feature $0.2*\sin(320^*x)$ localised at $0.1<x<0.3$.

![Figure 3.8](image)

**Figure 3.8** Separating the local and global features at different resolutions using wavelet decomposition

Wavelets also allow “sparse representation” of a signal; this is because the magnitudes of the wavelet expansion coefficients drop off rapidly for a large class of signals. The signal can therefore be efficiently represented by a small number of
wavelets. Consequently, wavelets and multiresolution analysis offer a natural and powerful technique for data compression and denoising. The need for efficient data compression algorithms capable of capturing and preserving localised features arises in diverse areas. These range from the archiving of fingerprint records by the FBI to the storage of the vast amounts of historical operating data logged daily in modern chemical and petrochemical plants. The following simple example is chosen to demonstrate the data compression capability offered by the wavelet transform. The denoising capability of the wavelet transform is the main subject of this study and will be discussed extensively in Chapter 5. The top left picture of the Figure 3.9 shows a signal consisting of 300 samples of an oscillatory function, direct storage of this data requires 2x300 numbers. The objective of data compression is to capture the picture over the entire range \([0, 1]\) without losing any local features while storing a smaller set of numbers. The top right picture shows a relatively coarse approximation of the signal, \(F_0(x)\), based on 19 scaling functions. We note that the picture is preserved with a root mean square error of 5% using only 19 stored coefficients. The bottom left and bottom right pictures demonstrate that on increasing the resolution of the approximation, the root mean square error is reduced at the expense of an increase in the number of coefficients. With 38 coefficients the error is 1.2% and with 76 coefficients the error is as low as 0.6%.

![Figure 3.9 An example of the data compression capability of wavelets](image-url)
3.3 Important Properties of Wavelet Systems

There are many types of carefully designed families of wavelets with specific mathematical properties. Some popular families of wavelets are Daubechies orthonormal wavelets, Cohen-Daubechies-Feauveau system of biorthogonal wavelets, Battle-Lamarie spline wavelets, Gaussian wavelets, Symlets, Coiflets, and Chui-Wang B-spline wavelets (see Daubechies, 1992). The qualities of different wavelet families vary according to several criteria, such as compactness, symmetry, number of vanishing moments, regularity and orthogonality.

**Localization**

The localization in time and frequency domains determines how wavelets (and scaling functions) tend to zero. It is desired that the wavelets have compact support (finite duration) because in this case the corresponding filters have a finite number of nonzero coefficients (FIR filter) and the wavelet transform can be efficiently calculated. On the other hand, a compact support in the frequency domain results in more regularity (smoothness) of wavelets.

**Symmetry**

A wavelet can be designed to be symmetric around its central time axis. A symmetric wavelet has a linear phase in the frequency domain. The linear phase property of symmetric wavelets is important in some applications such as signal and image processing, because it avoids the dephasing problem.

**Vanishing moments**

The number of vanishing moments of wavelets and scaling function

\[ \int x^n \psi(x) dx = 0 \quad \text{for } m = 0, \ldots, p - 1 \]

(3.33)

is an important indicator of the approximation power of wavelets. For a wavelet with a small number of vanishing moments, it may become necessary to go to very high resolutions in order to achieve a close approximation to a function \( f(x) \). The same approximation can, however, be achieved at a lower resolution level using a wavelet.
with a larger number of vanishing moments. The larger number of vanishing moment results in a more efficient (more compressed) approximation.

**Regularity**

The regularity of wavelets and scaling functions is a measure of its smoothness and is closely related to the number of vanishing moments. Generally, a higher number of vanishing moments results in a higher degree of regularity, which is useful for getting visually pleasing features, for example the smoothness of estimated function or the reconstructed signal or image.

**Orthogonality**

Orthogonality of the wavelet system is a key feature that results in considerable simplifications in constructing a wavelet-based approximation. In an orthogonal wavelet system there is no redundancy in the representation and each wavelet introduces an independent piece of information. Evidently, this results in a compact representation of an unknown function. In addition, the wavelet coefficients can be calculated using inner products that allows the implementation of a fast algorithm for the wavelet transform. An orthogonal wavelet transformation satisfies the Parseval Theorem and preserves the norm (energy) of the function: the norm of the function is equal to the norm of the coefficients. From a statistical viewpoint, an orthogonal wavelet transform maps the data to independent coefficients (white noise is mapped to white noise), which is of great significance in denoising applications.

Orthogonal wavelets can be designed either to have compact support or to decay toward zero sufficiently rapidly. The Daubechies family of wavelets (Daubechies, 1988, 1990) is perhaps the best known example of an orthonormal wavelet system with compact support (see Figure 3.10). However, with the exception of the discontinuous Haar wavelet, compact support orthogonal wavelets are not symmetric and lack smoothness. The Symlets (Daubechies, 1992) are designed to be the “least asymmetric” compactly supported orthogonal wavelets and Coiflets (Daubechies, 1992) are orthogonal compactly supported wavelets with improved smoothness (Figure 3.10). An example of an orthogonal wavelet family without compact support is the Battle-Lamarie spline wavelet (Daubechies, 1992), which has improved smoothness.
Truly symmetric wavelets can only be obtained at the expense of sacrificing a measure of orthonormality to arrive at the so-called biorthogonal wavelet. One example is the Cohen-Daubechies-Feauveau system of biorthogonal spline wavelets (Cohen et al., 1992). A biorthogonal system actually consists of two mutually orthogonal sets of wavelets and scaling functions. One set is used for decomposition and the other (dual) set is used for reconstruction purposes. Biorthogonal wavelet systems allow fast wavelet transform algorithms and can be designed to have symmetry (having linear phase filters) with desirable smoothness as well as compact support. However, as a result of loss of strict orthogonality, some important properties such as norm preservation and the independence of the coefficients are no longer available.

Attempts to make the biorthogonal wavelet systems nearly orthogonal have led to semi-orthogonal wavelets, an example of which is the Chui-Wang B-spline wavelet system. In a semi-orthogonal wavelet system, the wavelets at each resolution...
are orthogonal to all the wavelets at other resolutions but the wavelets and scaling functions at a specified resolution are not orthogonal. Figure 3.11 shows the cubic B-spline wavelet and scaling function, which are widely used in this study. There are some families of wavelets, which are neither orthogonal nor biorthogonal, but have other desirable properties such as symmetry and regularity. For example, Gaussian wavelets (nth degree derivative of the Gaussian function) are non-orthogonal with infinite support but they are symmetric and highly smooth (infinitely differentiable). One member this family is the Mexican hat wavelet, which is the second derivative of the Gaussian function and is illustrated in Figure 3.11.

\[ \text{Mexican Hat wavelet} \]

\[ \text{Cubic B-Spline wavelet} \]

\[ \text{Cubic B-Spline scaling function} \]

\[ \text{Gaussian scaling function} \]

\[ \text{Mexican Hat wavelet} \]

\[ \text{Cubic B-Spline wavelet} \]

\[ \text{Cubic B-Spline scaling function} \]

\[ \text{Gaussian scaling function} \]

**Figure 3.11** The biorthogonal Cubic B-Spline and the non-orthogonal Mexican Hat wavelets and scaling functions, see Appendix.

### 3.4 The Multidimensional Wavelet Transform

Up to this point, our discussion has been limited to the one dimensional case. In many practical applications, the data has more than one dimension. In image processing, the images are defined on a two-dimensional (2-D) mesh. In modelling and control, we frequently have to deal with two or more dimensions. Wavelet
transform and multiresolution analysis can be extended to multidimensional (n-D) cases. This section will focus on the construction of two-dimensional (2-D) wavelets and multiresolution analysis of two-dimensional square integrable functions \( f(x_1, x_2) \), \( f(x_1, x_2) \in L^2(R^2) \). However, extending the methods discussed to higher dimensions is straightforward.

From a construction point of view, the 2-D wavelets (and scaling functions) can be divided into separable and non-separable systems:

- Non-separable: genuinely 2-D wavelets (scaling functions)
- Separable: products of two 1-D wavelets (scaling functions)

In a non-separable n-dimensional system, there is a single n-dimensional wavelet that is directly constructed. This case will not be considered in this study. Multidimensional separable wavelets are much easier to construct than non-separable ones because they are simple products of one-dimensional wavelets. For a two-dimensional system, there is one 2-D scaling function and three 2-D wavelets, all of which are the products of 1-D scaling functions and wavelets. Suppose the scaling function and wavelet in the first dimension are \( \varphi(x_1) \) and \( \psi(x_1) \) and in the second dimension are \( \varphi(x_2) \) and \( \psi(x_2) \). The separable 2-D scaling function and wavelets are then given by:

\[
\begin{align*}
\varphi(x_1, x_2) &= \varphi(x_1) \varphi(x_2) \\
\psi^1(x_1, x_2) &= \varphi(x_1) \psi(x_2) \\
\psi^2(x_1, x_2) &= \psi(x_1) \varphi(x_2) \\
\psi^3(x_1, x_2) &= \psi(x_1) \psi(x_2)
\end{align*}
\]

where \( \psi^1, \psi^2 \) and \( \psi^3 \) capture the detail in the horizontal, vertical and diagonal directions, respectively.

It can be shown that the two-dimensional translations and dilations of the above wavelets form a two-dimensional multiresolution analysis (Daubechies 1992).
The 2-D scaling functions span the two-dimensional nested subspaces $v_j$ and the three 2-D wavelets span the complement (detail) subspaces $w_j$. If the one dimensional system is orthonormal, then $w_j$ is the orthogonal complement to $v_j$ and the separable two-dimensional wavelets form an orthonormal basis for $L^2(R^2)$. Figure 3.12 illustrates the two dimensional Mexican hat scaling function and wavelets. Using separable wavelets and two dimensional samples on an equally spaced dyadic rectangular grid ($2^L \times 2^P$), the 2-D wavelet transform can be simply converted to a sequence of fast 1-D wavelet transforms on the rows and columns of the data matrix.

![2-D Mexican hat scaling function](image1)

![First 2-D wavelet](image2)

![Second 2-D wavelet](image3)

![Third 2-D wavelet](image4)

**Figure 3.12** The two dimensional Mexican hat scaling function and wavelets

In the 2-D case a single 2-D scaling function and three 2-D wavelets are defined at each dilation and translation. Generally, for a separable n-dimensional wavelet system, there is one n-D scaling function and $2^{n-1}$ n-D wavelets, which makes the application of high-dimensional separable wavelet systems computationally more intensive.
3.5 Wavelets and Irregular Data

For uniformly sampled data, calculation of the coefficients in the multiresolutional wavelet approximation can take advantage of the simple refinement relations available for a uniform dyadic grid, leading to the highly efficient FWT algorithm. In this study, however, we are primarily concerned with irregularly spaced data with arbitrary sampling. In the case of non-uniformly sampled data, the convolution based decomposition methods, for example the FWT algorithm, cannot be applied directly and the calculation of the wavelet expansion coefficients becomes more complex.

When a finite number of samples on a uniform grid are available, the concept of discrete orthonormality becomes important. A set of wavelets are discrete orthonormal if

\[ \sum_{i=1}^{M} \psi_{j_1, k_1}(x_i) \psi_{j_2, k_2}(x_i) = 0 \quad \text{for } j_1 \neq j_2 \text{ and } k_1 \neq k_2 \tag{3.38} \]

\[ \sum_{i=1}^{M} \psi_{j, k}(x_i)^2 = 1 \tag{3.39} \]

where \( M \) is the number of uniform samples. For continuous orthonormal wavelets and a sufficiently dense equi-spaced data set, the above discrete orthonormality conditions are closely satisfied. This ensures that for uniform dyadic data the wavelet coefficients are essentially independent of each other. For irregularly sampled data, however, discrete orthonormality does not hold even for an orthonormal wavelet system. The combination of discrete orthogonality, simple refinement relations and independent wavelet coefficients, which together yield the fast wavelet transform algorithm, is not available with irregular data.

Using an orthogonal or biorthogonal wavelet system is not an essential prerequisite in dealing with non-uniformly sampled data. The orthogonality can therefore be sacrificed for the sake of symmetry and regularity of wavelets, both of which are important and desirable properties in function estimation. In the case of irregular data we can make equal use of orthogonal, biorthogonal and nonorthogonal wavelets. In this study we emphasize on the use of smooth symmetric wavelets, such
as the Symlet8, cubic B-spline and Mexican hat wavelets. The wavelet decomposition of irregularly spaced data must be performed using specially designed procedures. The various wavelet decomposition procedures suggested for irregular data will be considered in Chapter 4.
Chapter 4

Multiresolution Function Approximation for Irregularly Sampled Data

4.1 Introduction

Recovery of a continuous function (or signal) from its discrete samples is one of the major activities in real-world measurement. Traditional methods of signal reconstruction assume uniform spacing of the samples. However, actual data are not always uniformly sampled and may suffer from irregularities such as data dropout, multirate sampling, and sampling jitter. Figure 4.1 illustrates these irregularities in comparison with a uniform one-dimensional sampling grid.

![Figure 4.1 Different types of non-uniform sampling](image)
Traditional signal processing techniques cannot handle data irregularities directly. The problem of signal reconstruction from non-uniformly sampled data arises in diverse areas of science and engineering such as signal and image processing, computer graphics, communication systems, geo-science and identification and control.

The basic problem can be considered as the reconstruction of a multidimensional response surface from a limited number of irregularly spaced discrete samples, which are often noisy. From a mathematical view point, the fundamental problem is perhaps best considered as a function estimation problem. This allows us to use the substantial theoretical results and the ensuing algorithms developed for function estimation. A number of distinct approaches for the recovery of a function from irregular samples have been proposed. These include:

- **Statistical methods** (see Davis, 1986, Fieguth et al, 1995),
- **Fourier techniques employing a generalized sampling theorem** (see Feichtinger and Grochenig, 1992, Ferreira, 1992), and

In this study we are primarily concerned with the basis fitting methods, which approximate the function as the linear sum of a set of basis functions. Multiresolution basis fitting for irregular data can be considered using scaling functions alone (see for example, Iske et al 2002, and Iske 2004). In this study, however, we consider the wavelet basis functions in a multiresolution structure of the form,

\[
F_j(x) = \sum_{k=1}^{N} c_{j,k} \varphi_{j,k}(x) + \sum_{j=J}^{L-1} \sum_{k=1}^{N} d_{j,k} \psi_{j,k}(x) \quad (4.1)
\]

to estimate the unknown function from a limited set of possibly noisy samples \((x_i, y_i), i = 1, \ldots, M\). For uniformly sampled data, the calculation of the coefficients \(c_{j,k}\) and \(d_{j,k}\) in expansion (4.1) can take advantage of the fast wavelet transform (FWT) algorithm presented in Chapter 3. In the case of non-uniformly sampled data,
however, the FWT algorithm cannot be used directly and the estimation of the
wavelet coefficients becomes more complex. In this chapter we review the indirect
and direct approaches suggested for handling irregular data, and examine their
suitability for multidimensional function estimation from irregular noisy data sets.

4.1.1 Indirect Methods

In the most naive approach we can ignore the irregularity of the data
altogether. For example, in a one dimensional case we can simply use the rank of \( x_i \)
in place of \( x_i \). The data is first sorted into ascending order and then ranked on the
desired finest resolution \( L \) according to

\[
\text{rank}(x_i) = \frac{i}{2^L}, i = 1, \ldots, M
\]  

(4.2)

The sorted and ranked data can then be treated by the FWT algorithm. This approach
may prove useful in highly special cases involving weak departure from dyadic
regularity, for example a signal with minor sampling jitter. For strong irregularities,
however, the naive approach has severe drawbacks and the estimated function
inevitably has poor mean squared error performance, weak regularity properties and is
also graphically unpleasant (Cai and Brown, 1998, Lenarduzzi, 1997). A robust and
general technique must recognize the data irregularity and deal with it in a more
satisfactory manner.

A more useful indirect method involves the projection of the irregular data
onto a regular grid using a suitable projection technique (Lenarduzzi, 1997, Anoniadis
Kovac and Silverman 2000). Once the data has been projected, we are free to take
full advantage of the well developed machinery of the FWT algorithm with its
attendant computational efficiency. This method has been particularly well developed
by Kovac and Silverman (2000) in relation to one-dimensional denoising applications
and will be considered in more detail in section 4.2. A serious drawback of this
approach lays in the difficulty of its extension to two and higher dimensional function
estimation problems. In addition, the performance of projection methods is strongly
dominated by the particular interpolation scheme employed.
4.1.2 Direct Methods

In direct methods, the available irregular data is used directly without any manipulation. This has the advantage of circumventing the problems associated with the indirect projection method but comes at extra computational cost. We shall consider two schemes for direct handling of irregular data within the multiresolution wavelet structure: the lifting scheme popularized by Wim Sweldens and his co-workers (Sweldens 1994, 1995-a, 1995-b, 1996, Sweldens and Schröder 1995) and an optimization approach which is favoured in this study for denoising applications in high dimensions (Bakshi and Stephanopoulos 1993, Safavi et al 1994, Safavi 1995, Ford and Etter 1998). See also Kerkyacharian and Picard (2004) for a new direct method in one dimension based on wrapped wavelets.

The most direct way of handling irregular data is to employ data adapted basis functions or the so called second generation wavelets. This type of wavelet decomposition can be performed through the lifting scheme, which was originally developed for regular data as an alternative computationally more efficient algorithm than the FWT. The lifting scheme has since been extended to the irregular setting but at a substantial cost. In particular, the wavelets at a given resolution are no longer the translated version of the same basis function. In addition, the wavelet basis functions at successive resolution are no longer the dilated versions of each other. This means that the coefficients at each resolution must be determined and stored separately, which increases the computational load. The chief advantage of the lifting scheme lies in its ability to provide the perfect reconstruction of the original irregular data, which is important in some application such as computer graphics. Perfect reconstruction is not, however, a primary aim in denoising applications. In section 4.3 we present an introduction to the irregular lifting scheme in order to highlight some of its inherent shortcomings for multidimensional function estimation from noisy data sets.

The difficulties associated with the lifting scheme can be largely circumvented by resorting to an optimization approach. In this approach we shall use basis functions that are closely related to each other at each resolution as well as at successive resolutions. They are in fact the same as the basis functions used in the regular setting and hence the translated and dilated versions of the same mother
wavelet. In contrast to the projection and lifting approaches, however, we do not necessarily use the entire set of basis functions defined within the multiresolution structure. Instead at each resolution we first identify and select those basis functions that can contribute significantly to the approximation. We then set up and solve an optimization problem to obtain the coefficients for the selected basis functions. The optimization approach is discussed in detail in section 4.4 and is motivated by the observation that within the irregular data set certain regions can only support limited resolutions.

4.2 The Linear Projection Method

In this section we briefly review a linear projection method well developed by Kovac and Silverman (2000) using an orthonormal wavelet system. Given M irregular samples at locations $x = (x_1, x_2, ..., x_M)^T$ in the interval $I = [0, 1]$ and the corresponding measured values $y = (y_1, y_2, ..., y_M)^T$, Kovac and Silverman (2000) proceed by projecting the data on a regular grid with $M' = 2^L$ sample points given by,

$$x_k' = \frac{1}{M'} \left( k - \frac{1}{2} \right) \quad \text{for } k = 1, 2, ..., M' \quad (4.3)$$

The number of grid points $M'$ is taken as the first power of 2 larger than the number of the original data points,

$$L = \min \left\{ j \in \mathbb{Z} : 2^j > M \right\} \text{ and } M' = 2^L \quad (4.4)$$

The original irregular data is linearly interpolated to project the measured values onto this equally spaced grid (see Nason 2001),

$$y_k'' = \begin{cases} y_i & \text{if } x_k'' < x_i \\ y_i + (x_k'' - x_i) \frac{y_{i+1} - y_i}{x_{i+1} - x_i} & \text{if } x_i \leq x_k'' \leq x_{i+1} \\ y_M & \text{if } x_k'' < x_M \end{cases} \quad (4.5)$$

Note that we have used the Greek superscript $\mu$ to distinguish the variables associated with the synthetic equi-spaced grid. It should also be noted that a higher order interpolation can also be easily used in place of the linear interpolation.
The linear projection (4.5) can be represented in compact matrix notation by,

$$y'' = R'' y,$$  \hfill (4.6)

where the $M'' \times M''$ interpolation matrix $R''$ depends on both $x''$ and $x$. For linear interpolation, the matrix $R''$ is banded with each row having two non-zero entries which sum to one. The wavelet coefficients of the equi-spaced projected data are simply obtained using the FWT algorithm and may be represented by

$$d = W'' y''$$ \hfill (4.7)

where $W''$ is the $M'' \times M''$ orthogonal matrix associated with the particular set of orthogonal wavelets employed.

Substituting for $y''$ from (4.6) into (4.7) leads to,

$$d = \left[ R'' W'' \right] y$$ \hfill (4.8)

Here we note that the matrix $\left[ R'' W'' \right]$ is not necessarily orthogonal. In other words, recovering the wavelet coefficients of the original irregular data via a projection onto a regular grid does not yield an independent set of coefficients. This is true even when the irregular data is i.i.d. and the wavelets used are strictly orthogonal. This has important consequences in developing a denoising procedure, which will be further explored in the next chapter.

Another difficulty with the projection method is brought out by considering a simple example. Figure 4.2 shows the reconstruction of a simple sine function using Kovac and Silverman's approach. The irregular data consists of 50 randomly spaced samples of the sine wave shown by the plus sign. This data is projected on a uniform grid with 64 equi-spaced divisions using linear interpolation. The projected regular data is first transformed using the nearly symmetric and relatively smooth Symlet8 orthogonal wavelet system (Daubechies, 1992) and the function is reconstructed by taking the inverse transform. Overall, the reconstructed function is close to the original sine wave but there are significant differences in regions were the original data is sparse. In particular, the reconstructed function is practically linear in the sparse regions and this is not affected by changing the orthogonal wavelet system employed. This shows clearly that, within the sparse regions, the function
reconstructed by projection is dominated by the method of interpolation. A better approximation in the sparse region can only be obtained by increasing the order of interpolation. Increasing the order of interpolation, however, may lead to unwanted oscillatory behaviour in the boundaries between dense and sparse data regions. In addition, Kovac and Silverman (2000) do not consider the extension of their projection method to higher than one dimension, most probably because of the difficulties associated with developing two and higher dimensional interpolation schemes for irregular data.

![Figure 4.2 Wavelet reconstruction of a sinusoid from irregular samples using the Kovac-Silverman (2000) projection method](image)

4.3 The Lifting Scheme

The initial approach for constructing wavelets and the wavelet transform used sub-band filtering to decompose a signal into frequency bands. In particular, frequency domain analyses were employed to build the space-frequency localization property of wavelets, which limits this approach to equally spaced data. Building on previous work by Donoho (1992) and Lounsbery et al (1994), Wim Sweldens and his coworkers (Sweldens 1994, 1995-a, 1995-b, 1996, Sweldens and Schröder 1995,
Schröder and Sweldens 1995-a and 1995-b) have developed an alternative approach, the lifting scheme, in which all constructions are carried out in the spatial domain. Consequently, the lifting scheme can be readily generalized to handle irregularly spaced data. The lifting scheme has interesting algorithmic advantages such as in-place calculation, computational efficiency, and parallelism in implementation. Such features make the lifting scheme a powerful approach for a variety of applications employing wavelets.

The basic idea of the lifting scheme and its connection to multiresolution analysis is most easily presented using a notation borrowed from Sweldens (1995). Suppose we are given a set of \( M_j \) discrete samples of a function \( C(x) \),

\[
E_j = \{c_{j,1}, c_{j,2}, \ldots, c_{j,M_j}\}
\]

(4.9)

at arbitrarily spaced locations within the interval \( I=[0, 1] \),

\[
\mathcal{X}_j = \{x_{j,1}, \ldots, x_{j,M_j}\}
\]

(4.10)

The goal of multiresolution analysis is to transform this set into a smaller set \( E_{j-1} \) containing the coarse contents of \( E_j \),

\[
E_{j-1} = \{c_{j-1,1}, c_{j-1,2}, \ldots, c_{j-1,M_{j-1}}\}
\]

(4.11)

and a set \( d_{j-1} \) representing the details of \( E_j \),

\[
d_{j-1} = \{d_{j-1,1}, d_{j-1,2}, \ldots, d_{j-1,M_{j-1}}\}
\]

(4.12)

We shall subsequently identify the index \( j \) with the resolution (scale) and the samples \( c_{j,k} \) and \( d_{j,k} \) with the coefficients of the multiresolution wavelet expansion

\[
C_L(x) = \sum_{k=1}^{M} c_{J,k} \phi_{J,k}(x) + \sum_{j=J}^{L-1} \sum_{k=1}^{M_j} d_{j,k} \psi_{j,k}(x)
\]

(4.13)

where \( J \) and \( L \) represent the coarsest and finest resolutions of interest.

In the lifting scheme, a single step of the forward transform is achieved through sequential split, predict, and update steps shown schematically by the wiring diagram below:
In the split stage, the data set $c_j$ is simply divided into even and odd indexed sets, each with half the number of samples.

$$\text{SPLIT : } \left\{ \bar{c}_{j-1}, \bar{d}_{j-1} \right\} = S(c_j) \quad (4.14)$$

The set $\bar{c}_{j-1}$ includes the even indexed samples (the even or coarse set) and the set $\bar{d}_{j-1}$ includes the odd indexed samples (the odd or detail set).

$$\bar{c}_{j-1,k} = c_{j,2k} \quad k = 0, 1, ... \quad (4.15)$$

$$\bar{d}_{j-1,k} = c_{j,2k+1}$$

The split transform is of little value in itself as it does not decorrelate the original samples. However, it can be improved (lifted) through a predict stage operating on the even and odd sets.

Signals usually have local correlations and the even and odd sets emerging from the split will be highly correlated. In other words given one set, the other can usually be accurately predicted through a suitably defined prediction operator,
PREDICT:  \( d_{j-1} = \tilde{d}_{j-1} - P(\tilde{c}_{j-1}) \)  \hspace{1cm} (4.16)

A simple example is the linear predict operator which acts on adjacent members of the set \( \tilde{c}_{j-1} \)

\[
d_{j-1,k} = \tilde{d}_{j-1,k} - (p_0 \tilde{c}_{j-1,k} + p_1 \tilde{c}_{j-1,k+1}) \quad k = 0, 2, ..., M_{j-1} \quad (4.17)
\]

with

\[
p_0 = \frac{x_{j,2k+2} - x_{j,2k+3}}{x_{j,2k+1} - x_{j,2k+3}} \quad \text{and} \quad p_1 = \frac{x_{j,2k+2} - x_{j,2k+1}}{x_{j,2k+1} - x_{j,2k+3}} \quad (4.18)
\]

Due to the local correlation, we expect the coefficients \( d_{j-1,k} \) to be small (they will be exactly zero for a linear function) and regard the set \( d_{j-1,k} \) as the detail of the original set \( c_j \).

The raw even set \( \tilde{c}_{j-1} \) is not, however, a particularly useful representation of the coarse content of the sampled function. This is because \( \tilde{c}_{j-1} \) does not necessarily preserve key properties of the original signal such as its zero order (mean) or higher order moments. An update operator is introduced to ensure that the desired moments of the original signal are exactly preserved at successive resolutions. The \( q \)th order moment of a function \( C(x) \) is defined as,

\[
\mu^q = \int_0^1 x^q C(x) \, dx \quad (4.19)
\]

For a discrete representation at resolution \( j \) with samples \( c_{j,k} \), the above integral can be replaced with the sum,

\[
\mu^q = \sum_{k=1}^{M_j} c_{j,k} M_{j,k} \quad (4.20)
\]

where the scalar \( M_{j,k} \) is the \( q \)th order moment of the scaling function \( \varphi_{j,k}(x) \)

\[
M_{j,k} = \int_0^1 x^q \varphi_{j,k}(x) \, dx \quad (4.21)
\]
The update operator is designed to operate on the detail $d_{j-1}$ to transform the raw even set $\tilde{c}_{j-1}$ to an acceptable coarse representation $c_{j-1}$ which preserves the desired moments:

\[
\text{UPDATE: } c_{j-1} = \tilde{c}_{j-1} + U(d_{j-1}) \quad (4.22)
\]

A simple update operator which preserves the mean (zero order moment) is given by:

\[
c_{j-1,k} = \tilde{c}_{j-1,k} + (u_0 d_{j-1,k-1} + u_1 d_{j-1,k}) \quad (4.23)
\]

with the coefficients,

\[
u_0 = \frac{M_{j,2k-1}^0}{M_{j,2k-1}^0 + M_{j,2k}^0} \quad \text{and} \quad u_1 = \frac{M_{j,2k}^0}{M_{j,2k-1}^0 + M_{j,2k}^0} \quad (4.24)
\]

To summarize, one step of the forward transform is conducted through the three consecutive stages:

\[
\begin{align*}
\text{SPLIT: } & \{ \tilde{c}_{j-1}, \tilde{d}_{j-1} \} = S(c_j) \quad (4.25) \\
\text{PREDICT: } & \tilde{d}_{j-1} = \tilde{d}_{j-1} - P(\tilde{c}_{j-1}) \quad (4.26) \\
\text{UPDATE: } & c_{j-1} = \tilde{c}_{j-1} + U(d_{j-1}) \quad (4.27)
\end{align*}
\]

Starting from the finest resolution $L$ and iterating to the coarsest resolution $J$ provides a multiresolution decomposition of the function $C(x)$ in terms of a single coarse set (scaling coefficients) $c_J$ and the detail sets (wavelet coefficients) $d_J, d_{J+1}, \ldots, d_{L-1}$. Because the wavelet coefficients are typically sparse sets, this is likely to give a compact representation of the function. In addition, depending on the update used, one or more moments of the original function will be preserved as we traverse the multiresolution structure.

All the arithmetic operations on coefficients involved in the lifting scheme can be performed in place and are well suited to parallel implementation. More significantly, the inverse transform is simply obtained by reversing the order of operations and flipping the sign:
UNDO SPLIT: \[ c_j = \text{merge} \{ c_{j-1}, d_{j-1} \} \] (4.28)

UNDO PREDICT: \[ d_{j-1} = d_{j-1} + P(c_{j-1}) \] (4.29)

UNDO UPDATE: \[ c_{j-1} = c_{j-1} - U(d_{j-1}) \] (4.30)

The sketch below shows the wiring diagram of one step of the inverse transform. Starting from the coarsest resolution \( J \) and iterating the inverse transform to the highest resolution \( L \) provides a perfect reconstruction of the original discrete samples of the function \( C(x) \).

![Diagram](image)

**Figure 4.4** The lifting scheme inverse transform. Undo-update, Undo-predict and Merge stages

4.3.1 Second Generation Wavelets

The scaling and wavelet functions appearing in the multiresolution expansion of a regularly sampled function are translated and dilated versions of a mother wavelet \( \psi(x) \) and an associated scaling function \( \varphi(x) \). For irregular data, however, each coefficient \( c_{j,k} \) is associated with a unique scaling function \( \varphi_{j,k}(x) \) and each coefficient \( d_{j,k} \) is associated with a unique wavelet function \( \psi_{j,k}(x) \). These functions are no longer translates and dilates of each other, and their characteristics are strongly
influenced by the irregular sample spacing. These data adapted wavelets are often called second generation wavelets. Lifting provides a convenient means for obtaining both $\phi_{j,k}(x)$ and $\psi_{j,k}(x)$. To obtain $\phi_{j,k}(x)$, we construct a signal with all the detail and coarse coefficients set to zero except for the coarse coefficients $c_{j,k}$ located at $x_{j,k}$ which is set to one $c_{j,k} = \delta_{j,k}$. We then iterate the single step inverse transform ad infinitum, the resultant limit function gives $\phi_{j,k}(x)$.

$$c_j = \{c_{j,k} = \delta_{j,k}\}$$

$$\tilde{c}_j = \text{even}_j$$

$$d_j = \{d_{j,k} = 0\}$$

$$\tilde{d}_j = \text{odd}_j$$

**Figure 4.5** Construction of the scaling functions $\phi_{j,k}(x)$ using the lifting scheme

Evidently, the nature of the scaling function $\phi_{j,k}(x)$ depends on both the location (spacing of irregular data) and the particular predict operator employed. A family of predict operators can be readily constructed through the interpolating subdivision shown schematically in Figure 4.6. The basic task in subdivision is to fill the intermediate values in a smooth fashion by inserting a new predicted value between each pair of existing values. Interpolating subdivision can be achieved very efficiently using the Neville algorithm for polynomial interpolation (Press, *et al.*, 1992).
Figure 4.6 Schematic representation of subdivision for (a) linear interpolation and (b) cubic interpolation.

Figure 4.7 shows a particular spacing of irregular data by the crosses placed on the ordinate axis. The scaling functions constructed at two locations using a cubic interpolating predict operator are also shown on Figure 4.7. Note that there is a unique scaling function associated with each location. The character of the second generation scaling functions obtained is evidently highly dependent on the data spacing. Furthermore, in contrast to the first generation (regular) scaling functions, the second generation scaling functions are not simple translates of one another.

Figure 4.7 The second generation cubic interpolating scaling functions.
To determine the second generation wavelet function \( \psi_{j,k}(x) \) we proceed in a similar fashion. We first construct a signal with all the detail and coarse coefficients set to zero except for the detail coefficient \( d_{j,k} \) located at \( x_{j,k} \) which is set to one \( d_{j,k} = \delta_{j,k} \). The limiting function obtained on iterating the inverse transform ad infinitum provides \( \psi_{j,k}(x) \).

\[
\tilde{c}_j = \begin{cases} c_{j,k} = 0 \end{cases} \quad \text{even } j
\]

\[
d_j = \begin{cases} d_{j,k} = \delta_{j,k} \end{cases} \quad \tilde{d}_j = \begin{cases} d_{j,k} = \delta_{j,k} \end{cases} \quad \text{odd } j
\]

**Figure 4.8** Construction of the wavelet functions \( \psi_{j,k}(x) \) using the lifting scheme

The wavelet function \( \psi_{j,k}(x) \) depends on both the predict and the update operators. Predict operators are easily built by subdivision. Update Operators can also be readily designed. For example if we wish to preserve the first \( q \) moments of the signal between resolution \( j \) and \( j+1 \), we can proceed by forming the linear system

\[
\begin{bmatrix}
M_{j+1,k-q/2}^0 & M_{j+1,k-q/2+1}^0 & \cdots & M_{j+1,k+q/2}^0 \\
M_{j+1,k-q/2}^1 & M_{j+1,k-q/2+1}^1 & \cdots & M_{j+1,k+q/2}^1 \\
\vdots & \vdots & \ddots & \vdots \\
M_{j+1,k-q/2}^{q-1} & M_{j+1,k-q/2+1}^{q-1} & \cdots & M_{j+1,k+q/2}^{q-1}
\end{bmatrix}
\begin{bmatrix}
u_0 \\ u_1 \\ \vdots \\ u_{q-1}
\end{bmatrix} = \begin{bmatrix}
M_{j+1,k-1}^0 \\ M_{j+1,k-1}^1 \\ \vdots \\ M_{j+1,k-1}^{q-1}
\end{bmatrix}
\] (4.31)

at each location and solving it to obtain the corresponding update coefficients. The second generation wavelets constructed for the same irregular spacing and the cubic
interpolation predictor of Figure 4.7 and an update operator which preserves the zero and first order moments are shown in Figure 4.9. Once again the second generation wavelets are unique for each location and are not translates of one another.

The lifting scheme and second generation wavelets are the most direct way of dealing with irregular data. In particular, the irregular lifting scheme possesses the perfect reconstruction capability and is the method of choice when this is the ultimate goal. Figure 4.10 shows the perfect reconstruction of a sine wave function from 50 randomly spaced samples using the cubic interpolating predict operator and an update operator preserving the first two moments.
Figure 4.10 Reconstruction of a sinusoidal function from 50 randomly spaced samples using the second generation cubic interpolating wavelets.

The application of the elegant irregular lifting scheme for multidimensional denoising problems of interest in this study is, however, problematic for two reasons. First, the extension of the lifting scheme to two and higher dimensional data with arbitrary spacing is not possible at present. This must await further theoretical development of wavelets defined on irregular meshes, which is an area of research in its infancy (Daubechies et al. 1998, 1999; Guskov et al. 1999). Second, the development of denoising procedures via the lifting scheme is not straightforward, even in one dimension (Jansen and Bultheel, 1998, 1999; Simoens and Vandewalle, 2000-a, 2000-b; Delouille et al., 2001-a, 2004, Vanraes et al., 2002, Stepien and Zielinski, 2001). Denoising is achieved by first constructing the forward wavelet transform and then deleting a number of the smaller wavelet coefficients according to some threshold criterion before reconstruction. As a simple example, we can consider the samples of the sine wave of Figure 4.10 with a small level of Gaussian noise (standard deviation $\sigma = 0.05$) added at each point. Figure 4.11 shows the denoised signal reconstructed after setting the detail coefficients with amplitude less than 5% of the largest detail coefficient to zero. It is clear that thresholding the irregular lifting coefficients by this simple and frequently employed procedure leads to significant
spurious features.

![Noisy reconstruction](image1)

![Denoised reconstruction](image2)

**Figure 4.11** Reconstruction of a sinusoidal function from 50 randomly spaced noisy samples using second generation cubic interpolating wavelet

(a) all the wavelet coefficients are used

(b) only the larger wavelet coefficients are used

In the case of first generation (regular) wavelets it is possible to construct more sophisticated thresholding criteria in order to achieve a more robust denoising performance. Such procedures take full advantage of the fact that wavelets at successive resolutions form an orthogonal system and are simple translates and dilates of a single mother wavelet. The extension of such techniques to the irregular second generation wavelets is not straightforward, this is because they are not simple dilates and translates, and highly cumbersome thresholding procedures will be required. Even if we pay the price for developing more sophisticated thresholding techniques, the extension of the irregular lifting scheme to two and higher dimensions is not readily possible at the present. In this study we focus on a different direct approach, the least squares wavelet decomposition, which does not manipulate the irregular data but uses the first generation wavelets in a selective manner. The least squares approach can handle noisy irregular data directly and is also readily extended to higher dimensions.
4.4 The Least Squares Wavelet Decomposition

In this section and the remainder of this thesis we consider an alternative general optimization based decomposition in which the irregular data points are not manipulated in any way. In contrast to the lifting scheme, however, this approach relies on first generation wavelets which are the dilated and translated versions of a mother wavelet. The general approach described in this section relies on the formulation and solution of a variety of optimization problems to determine the coefficients in the multiresolution wavelet expansion,

\[ F_L(x) = \sum_{k=1}^{N_j} c_{j,k} \phi_{j,k}(x) + \sum_{j=J_k+1}^{L-1} \sum_{k=1}^{N_j} d_{j,k} \psi_{j,k}(x) \]

c.f. (4.1)

By taking this approach we sacrifice the strict interpolation capability of the irregular lifting scheme. We note, however, that strict interpolation is not of primary concern in denoising applications and the extension of the optimization-based procedures to higher dimensions is automatic. In our view, the direct optimization-based approach provides a pragmatic technique for handling multidimensional denoising applications. A popular optimization-based procedure is the least squares approach, which has also been employed by a number of other authors (Bakshi and Stephanopoulos 1993, Safavi, 1995, Ford and Etter, 1998), and is critically reviewed in this section.

In the case of regular data, the basis functions are defined on a multiresolution grid with nodes that precisely coincide with the data points. Each basis function is therefore seen by at least one data point and all basis functions make a significant contribution and are included in the expansion of the estimated function. Indeed in the FWT algorithm it is not necessary to consider the basis functions explicitly and it is sufficient to deal with the corresponding filter coefficients. The situation is substantially more complex with irregular data. This is because the spacing of the data does not coincide with the spacing of the regular multiresolution grid. There is in fact no longer a guarantee that each basis function has at least two data points in its (effective) compact support. This is illustrated in Figure 4.12, which shows an irregularly sampled function and a set of 16 regular triangular basis functions. The number of data points falling in the compact support of each basis function is also
shown on the abscissa. It is evident that there may be some basis functions with several data points in their compact support while other basis functions may have no data point in their support.

![Graph showing irregular samples of a function and regular triangular basis functions](image)

**Figure 4.12** Illustration of irregular samples of a function and regular triangular basis functions

Adding a basis function which is not seen by the data to the multiresolution expansion does not help in finding a closer approximation and may result in severe instabilities. When data is non-uniformly distributed in the input domain, there may be some regions which are denser than others. The dense regions can support a higher resolution than the sparse ones. This is because as we go up in resolution the basis functions become narrower and the probability of being seen by sparse data reduces. Figure 4.13, gives a sketch of the regular multiresolution grid with the centre of each regular basis function indicated by a cross. To use this grid and first generation (regular) wavelets for irregularly spaced data we must carefully select those basis functions that are seen by the data at each resolution.
4.4.1 Selection of Basis Functions

The application of the least squares method calls for identifying and putting aside those (regular) basis functions that cannot contribute to the wavelet expansion. The key criterion for including a basis function in the expansion is whether it contains sufficient data in its support. Various procedures can be developed by associating a particular wavelet to each tile in the space-frequency (or space-resolution) tiling of the wavelet system, the natural choice is the wavelet that is centred on each tile. The multiresolution tiling and the centre of the wavelets are shown on Figure 4.14 with a particular irregular data set superimposed. The key question to answer is whether there is sufficient data in a tile so that we may select its associated wavelet. It is clear that wavelets associated with tiles containing less than two data points should not be selected, otherwise the number of total basis functions at all resolution becomes more than the number of data points. In addition, the principal function of a multiresolution structure is to capture the difference between (at least two) data values by going to progressively higher resolutions with dilated wavelets. Included on Figure 4.14 are the locations of a set of arbitrarily spaced data points. The tiles containing less than two data points are shown as crossed tiles on Figure 4.14. We note immediately that with
irregular data, different regions can support different resolutions based on the *local* density of the data.

![Figure 4.14](image.png)

**Figure 4.14** The spatial variation in the local resolution supported by a typical irregular data set. Tiles with less than two data points are crossed.

The next question to answer is whether it is prudent to select the wavelets associated with tiles containing two or more data points. This is a difficult question since the answer depends on both the type of wavelet used and the distribution of the points within a tile. As we shall subsequently demonstrate, the criterion used for selection of the wavelets has a strong influence on the overall performance of the reconstruction algorithms based on the least squares method. To our knowledge this issue has not been directly addressed in previous literature and is therefore explored further.

We start with the simplest Haar wavelet system, which is sketched in Figure 4.15 at two successive resolutions. In order to select a Haar wavelet there must be at least one data point within each half of its support, it is only then that we can characterize the difference between these points. A tile with both data points on one
half of the support is not chosen and is shown as a crossed tile on Figure 4.15. We see from the left hand panel on figure 4.15 that the selection at each resolution can be performed independently.

![Figure 4.15 Selection of Haar wavelets based on the location of data, (*) centre of wavelet, and (o) location of data. Crossed tiles are not selected.](image)

Further complications arise for other wavelets with more complex shape. Figure 4.16 shows the tiling for the next simplest triangular wavelets at resolution $j=2$. It is intuitively clear that the wavelets associated with the two leftmost tiles should probably be selected. It is also clear that the wavelet associated with the rightmost tile should not be selected. This is because the two data points happen to coincide with the zero values of the associated wavelet. The situation for the third tile is less clear and depends on the value of the wavelet at the location of the data points.
A computationally simple solution is to select more conservatively and set the minimum number of data points required in a tile to a higher value than two. This would reduce, but not eliminate, the probability of encountering cases similar to those shown on the two rightmost tiles of Figure 4.16. An alternative but computationally more intensive approach is to use an additional criterion based on the discrete power of the wavelet. This may be defined as the sum of squared values of a basis function at the location of \( N_p \geq 2 \) data points falling in its associated tile,

\[
P_{j,k} = \sum_{i=1}^{N_p} (\psi_{j,k}(x_i))^2
\]

(4.32)

with only those wavelets with sufficient discrete power selected.
4.4.2 Computing the Wavelet Coefficients: Formulating the Least Squares Decomposition

Once the set of basis functions has been selected, it remains to form and solve a least squares problem to determine the coefficients in the multiresolution wavelet expansion. Consider the multiresolution wavelet expansion of a function written in its expanded form,

\[
F_L(x) = \sum_{k=1}^{N_J} c_{j,k} \varphi_{j,k}(x) + \sum_{k=1}^{N_J} d_{j,k} \psi_{j,k}(x) + \sum_{k=1}^{N_{J+1}} d_{j+1,k} \psi_{j+1,k}(x) + \ldots + \sum_{k=1}^{N_L} d_{L-1,k} \psi_{L-1,k}(x)
\]

(4.33)

Here \( \varphi_{j,k}(x) \) and \( \psi_{j,k}(x) \) are the dilated and translated versions of a single scaling function and a single mother wavelet respectively. The coarsest resolution is indicated by \( J \) and \( N_J \) is the number of basis functions selected at the coarsest resolution. Similarly, \( N_j \) represents the number of wavelet basis functions selected at resolution \( j > J \) and \( L \) represents the maximum finest resolution considered. Given \( M \) irregularly spaced samples of an unknown function in the interval \( I = [0, 1] \), \( (x_i, y_i), i = 1, ..., M \), our objective is to determine the coefficients in the above expansion so that the function may be evaluated at any arbitrary location within the interval \( I \).

The coefficients \( c_{j,k} \) and \( d_{j,k} \) can be determined by forming and solving least squares problems. The procedure used to formulate the least squares problem can seriously affect the quality of the reconstruction, a point that has been largely overlooked. In the reported applications of the least squares wavelet decomposition technique (Bakshi and Stephanopoulos 1993, Safavi, 1995, Ford and Etter, 1998) the coefficients are determined in a level by level (sequential) procedure. The level by level approach has a number of inherent shortcomings; in particular it may lead to gross errors in reconstruction. As an alternative we develop a formulation in which all the coefficients are determined simultaneously. The simultaneous formulation, which
to our knowledge has not been previously reported, overcomes the difficulty associated with the level by level sequential decomposition and can provide better reconstruction with irregular data. We shall first present the sequential and simultaneous formulation of the least squares decomposition procedures and follow this by a careful assessment of their characteristic performance in section 4.5.

In the level by level (sequential) approach, we start from the coarsest resolution \( J \) and as a first attempt try to approximate the unknown function with scaling functions alone,

\[
F_J(x) = \sum_{k=1}^{N_J} c_{J,k} \phi_{J,k}(x)
\]  

The coefficients \( c_{J,k} \) are found by solving the following least squares problem

\[
\min_{\vec{c}_J} \mathcal{R}_{\text{emp}}(\vec{c}_J) = \frac{1}{2} (y - G_J \vec{c}_J)^T (y - G_J \vec{c}_J)
\]

where \( \vec{c}_J = (c_{J,1}, \ldots, c_{J,N_J})^T \) is the vector of coefficients and the (design) matrix \( G_J \) is given by

\[
G_J = \begin{bmatrix}
\phi_{J,1}(x_1) & \cdots & \phi_{J,N_J}(x_1) \\
\vdots & \ddots & \vdots \\
\phi_{J,1}(x_M) & \cdots & \phi_{J,N_J}(x_M)
\end{bmatrix}
\]

The minimization of \( \mathcal{R}_{\text{emp}}(\vec{c}_J) \) reduces to the solution of the linear system

\[
(G_J^T G_J) \vec{c}_J = G_J^T y
\]

and methods for solving this linear system were discussed in Chapter 2. The first estimation of the function at the data points is given by

\[
\hat{y}_J = G_J \vec{c}_J
\]

and its residual with the measured values is defined as

\[
\varepsilon_J = y - \hat{y}_J
\]

We can view the residuals \( \varepsilon_J \) as the detail not captured by the scaling functions and attempt to fit them with the wavelets selected at resolution \( J \), by minimizing

\[
\min_{\vec{d}_J} \mathcal{R}_{\text{emp}}(\vec{d}_J) = \frac{1}{2} (r_J - H_J \vec{d}_J)^T (r_J - H_J \vec{d}_J)
\]
\[ \tilde{d}_j = (d_{j,1}, \ldots, d_{j,N_j})^T \]

is the vector of detail coefficients and the matrix \( H_j \) is given by

\[
H_j = \begin{bmatrix}
\psi_{j,1}(x_1) & \cdots & \psi_{j,N_j}(x_1) \\
\vdots & \ddots & \vdots \\
\psi_{j,1}(x_M) & \cdots & \psi_{j,N_j}(x_M)
\end{bmatrix}
\]  
(4.41)

This minimization in turn reduces to the solution of the linear system

\[
(H_j^T H_j) \tilde{d}_j = H_j^T r_j
\]  
(4.42)

and the next approximation to the measured values is given by

\[
\hat{y}_{j+1} = \hat{y}_j + \hat{\tilde{y}}_j = G_j \xi_j + H_j \tilde{d}_j
\]  
(4.43)

The detail not captured so far is given by the updated residuals

\[
\xi_{j+1} = y - \hat{y}_{j+1}
\]  
(4.44)

and an attempt is made to capture this detail with the wavelets selected at resolution \( J+1 \),

\[
\min_{\tilde{d}_{j+1}} \mathcal{R}_{\text{emp}}(\tilde{d}_{j+1}) = \frac{1}{2} (\xi_{j+1} - H_{j+1} \tilde{d}_{j+1})^T (\xi_{j+1} - H_{j+1} \tilde{d}_{j+1})
\]  
(4.45)

Solving the associated linear system,

\[
(H_{j+1}^T H_{j+1}) \tilde{d}_{j+1} = H_{j+1}^T \xi_{j+1}
\]  
(4.46)

provides the coefficients \( \tilde{d}_{j+1} \) and the next approximation is given by

\[
\hat{y}_{j+2} = \hat{y}_{j+1} + \hat{\tilde{y}}_{j+1} = (G_j \xi_j + H_j \tilde{d}_j) + H_{j+1} \tilde{d}_{j+1}
\]  
(4.47)

Repeating this process up to the finest resolution \( L \) provides all the coefficients \( c_{j,k} \) and \( d_{j,k} \) required for the multiresolution approximation.

As an alternative to the level by level decomposition, we may attempt to recover all the scaling and wavelet coefficients simultaneously. This is achieved by appending all the coefficient vectors together to form a single vector:

\[
a^T = (c_j^T \mid d_j^T \mid \tilde{d}_{j+1}^T \mid \cdots \mid d_{L-2}^T \mid d_{L-1}^T),
\]  
(4.48)

forming a large design matrix

\[
A = \begin{bmatrix}
G_j \mid H_j \mid H_{j+1} \mid \cdots \mid H_{L-2} \mid H_{L-1}
\end{bmatrix}
\]  
(4.49)

The overall linear system,
\[ \mathbf{Aa} = \mathbf{y} \]  

(4.50)

is in general over-determined and does not admit a solution. A solution which 
minimizes the sum of squared errors is obtained by attempting to minimize

\[
\min \mathcal{H}_{\text{amp}}(a) = \frac{1}{2} (\mathbf{y} - \mathbf{Aa})^T (\mathbf{y} - \mathbf{Aa})
\]  

(4.51)

The solution to this large minimization problem is obtained by solving the linear 
system,

\[
(\mathbf{A}^T \mathbf{A})\mathbf{a} = \mathbf{A}^T \mathbf{y}
\]  

(4.52)

using the methods described in Chapter 2.

The level by level and simultaneous formulation of the least squares wavelet 
decomposition yield identical results for regularly spaced data. In such cases discrete 
orthogonality ensures that the design matrices at different resolutions are mutually 
orthogonal,

\[
\mathbf{G}_j^T \mathbf{H}_j = [0] \quad \text{for } j < L
\]

\[
\mathbf{H}_j^T \mathbf{H}_k = [0] \quad \text{for } j < k, j \neq k
\]  

(4.53)

and the overall system (4.51) reduces to:

\[
\begin{bmatrix}
\mathbf{G}_j^T \mathbf{G}_j & [0] & \cdots & [0] \\
[0] & \mathbf{H}_j^T \mathbf{H}_j & \cdots & [0] \\
\vdots & \vdots & \ddots & \vdots \\
[0] & [0] & \cdots & \mathbf{H}_{L-1}^T \mathbf{H}_{L-1}
\end{bmatrix}
\begin{bmatrix}
\mathbf{e}_j \\
\mathbf{d}_j \\
\vdots \\
\mathbf{d}_{L-1}
\end{bmatrix}
= 
\begin{bmatrix}
\mathbf{G}_j^T \mathbf{y} \\
\mathbf{H}_j^T \mathbf{y} \\
\vdots \\
\mathbf{H}_{L-1}^T \mathbf{y}
\end{bmatrix}
\]  

(4.54)

We can also express all but the first element of the right hand side vector in (4.54) in 
terms of successive residuals. For example, noting that \( \mathbf{\gamma}_j = \mathbf{G}_j \mathbf{e}_j \) and \( \mathbf{r}_j = \mathbf{y} - \mathbf{\gamma}_j \),
we have

\[
\mathbf{H}_j^T \mathbf{r}_j = H_j^T (\mathbf{y} - \mathbf{\gamma}_j)
\]

\[
= H_j^T \mathbf{y} - [H_j^T \mathbf{G}_j] \mathbf{e}_j
\]

\[
= H_j^T \mathbf{y}
\]  

(4.55)

For regular data, therefore, discrete orthogonality ensures that the overall system 
(4.52) is reduced to exactly the same decoupled sequence of smaller least squares 
problems solved in the level by level method.
Discrete orthogonality does not hold for irregularly spaced data and the level by level design matrices are no longer strictly orthogonal. Different results can therefore be expected from the applications of level by level and simultaneous least squares decompositions. The application of the level by level least squares wavelet decomposition for irregular data can lead to gross interpolation error due to lack of discrete orthogonality. On the other hand, the sequential determination of the wavelet coefficients from successively smaller residuals imparts a degree of stability against improper wavelet selection. The simultaneous least squares decomposition does not rely on discrete orthogonality and therefore does not suffer from gross interpolation error. However, it is more sensitive to improper wavelet selection because it operates directly on the data vector rather than the residuals.

4.5 Results and Discussion

In the case of regular data, the data spacing (the finest resolution) is not spatially variable and a single finest resolution applies globally. This offers an opportunity to develop the highly efficient transformation embodied in the FWT algorithm. In the case of irregular data, the finest resolution is spatially variable and is dependent on the local density of the data. The lifting scheme employs the data adapted second generation wavelets and by developing all its constructions in the spatial domain deals with spatially variable data directly. The extension of the lifting scheme to arbitrarily spaced multidimensional data is not, however, possible at the present time. The least squares wavelet decomposition method is an approach which is easily extended to two and higher dimensions but employs regular first generation wavelets. In this section we explore the performance of the level by level (sequential) and the alternative simultaneous least squares wavelet decompositions for simple but carefully chosen irregularly spaced examples. Our objective is to highlight clearly the advantages and shortcoming of each technique and, where possible, make suggestions for improving the performance.
4.5.1 The Level by Level (Sequential) Least Squares Wavelet Decomposition

In the case of irregularly spaced data, the reconstructed function obtained by the level by level least squares decomposition may exhibit large deviation from the original data points. This gross interpolation error is strongly influenced by the starting coarse resolution selected. This point has not been directly addressed in previous literature, most probably because the irregular data sets employed did not deviate significantly from regular spacing. It is particularly surprising that this problem has not been discussed in articles on wavelet neural networks (Wave-Net), which are aimed specifically at handling irregular data (Bakshi and Stephanopoulos 1993, Safavi, 1995). It is therefore worthwhile to take a closer look at the basic reasons for the potential gross interpolation error in the level by level least squares wavelet reconstruction and propose suggestions to remedy this problem.

To highlight the serious nature of this difficulty we consider a carefully constructed irregular data set and the simplest Haar wavelet system. Figure 4.17 shows the level by level reconstruction of a sine wave from 50 irregular samples starting from 4 different coarse resolution at $J=0, 2, 4,$ and 6. The data set consists of $M=50$ discrete samples of a sine wave located at

$$x_i = \left(\frac{i}{M}\right)^2, \quad i = 0, 1, ..., M-1 \quad (4.56)$$

Selecting the sample locations in this way strongly emphasizes the irregularity of the data at low $x$ values. It is striking that the starting coarse resolution has such a strong influence on the form and quality of the reconstructed function. Starting from the coarsest possible resolution $J=0$ gives a reconstruction that is strongly biased and deviates significantly from the sample values over the entire interval. Setting the coarse resolution at $J=2$, the major deviation from sample values is confined to $x < 0.25$, where the irregularity is stronger. Starting from an even higher coarse resolution $J=4$ gives a better approximation and the bias is confined to $x < .0625$, where the irregularity is strongest. An overall measure of deviation from the sample values is the sum of squared error (SSE), which is 2.38 at $J=0$, 0.86 at $J=2$ and 0.043 at $J=4$. 


Figure 4.17 Influence of the starting coarse resolution on the performance of the level by level least squares wavelet decomposition
We should note at this point that the starting coarse resolution cannot be increased arbitrarily. This is demonstrated for the function reconstructed starting from a coarse resolution set at \( J = 6 \). The reconstructed function interpolates the data closely with an SSE of 0.0019 but there are several regions where it is identically zero. This is because as we increase the starting coarse resolution, the number of scaling functions increases and the tiles become narrower. In the case of irregular data, we may therefore encounter several tiles with no data point in the (effective) support of their associated scaling functions. Such scaling functions, which do not contribute to the wavelet expansion, are not selected and hence the reconstructed function will have an identically zero value within such intervals.

Understanding the basic reason for the observed gross error is essential for selecting the appropriate starting coarse resolution for the level by level least squares wavelet decomposition. We start by noting that gross reconstruction error does not arise with regularly spaced dyadic data but can be severe for irregularly spaced data. Figure 4.18 compares the reconstruction of a sine wave from 64 regular samples with that from 64 irregular samples. In both cases the level by level least squares decomposition was started at the coarsest resolution \( J = 0 \). The data is strictly interpolated for the regular case but shows gross trend error in the irregular case. The fundamental reason of the large difference between the reconstruction qualities is the absence of discrete orthogonality in the case of irregular data.

Figure 4.18 Reconstruction of a sine wave from 64 regular and 64 irregular samples using the level by level least squares decomposition with \( J = 0 \)
In a wavelet multiresolution structure, the approximation at level \( j+1 \) is built up by adding the detail at level \( j \) to the approximation at level \( j \):

\[
F_{j+1}(x) = F_j(x) + f_j(x)
\]  
\((4.57)\)

The discrete analog of the above relation in terms of the wavelet coefficients is given by:

\[
G_{j+1} = G_j + H_j d_j
\]  
\((4.58)\)

In the case of dyadic regularly spaced data, the sub-spaces of the scaling functions and the wavelets are designed to be the orthogonal complement of each other. This ensures that discrete orthogonality holds and the discrete refinement relation \((4.58)\) is satisfied exactly. The starting coarse resolution \( J \) in the level by level least squares decomposition is then immaterial and the same reconstruction is obtained starting from any chosen coarse resolution. For irregularly spaced data, however, discrete orthogonality is not guaranteed and the coefficients computed by the level by level decomposition do not satisfy the discrete refinement relation \((4.58)\) precisely. Consequently, the choice of the starting coarse resolution can strongly affect the quality of the reconstruction from irregular data as demonstrated on Figure 4.18.

The improvement in reconstruction quality observed in Figure 4.17 on increasing the starting coarse resolution \( J \) can be clearly explained for the Haar wavelet system employed. The task of the Haar scaling function is to capture the local average of the signal. The Haar wavelets, in common with all other wavelets, have a zero mean and their linear combination cannot alter the local averages. Figure 4.19 shows the local averages captured by the Haar scaling functions for regularly and irregularly sampled sine waves. We also note that, for the simple Haar scaling function, the numbers shown on Figure 4.19 are equivalent to the scaling coefficients produced by the decomposition. In the regular case, the sample averages coincide with the averages of the sampled function in each tile of the multiresolution structure. The Haar scaling functions at each resolution can therefore capture the local signal average closely within each tile (see left panel of Figure 4.19). We can therefore start the level by level reconstruction at any coarse resolution and the reconstructed signal will interpolate the regularly sampled data exactly.
For irregularly spaced data, however, the sample average within any given tile may differ markedly from that of the true function (see the right panel in Figure 4.19). The Haar scaling functions will then capture the (incorrect) sample average rather than the true signal average. The Haar wavelets added at the higher resolutions cannot alter such incorrect averages and this may lead to gross reconstruction error. The advantage of starting at a higher coarse resolution is due to the larger number of narrower scaling functions employed. Reducing the support of the scaling functions tends to reduce the effect of sample irregularity within the associated tiles. Selecting $J=0$ and a single scaling function, the reconstructed signal maintains an average of 0.17 at all resolutions rather than the true value of zero. Using four scaling functions by setting $J=2$, the average of the reconstructed function will be maintained at 0.05 which is closer to zero and the reconstruction error is reduced as demonstrated on Figure 4.17.

![Figure 4.19](image_url)

**Figure 4.19** Local averages within each tile for 64 regular and 64 irregular samples of a sine wave for the level by level least squares decomposition
The above arguments were confined to the simple Haar wavelet system for reasons of clarity. The same arguments, however, apply to any other wavelet system provided we consider weighted averages rather than simple averages. To reduce the potential for gross interpolation error for irregularly spaced data, the starting coarse resolution in the level by level least squares decomposition must be carefully selected. There appears to be no problem independent procedure available for such a selection and in our opinion a better approach is to resort to the simultaneous least squares decomposition, with all the coefficients determined at once rather than sequentially.

4.5.2 The Simultaneous Least Squares Wavelet Decomposition

In the simultaneous least squares decomposition all of the wavelet coefficients and the scaling coefficients are determined together by forming and solving a single linear system (4.52). Furthermore, no reference is made to the discrete refinement relation (4.58) in formulating this linear system. This allows the wavelet coefficients to adapt so that the details at the various resolutions are captured simultaneously and enables the scaling coefficients at the chosen coarse resolution to capture the local average of the signal accurately. Figure 4.20 shows the local averages captured by the simultaneous least squares decomposition with the Haar wavelet system for various starting coarse resolutions for the same data considered in Figure 4.19. It is evident that the local averages are captured accurately irrespective of the starting coarse resolution. There is in fact no discernable difference between the local averages computed from the regular and highly irregular data sets considered in Figure 4.19.

Figure 4.21 shows the reconstruction obtained from 64 regular and 64 irregular samples of a sine wave using the simultaneous least squares decomposition with the Haar wavelet system. It is evident that the irregular samples are interpolated perfectly with an SSE=3.6x10^{-31}. Using the level by level decomposition on the same irregular data set exhibited gross reconstruction error with an SSE=3.19, see Figure 4.18.
Figure 4.20 Local averages within each tile for 64 regular and 64 irregular samples of a sine wave for the simultaneous least squares decomposition.

Figure 4.21 Reconstruction of a sine wave from 64 regular and 64 irregular samples using the simultaneous least squares decomposition with $J=0$. 

Average $= 0.00$

Average $= 0.01$

$J=0$

$J=1$

$J=2$
The perfect reconstruction of the irregular data observed in Figure 4.21 is a special feature of the Haar wavelet system. For the simple Haar system, the number of wavelets selected $N$ equals the number of the irregular data points $M=N$. This turns the design matrix $A$ in system (4.50) into a full rank $N \times N$ matrix permitting perfect interpolation. For other wavelets with more complex shapes, some wavelets may contain insufficient data in their (effective) compact support and will not be selected. The number of selected basis functions is then smaller than the number of the irregular data points, $N < M$, making the linear system (4.50) over-determined. The least squares solution obtained by minimizing the empirical risk (4.51) will no longer achieve perfect interpolation. Figure 4.22 compares the reconstruction obtained from 64 regular and irregular samples of a sine wave using the cubic B-spline wavelet systems. In the irregular case, perfect interpolation is not achieved but the irregular data is still well interpolated with an $\text{SSE} = 1.7 \times 10^{-3}$.

![Figure 4.22 Reconstruction of a sine wave from 64 regular and 64 irregular samples using the simultaneous least squares decomposition using cubic B-spline wavelets](image)

**4.5.3 Influence of Basis Selection Procedure on the Least Squares Reconstruction**

Reconstructions based on the least squares decomposition are sensitive to the basis selection procedure adopted. The level by level procedure is sensitive to improper basis selection at low frequency (resolution). Selecting the scaling functions incorrectly may lead to gross interpolation error for irregularly spaced data. The level
by level method is, however, resistant to improper basis selection at high frequency (resolution). This is because the high resolution wavelet coefficients are used to approximate successively smaller residuals as we climb the resolutions. In the simultaneous decomposition, the scaling coefficients and all of the wavelet coefficients are determined together to approximate the original data vector rather than successive residuals. This eliminates the gross interpolation error tendency of the level by level decomposition but the reconstructed function may suffer from large peak oscillations. There is no simple problem independent remedy for curing the gross interpolation error tendency of the level by level decomposition for irregular data. The potential high peak oscillations of the simultaneous decomposition can, however, be readily circumvented by choosing the wavelet basis functions more conservatively. A simple example serves to highlight the problem and its remedy.

Figure 4.23 compares the reconstruction achieved from 150 randomly spaced samples of the function, $y = \sin(2\pi x) + 0.2\sin(20\pi x)$, using the level by level and simultaneous least squares decompositions with the cubic B-spline wavelets. In both cases, the starting coarse resolution was set at $J=5$ and only those basis functions with a minimum of two data points in their compact support were selected. It is clear that the level by level reconstruction cannot interpolate the data closely with an SSE as large as 0.1 and exhibits major local trend errors. The simultaneous reconstruction interpolates the data very closely with an SSE=4.5e-11 but suffers from large peak oscillations in sparse data regions.

**Figure 4.23** Comparison between reconstructed functions using the level by level and the simultaneous least squares decompositions
The large peak oscillations observed are, however, readily circumvented by selecting the basis functions more conservatively. This can be achieved either by increasing the required minimum number of data points for basis selection or by computing the discrete power of each wavelet (see equation 4.32) and selecting only those with sufficient discrete power. Figure 4.24 compares the reconstruction performance of the simultaneous least squares decomposition for progressively more conservative basis selection criteria. It is clear that we can eliminate the oscillations while maintaining good interpolation of the irregular data with an SSE of the order of $1 \times 10^{-4}$.

![Figure 4.24](image)

**Figure 4.24** The effect of different wavelet selection criteria on the quality of the simultaneous least squares reconstruction

In this study we are primarily interested in denoising applications and favour the simultaneous least squares decomposition for the following reason. The noise is usually high frequency and the denoising process will at the same time cure the high frequency problems of the simultaneous reconstruction. In other applications where the primary objective is accurate reconstruction from discrete irregular samples,
hybrid decomposition can combine the low frequency advantages of the simultaneous
decomposition with the high frequency benefits of the level by level decomposition.
For example, we can apply the simultaneous approach by selecting all the basis
functions with a minimum of say four points in their local support. Less “robust”
wavelets with a fewer number of data in their supports can be added, in a level by
level manner, starting from some intermediate resolution. Figure 4.25 demonstrates
the excellent and stable reconstruction achieved by the hybrid approach, using 4 and 2
data points for basis selection in the simultaneous and level by level decompositions
respectively.

![Figure 4.25 Reconstruction performance of the hybrid least squares wavelet
decomposition](image)

**4.6 The Multidimensional Least Squares Wavelet Decomposition**

A distinct advantage of the least squares wavelet decomposition is the
simplicity of its extension to multidimensional data with arbitrary irregular sample
locations. Here, we shall focus on the two-dimensional (2-D) extension but extending
the least squares decomposition to higher dimensions remains straightforward. To
perform 2-D wavelet decomposition, it is necessary to establish a 2-D multiresolution
grid and populate it with two dimensional wavelets. Figure 4.26 shows the two-
dimensional pyramidal grid of wavelet centres, which is simply an extension of the
one-dimensional grid of Figure 4.13. Note that the number of grid points at each
resolution $j$ is $2^j \times 2^j = 2^{2j} = 4^j$. A 2-D wavelet (or scaling function) has a
rectangular effective support whose dimensions are halved at the next higher
resolution.
We shall employ separable two dimensional wavelets, which are formed as the products of one dimensional wavelets and scaling functions. There is one 2-D scaling function and three 2-D wavelets centred on each tile of the 2-D multiresolution structure. For a tile with dilation index (resolution) $j$ and translation indices $k$ and $l$ in the $x_1$ and $x_2$ directions, we have:

$$\varphi_{j,k,l}(x_1,x_2) = \varphi_{j,k}(x_1)\varphi_{j,l}(x_2)$$

(4.59)

$$\psi^{1}_{j,k,l}(x_1,x_2) = \varphi_{j,k}(x_1)\psi_{j,l}(x_2)$$

(4.60)

$$\psi^{2}_{j,k,l}(x_1,x_2) = \psi_{j,k}(x_1)\varphi_{j,l}(x_2)$$

(4.61)

$$\psi^{3}_{j,k,l}(x_1,x_2) = \psi_{j,k}(x_1)\psi_{j,l}(x_2)$$

(4.62)

Here, $\varphi(x)$ and $\psi(x)$ are regular 1-D scaling function and wavelets. The 2-D wavelets $\psi^{1}$, $\psi^{2}$ and $\psi^{3}$ capture the details in the horizontal, vertical and diagonal directions respectively. Since there are three 2-D wavelets centred on each tile, the total number
of wavelets at each resolution $j$ is $3 \times 4^j$. Figure 4.27 illustrates the two dimensional scaling function and wavelets formed as products of the regular 1-D cubic B-spline scaling function and wavelet.

![2-d Cubic B-Spline scaling function](image1)

![First 2-d wavelet](image2)

![Second 2-d wavelet](image3)

![Third 2-d wavelet](image4)

**Figure 4.27** Two dimensional cubic B-spline scaling function and wavelets

Using the 2-D scaling functions and wavelets, the two dimensional multiresolution approximation of a function up to the finest resolution $L$ can be written as:

$$F_L(x_1, x_2) = \sum_{k=1}^{N_1} \sum_{l=1}^{N_2} c_{J,k,l} \phi_{J,k,l}(x_1, x_2)$$

$$+ \sum_{j=L}^{L} \sum_{k=1}^{N_1} \sum_{l=1}^{N_2} \begin{bmatrix} d_{j,k,l}^1 \psi_{j,k,l}^1(x_1, x_2) \\ d_{j,k,l}^2 \psi_{j,k,l}^2(x_1, x_2) \\ d_{j,k,l}^3 \psi_{j,k,l}^3(x_1, x_2) \end{bmatrix}$$  \hspace{1cm}(4.63)

As for the one dimensional case, the simultaneous least squares wavelet decomposition can be applied and a linear system formed by selecting those basis
functions with sufficient data in their compact support. Solving the corresponding least squares problem provides the 2-D scaling and wavelet coefficients in expansion (4.63). As an illustrative 2-D example we consider the reconstruction of the sinusoidal function,

$$f(x_1, x_2) = 1 + \sin(2\pi x_1) \sin(2\pi x_2), \quad (4.64)$$

from 400 randomly spaced samples within the unit square. Figure 4.28 shows the sample locations, the true function and the reconstruction achieved by the simultaneous least squares decomposition using the 2-D cubic B-spline wavelets. Only those basis functions with a minimum of 4 data points in their compact support were selected. It is clear that a smooth and faithful reconstruction is achieved and an \text{SSE}=4.1\times10^{-3} confirms that the irregular data is closely interpolated.

### 4.7 Conclusions

The highly efficient FWT algorithm is designed for strictly regularly spaced dyadic data and cannot be used for irregularly spaced data, which must be decomposed using special procedures. Projection of the irregular data onto a regular grid enables the use of the efficient FWT algorithm. Such procedures, however, do not deal with the original data set and their performance is sensitive to the type of projection employed. The lifting scheme, using data adapted second generation wavelets, can deal with irregular data directly but the basis functions employed are no longer simple dilates and translates of a mother wavelet. In addition, the extension of the second generation lifting scheme to two and higher dimensions is not straightforward. The least squares wavelet decomposition method offers a procedure that retains the simplicity of the first generation wavelets, deals with the original irregularly spaced data directly and is readily extended to higher dimensions.

The properties of the least squares wavelet decomposition was thoroughly investigated in this chapter in terms of its reconstruction capability. The assumption of discrete orthogonality is inherent in the commonly employed level by level least squares wavelet decomposition. Discrete orthogonality is lost with irregularly spaced data, even when the wavelet system employed is itself strictly orthogonal. In particular, it was clearly demonstrated that the commonly employed level by level
Pattern of irregularly (random) spaced samples in 2-d input domain

True function

Reconstructed function

Figure 4.28 Two dimensional simultaneous least squares wavelet reconstruction from an irregular 2-D data set
least squares wavelet decomposition can suffer from gross interpolation error in the case of strongly irregular data sets. To our knowledge, this has not been pointed out in previous literature (Bakshi and Stephanopoulos 1993, Safavi, 1995, Ford and Etter, 1998), most probably because the data sets employed were only slightly irregular. The gross interpolation error tendency of the level by level least squares wavelet decomposition can seriously compromise the quality of the reconstruction achieved.

An alternative formulation of the least squares wavelet decomposition, where all the scaling and wavelet coefficients are determined simultaneously rather than level by level, was developed in this chapter. The simultaneous least squares wavelet decomposition does not suffer from gross interpolation error in the case of highly irregular data sets. Its performance at the higher resolutions, however, is sensitive to the wavelet selection criteria employed. In cases where the ultimate objective is accurate function reconstruction, a hybrid decomposition can be employed to combine the high (frequency) resolution stability of the level by level decomposition with the low (frequency) resolution accuracy of the simultaneous decomposition. For denoising applications, however, the simultaneous least squares wavelet decomposition is preferable. This is because noise is usually high frequency and the denoising algorithm is aimed at removing the high frequency noise. This should at the same time reduce the sensitivity of the simultaneous least squares wavelet decomposition to erroneous wavelet selection. In the next chapter we shall combine the simultaneous least squares wavelet decomposition with a new data domain denoising technique to handle irregular data sets contaminated with Gaussian measurement error.
Chapter 5

Denoising Based on Local Goodness of Fit (LGF): A Method for Irregular Data with Gaussian Noise

5.1 Introduction

Real data is inevitably contaminated with measurement error and is corrupted by some level of uncertainty or noise. Recovering the true underlying function from a noisy data set is an essential step in signal processing, image processing, modelling of data, identification and control etc. Regardless of the method employed, estimating the true function from noisy samples involves a trade-off between two conflicting measures:

1- Agreement between data and the estimated function (sharpness).
2- Smoothness or stability of the solution.

![Figure 5.1 Trade-off between agreement (sharpness) and smoothness (stability)](image)

In the statistics literature, the compromise between sharpness and smoothness is referred to as the trade-off between bias and variance of the estimation. A sharp
estimator has a small bias but a large variance while a smooth estimator gives a large bias but a small variance (Ogden 1997, Press et al 1992).

Denoising is closely related to the general function estimation problem and is a key task addressed in parametric statistical modelling (i.e. nonlinear regression) and in the nonparametric learning theories of neural networks. In the function estimation framework, noise reduction reduces to the selection of an appropriate number of suitable basis functions as a means of controlling the complexity of the estimated function. Selection of a large size function space (large number of basis functions) leads to over-fitting and results in a noisy estimation. The objective of denoising procedures is to control the complexity of the approximating function so that it captures the underlying true function but rejects the noise.

The critical role of a structured function space to assist in complexity control was highlighted in Chapter 2. The combination of wavelets and multiresolution analysis provides a structured space ideally suited for function estimation from noisy data. It is therefore not surprising that multiresolution wavelet decomposition has led to the development of many fast and powerful techniques for denoising of uniformly spaced data with Gaussian noise. Such techniques take full advantage of the “locality and orthogonality” of wavelet systems, which in the case of regular data lead to a sparse set of independent and identically normally distributed wavelet coefficients. This enables the denoising to be carried out using simple but effective procedures in the wavelet domain. A simple and powerful method for denoising is to shrink the wavelet coefficients smaller than a specified threshold to zero. This approach is commonly referred to as the wavelet shrinkage or the wavelet thresholding method (See Donoho and Johnstone, 1994, 1995). Selecting an appropriate threshold value is a fundamental issue in wavelet-based denoising methods and the methods commonly used will be reviewed in this chapter.

The conventional thresholding mechanisms, developed primarily for uniformly spaced data with Gaussian noise, do not perform satisfactorily for irregular data. This is primarily because of the inevitable loss of (discrete) orthogonality of the wavelet systems associated with irregular data, which results in a set of wavelet coefficients that are neither independent nor identically distributed. Complex and
coefficient dependent thresholding mechanisms are then needed to (partially) allow for the fact that the wavelet coefficients are no longer i.i.d. (Kovac and Silverman 2000).

The first part of this chapter is devoted to a review of traditional denoising procedures developed for regular data and their limitation in the case of irregular data. The obvious approach for dealing with irregular data is to project the data onto a regular grid and employ denoising methods developed for regular data. The projection method is well developed by Kovac and Silverman (2000) and will be reviewed in some detail to highlight its limitations. An alternative approach for irregular data is to perform the denoising in the original data domain rather than in the wavelet domain. In this chapter, we propose a new denoising method for irregular data with Gaussian noise based on a Local Goodness of Fit (LGF) criterion. The LGF denoising procedure is carried out entirely in the data domain but takes full advantage of the locality of the multiresolution basis functions in both space and frequency. The fact that the multiresolution basis functions have local support is sufficient to allow for complexity control by checking a local measure of the goodness of fit. Furthermore, the LGF denoising procedure is readily combined with the least squares wavelet decomposition method described in Chapter 4 to form a computationally efficient algorithm. The combination of the LGF criterion and the least squares wavelet decomposition is readily extended to higher dimensions and is also capable of handling spatially variable Gaussian noise. The case of irregular data contaminated with non-Gaussian noise will be covered in the next chapter.

5.2 Wavelet Domain Denoising of Regular Data: A Review

Wavelet-based denoising is in general based on selective wavelet reconstruction. The wavelet transform of the noisy data set is first computed to determine the wavelet coefficients of the expansion. A number of the coefficients (basis functions) are then selected and the inverse transform is taken using only the selected coefficients to reconstruct the function. In the simplest approach, the wavelet coefficients belonging to high resolutions are simply left out of the expansion. This is
a “linear” approach in the sense that the coefficients included in the reconstruction are selected solely on the basis of their frequency contents (resolution) independent of the amplitude of the wavelet coefficients. This approach has been used for both regularly and irregularly spaced data (Antoniadis, et al 1994) but is surpassed by the nonlinear selection techniques first put forward by David Donoho and Ian Johnstone (1994, 1995). Instead of using the first few (say 100) coefficients, we could instead use the largest few (say 100) coefficients to perform the inverse wavelet transform. Figure 5.2 compares the reconstruction of the Blocks function using wavelet based linear (100-first coefficients) and nonlinear (100-biggest coefficients) approaches from 1024 samples using the Daub4 wavelet. Note that the nonlinear approximation has a much smaller sum of squared error (SSE=12) compared to the linear approximation (SSE=177).

![Wavelet reconstruction of the Blocks function using linear (100-first coefficients) and nonlinear (100-biggest coefficients) approaches](image)

**Figure 5.2** Wavelet reconstruction of the Blocks function using linear (100-first coefficients) and nonlinear (100-biggest coefficients) approaches

The nonlinear approach, originally developed by David Donoho and Ian Johnstone in a series of seminal papers (See Donoho and Johnstone 1992, 1994, 1995,
Donoho 1992, and Donoho et al 1993), has received a great deal of attention in statistics, function estimation, signal processing and modelling. It is the backbone of almost all wavelet denoising and compression techniques commonly employed for regular data and will be considered in some detail. Empirical data is usually contaminated by some amount of noise or uncertainty:

\[ y_i = f(x_i) + \epsilon_i \quad \text{for } i=1, \ldots, M \]  

Here \( f(x_i) \) is a sample of the true function and \( \epsilon_i \) is its additive noise. In order to estimate the true function \( f(x) \), it is necessary to remove or reduce the noise effects. The noise \( \epsilon \) can be considered as an unknown random variable but its descriptive statistics can not be determined without further assumptions. It is customary in statistics to assume a zero mean normal (Gaussian) distribution for the random noise. The variance of the noise is then either given a priori or it may be estimated from the residuals between an estimate of the true function and the measured data. The popularity of the normal distribution, besides its mathematical simplicity, is based on the central limit theorem of statistics: “the probability distribution of the sum of a very large number of small random fluctuations almost always converges to a normal (Gaussian) distribution”. In this chapter, we shall assume that the measurement noise is Gaussian and defer the more complex case of non-Gaussian noise to the next chapter.

To justify the wavelet domain denoising approach, we start with the wavelet decomposition of the true function \( f(x) \). The inaccessible wavelet coefficients \( d_{j,k} \) of the unknown true function \( f(x) \) are given by:

\[ d_{j,k} = \int_0^1 f(x) \psi_{j,k}(x) \, dx \]  

(5.2)

where \( \psi_{j,k}(x) \) is the dilated and translated version of a particular mother wavelet \( \psi(x) \). For a discrete set of \( M = 2^{J_{\text{max}}} \) equally spaced samples of the function \( f(x) \) on the interval [0, 1],

\[ x = \left( \frac{1}{M}, \frac{2}{M}, \ldots, \frac{M}{M} \right)^T, \quad f = (f(x_1), \ldots, f(x_M))^T, \]  

(5.3)

the discrete wavelet coefficients \( d_{j,k} \) can be expressed in matrix form as:
\[ d = \frac{1}{\sqrt{M}} W f \]  

(5.4)

Here \( W \) is the \( M \times M \) orthogonal matrix associated with an orthogonal set of wavelets and the factor \( 1/\sqrt{M} \) is usually included to unify the matrix and integral representations of the wavelet transform. An important feature of most practical signals is that they can be represented by a relatively small number of wavelet coefficients; this is called the \textit{sparsity} property of the wavelet representation. The sparseness property is clearly illustrated in the left panel of Figure 5.3. This plot results from applying the discrete wavelet transform to a set of 1024 equally spaced samples from the \textit{Heavisine} function of Donoho and Johnstone (1994). Note that only a small percentage of the total wavelet coefficients have large amplitude and others are effectively zero.

\[ \text{Figure 5.3} \text{ The wavelet (Symlet8) coefficients of the Heavisine function and its noisy version.} \]
Next, let us turn our attention to the $M$ equally spaced but noisy samples of the Heavisine function shown on the right panel of figure 5.3. The noisy samples are given by:
\[ y_i = f(x_i) + \varepsilon_i \]
where $\varepsilon_i$ is a normal variable drawn independently from a normal distribution with zero mean and variance $\sigma^2$ (i.e. $\varepsilon_i \sim N(0, \sigma^2)$). The objective is to find an estimate of the true function $f(x)$ using the discrete wavelet transform (DWT). The empirical wavelet coefficients $d_{j,k}$ are obtained using the DWT from:
\[ d = \frac{1}{\sqrt{M}} W y \]  
It can be shown (Ogden 1997) that for an orthogonal wavelet transformation in the limit of large $M$, the coefficients $d_{j,k}$ are themselves independent and normally distributed with an expectation:
\[ E(d_{j,k}) = d_{j,k} \]  
and a variance
\[ \text{var}(d_{j,k}) \approx \frac{\sigma^2}{M} \]  
The difference between $d_{j,k}$ and the (inaccessible) wavelet coefficients of the true function $d_{j,k}$ can be represented as
\[ d_{j,k} - d_{j,k} = z_{j,k}, \]  
The collection of $z_{j,k}$ is an unobservable set of $M$ independent and normally distributed random variable with zero mean and variance $\sigma^2 / M$,
\[ z_{j,k} \sim N(0, \frac{\sigma^2}{M}) \]  
The observable wavelet coefficients $d_{j,k} = d_{j,k} + z_{j,k}$ can therefore be thought of as the true wavelet coefficients contaminated with normal noise with zero mean and variance $\sigma^2 / M$. It follows that for large $M$, the noise in the original data is spread out uniformly amongst all the wavelet coefficients. Consequently, the high amplitude wavelet coefficients of the true signal are affected insignificantly and do not differ much in the presence of noise. This is reflected in Figure 5.3 which compares the wavelet coefficients of the true and noisy Heavisine functions. This is the key point of
the nonlinear approach and allows the recovery of the true function by thresholding the wavelet coefficients of the noisy function.

5.2.1 Denoising of Regular Data by Wavelet Thresholding

The heuristic idea behind Donoho-Johnstone (1994) wavelet thresholding method is that every empirical wavelet coefficient contributes noise, but only a very few wavelet coefficients contribute signal (sparseness). In general, therefore, large empirical coefficients contain significant signal while small coefficients contain substantially noise. Donoho and Johnstone (1994) simply propose that only those coefficients with an absolute value larger than a specified threshold should be included for estimating the true function. This “keep or kill” wavelet selection method is called “hard” thresholding and can be written as:

\[
\hat{d}_{j,k} = \eta^H(d_{j,k}) = \begin{cases} 
  d_{j,k} & \text{if } |d_{j,k}| > \tau \\
  0 & \text{otherwise}
\end{cases}
\]  

where \( \tau \) is the threshold value. Hard thresholding often results in an estimated function which is not sufficiently smooth. As an alternative to hard thresholding a procedure referred to as “soft” thresholding has been proposed (Donoho and Johnstone 1994):

\[
\hat{d}_{j,k} = \eta^S(d_{j,k}) = \begin{cases} 
  d_{j,k} - \tau & \text{if } d_{j,k} > \tau \\
  0 & \text{if } |d_{j,k}| \leq \tau \\
  d_{j,k} + \tau & \text{if } d_{j,k} < -\tau
\end{cases}
\]  

Here the coefficients smaller than the threshold \( \tau \) are set to zero but the remaining coefficients are also reduced by amount of \( \tau \) to account for their noise content.

Besides the visual inspection of the function estimated by different denoising techniques we also require a simple quantitative measure for performance comparison. In synthetic examples, where the true function is known to us, we may simply use the True Mean Square Error (TMSE) between the estimated and the true functions:
Here $F(x, \hat{a})$ is the estimated function and $M^\mu$ is a sufficiently large number. Figure 5.4 compares the result of hard and soft thresholding methods for denoising of a Heavisine function with added Gaussian noise with $\sigma = 0.5$. Visually, the soft thresholding method leads to a smoother estimated function with a $\text{TMSE} = 0.0431$ and the hard threshold estimate, although less smooth, has a smaller $\text{TMSE} = 0.0365$.

![Figure 5.4](image_url)

**Figure 5.4** Hard and soft thresholding denoising of a Heavisine function employing the Daub4 wavelet with a threshold value $\tau = 1.12$

One of the most desirable properties of wavelet thresholding methods is an inherent property of spatial adaptivity. In estimating a general function, we may typically require a large amount of smoothness in relatively flat portions of the
function and less smoothing in areas where the function has fine scale features. A spatially adaptive estimation method is one with the ability to recognize and apply the appropriate amount of smoothness in various portions of the domain based on the data. Without a priori knowledge about the local smoothness of a function, such information is difficult to extract from a discrete and noisy data set. Donoho and Johnstone (1994) demonstrated that, given a priori knowledge of smoothness, the wavelet thresholding method is competitive with all other spatially adaptive techniques utilizing a priori information. More importantly, they concluded that multiresolution wavelet threshold estimators work almost as well in the absence of a priori knowledge on smoothness. It is this built-in spatial adaptivity of wavelet thresholding methods that sets them apart from other techniques and forms the basis for an extremely powerful tool in statistical function estimation.

The selection of a proper threshold value is a fundamental issue in the wavelet domain denoising procedure. A very small threshold will allow many coefficients to be included in the reconstruction giving a noisy estimate. On the other hand, a very large threshold results in over-smoothing. The threshold value determines the number of basis function required for the denoising purposes and its value should ideally be determined by minimization of an estimate of the prediction risk. Three general approaches for selecting the threshold values have been proposed: global thresholding, level dependent thresholding, and coefficient dependent thresholding, which will be considered in turn.

5.2.2 Global Thresholding

In global thresholding, a single threshold value $\tau$ is applied to all wavelet coefficients irrespective of the resolution. Donoho and Johnstone (1994) have proposed two heuristics for choosing the global threshold value based on the number of data points and the variance of noise. The first heuristic is based on the observation that for normally distributed samples with a standard deviation of unity

$$z_i \sim N(0,1), \quad i = 1, \ldots, M$$

(5.13)

the probability for $z_i < \sqrt{2 \log M}$ tends to unity as $M$ tends to infinity: 

$$\lim_{M \to \infty} P(z_i < \sqrt{2 \log M}) = 0$$
\[ P_{M \to \infty} \left| \epsilon_i < \sqrt{2 \log M} \right|, \text{ for all } i = 1, \ldots, M \to 1 \] (5.14)

Now, the noise in the wavelet coefficients is i.i.d. distributed as
\[ z_i \sim N(0, \sigma^2 / M), \] (5.15)
we may therefore take
\[ \tau = \sqrt{2 \log M} \frac{\sigma}{\sqrt{M}}, \] (5.16)

This threshold, which is referred to as the \textit{universal threshold}, ensures that the noise in a wavelet coefficient can not be larger than this threshold. Furthermore, all coefficients larger than \( \tau \) are not due to the noise alone and contain some element of the signal.

An alternative \textit{minimax thresholding} heuristic is obtained by minimization of the constant term in an upper bound of the \( L^2 \) risk involved in estimating a function (Donoho and Johnstone, 1994). This heuristic cannot be stated in closed analytic form but can be approximated numerically for a given sample size and unit variance. The minimax threshold values for various sample sizes are presented in Table 5.1 (Donoho and Johnstone, 1992).

### Table 5.1 Minimax values for various sample size and unit noise variance.

<table>
<thead>
<tr>
<th>Sample Size (M)</th>
<th>64</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
<th>8192</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimax Threshold</td>
<td>1.474</td>
<td>1.669</td>
<td>1.860</td>
<td>2.047</td>
<td>2.232</td>
<td>2.414</td>
<td>2.594</td>
<td>2.773</td>
</tr>
</tbody>
</table>

For a given sample size \( M \), the universal threshold value is larger than the minimax threshold value and results in a less noisy and visually more appealing estimate. For this reason global thresholding using the universal threshold heuristic is frequently referred to as the \textit{"VisuShrink"} method. The denoising of a Blocks function...
It has become customary to threshold only the coefficients belonging to resolutions higher than a predetermined resolution $J_T$. The lower level coefficients are included in the reconstruction without thresholding because they correspond to the coarse contents (the smooth components) of the signal. The choice of $J_T$ can have a significant effect on the estimated function, especially when the signal to noise ratio is
small. Figure 5.6 shows the effect of choosing different $J_{r}$ values on the denoising of the Blocks function. Finding the proper resolution $J_{r}$ is a difficult task and no totally satisfactory method has been proposed (Akbaryan et al, 2000).

![Denoising plots](image)

**Figure 5.6** Denoising of a blocky function starting the global thresholding at different resolutions, $J_{r}=0$, and $J_{r}=7$

### 5.2.3 Level Dependent Thresholding

A criticism that may be levelled against global thresholding is that the same threshold $\tau$ is used at all resolutions. Better denoising performance can possibly be achieved by using a different threshold value $\tau_{j}$ at different resolutions. The selection of the level dependent threshold values may be based on the amplitude of the wavelet coefficients at each resolution. A variety of level dependent threshold selection
methods have been proposed based on hypothesis testing (Abramovich and Benjamini 1995, Ogden and Parzen, 1996-a, 1996-b), cross validation (Nason, 1994, 1996, 2002), Bayesian methods (Vidakovic, 1994, Chipman et al, 1995, Barber et al, 2000) and the Stein Unbiased Risk Estimate (SURE) and are well reviewed by Ogden (1997). In this section we focus on the popular and frequently employed level dependent SURE threshold originally developed by Donoho and Johnstone (1995).

The starting point for developing the SURE level dependent technique is the minimization of the $L^2$ risk functional

$$ R(\hat{f}, f) = \mathbb{E} \left[ \frac{1}{M} \sum_{i=1}^{M} (\hat{f}(x_i) - f(x_i))^2 \right] $$

(5.17)

where $\hat{f} = F(x, a)$ is the estimate of the true function $f(x)$. Using Parseval's theorem in the wavelet domain, we may express the above functional in terms of the wavelet coefficients

$$ R(\hat{f}, f) \approx \mathbb{E} \left[ \sum_{j} \sum_{k} (\hat{d}_{j,k} - d_{j,k})^2 \right] $$

(5.18)

In other words, we can attempt to minimize the risk functional in the wavelet domain rather than in the data domain. The appropriate threshold at each resolution $j$, $\tau_j$, is obtained by considering each resolution independently and separately. At each resolution $j$, we have the noisy coefficients

$$ d_{j,k} = \hat{d}_{j,k} + z_{j,k} \quad k = 1, \ldots, N_j $$

(5.19)

and we wish to estimate the true coefficients $d_{j,k}$. For an orthogonal wavelet system, we know that the noisy wavelet coefficients are distributed as

$$ d_{j,k} \sim N(d_{j,k}, \sigma^2 / M) $$

(5.20)

Now, let the estimate of the true coefficient $d_{j,k}$ be given in terms of the soft thresholding function $\eta_s^r(d_{j,k})$

$$ \hat{d}_{j,k} = \eta_s^r(d_{j,k}) $$

(5.21)

where $\eta_s^r(d_{j,k})$ is defined as:
If we let \( g(d_{j,k}) = \eta^*_\delta(d_{j,k}) - \hat{d}_{j,k} \) and write equation (5.21) as

\[
\hat{d}_{j,k} = d_{j,k} + g(d_{j,k})
\]

we can use the work of Stein (1981) to show that unbiased estimate of the true mean squared error is given by

\[
\text{SURE}(\tau_j, \hat{d}_j) = E(\|\hat{d}_j - d_j\|^2) = N_j + E\|g(d_{j})\|^2 + 2\{g(d_{j})\}
\]

(5.23)

After some algebraic manipulation (Donoho and Johnstone 1995), it can be shown that the SURE expectation may be stated in a compact form as:

\[
\text{SURE}(\tau_j, \hat{d}_j) = N_j - 2\{(k : |d_{j,k}| \leq \tau_j) + \sum_{k=1}^{N_j} \min\{d_{j,k}, \tau_j\}\}
\]

(5.24)

where \( N_j \) is the number of wavelet coefficients at resolution \( j \), and \#\{\} denotes the cardinality of the set \{\}. The threshold \( \tau_j \) is simply chosen as the value of \( \tau \) which minimizes the SURE function:

\[
\tau_j = \arg\min_{\tau \geq 0} \text{SURE}(\tau, \hat{d}_j)
\]

(5.25)

The popularity of the SURE method lies partly in the fact that the minimization of the fearsome looking SURE function is actually quite straightforward.

In practice, the SURE threshold values are less than the universal threshold value for the same data set. In addition for a sufficiently dense coefficient set, the SURE threshold decreases to zero at low resolutions. As a result, the coefficients selected based on the SURE threshold may include many small (and possibly noisy) coefficients and the estimated function is sharper but more oscillatory than the universal thresholding estimate. This makes the SURE method better suited for denoising of sharp signals and images. The SURE method does not perform well at any level where the wavelet representation is sparse and all but a few coefficients are
(essentially) zero. Donoho and Johnstone (1995) suggest a “hybrid” method to handle this issue. In the hybrid method a test of sparsity is performed first at each level and the universal rather than SURE threshold is used for the levels that are recognized as sparse. The SURE criterion is used only for computing the threshold value where sparsity does not present a difficulty. Figure 5.7 compares denoising of the Heavisine function using the global universal thresholding with level by level SURE thresholding. The SURE estimate is clearly sharper and has a smaller TMSE but exhibit more spurious oscillations (noisy spikes).

Figure 5.7 Comparison of denoising of the Heavisine function by universal soft thresholding and level dependent SURE thresholding, \( j_0 = J_T \)
5.2.4 Coefficient Dependent Thresholding

The hard and soft universal thresholding and the level by level SURE thresholding schemes share a number of basic assumptions:
- regularly spaced data,
- an orthogonal wavelets system, and
- independent and identically distributed (i.i.d.) Gaussian noise.

Such assumptions ensure that the true wavelet coefficients are contaminated with i.i.d. normal noise. In particular, the variance of the wavelet coefficients is the same and the simple thresholding techniques are effective. In many cases, however, the signal may be contaminated with spatially variable (non-stationary) Gaussian noise. Johnstone and Silverman (1997) have argued that the simpler thresholding techniques are inappropriate for such cases because the wavelet coefficients do not have identical variances.

An approach for dealing with spatially variable noise, implemented in the MATLAB wavelet toolbox, based on the work of Levielle (1999), involves dividing the signal into smaller intervals each with a different but constant noise variance. Standard thresholding techniques could then be applied to each interval separately. The inherent difficulty with this approach lies in estimating the appropriate number and width of the intervals, which is a difficult optimization problem. Kovac and Silverman (2000) have proposed an alternative method where each wavelet coefficient $d_{j,k}$ is individually thresholded in proportion to its own standard deviation $\sigma_{j,k}$. The standard deviations of the wavelet coefficients is readily obtained through the filter bank analysis employed for computing the discrete wavelet transform. We start by setting the highest resolution scaling coefficients (resolution $L$) to the original data,

$$\xi_L = y,$$  \hspace{1cm} (5.26)

and assign the known (or estimated) covariance matrix of the data $\Sigma_y$ to these coefficients

$$\Sigma_x^L = \Sigma_y,$$ \hspace{1cm} (5.27)
The scaling and wavelet coefficients and their associated covariance matrices (see Johnson and Wichern, 1982) at other resolutions are then obtained by the recursive application of:

\[ e_{j-1} = H_j e_j, \quad \Sigma^e_{j-1} = H_j \Sigma^e_j (H_j)^T \]  
\[ d_{j-1} = G_j e_j, \quad \Sigma^d_{j-1} = G_j \Sigma^d_j (G_j)^T \]  
(5.28)

where \( H_j \) and \( G_j \) are the low and band pass filter matrices corresponding to resolution \( j \). The variance of the wavelet coefficients at resolution \( j \) are then simply the diagonals of the covariance matrices \( \Sigma^e_j \) and \( \Sigma^d_j \). Once the standard deviations \( \sigma_{j,k} \) are determined, each coefficient may be thresholded using a threshold similar to that used in the standard universal thresholding method,

\[ \tau_{j,k} = \hat{\sigma}_{j,k} \sqrt{2 \log M} \]  
(5.30)

Alternatively, we may apply the SURE criterion to each individual coefficient using

\[ SURE(\tau, d) = \sum_{j,k} [\hat{\sigma}_{j,k}^2 - 2\hat{\sigma}_{j,k}^2 \# \{k : |d_{j,k}| \leq \tau \} + \min^2 (|d_{j,k}|, \tau_j)] \]  
(5.31)

\[ \tau_{j,k} = \hat{\sigma}_{j,k} \times \arg \min_{t \geq 0} SURE(t, d) \]  
(5.32)

Figure 5.8 compares the performance of the standard universal hard and soft thresholding methods with the coefficient dependent method of Kovac and Silverman (2000). The test function is the Heavisine function contaminated with a normal noise whose standard deviation varies spatially in proportion to the square of the distance from the origin,

\[ \sigma_i = x_i^2, \quad x_i = \frac{i}{2^{10}}, \quad i = 1, ..., 2^{10} \]  
(5.33)

The significantly improved performance of the coefficient dependent thresholding is self evident.
Figure 5.8 Comparison of global and coefficient dependent hard and soft thresholding of the Heavisine function with spatially variable noise using Daub4 wavelet.

5.3 Wavelet Domain Denoising of Irregular Data

In many practical applications, the sampling of the data is spatially irregular. Three methods were discussed in Chapter 4 for wavelet approximation of irregularly sampled data: the indirect projection method of Kovac and Silverman (2000) and the direct methods based either on second generation wavelets (the irregular lifting scheme), or the least squares wavelet decomposition method. Development of
denoising procedures based on second generation wavelets is not considered in this study for two reasons. First, second generation wavelets are not simple dilates and translates of each other, which calls for complex and sophisticated thresholding procedures. Second, the mathematical definition of second generation wavelets for irregular data in two and higher dimensions is still in its infancy and not sufficiently well-developed (see Jansen and Bultheel, 1998, 1999, Simoens and Vandewalle, 2000-a, 2000-b, Delouille et al, 2001-a, 2004, Vanraes et al, 2002). We shall therefore concentrate on denoising of irregular data based on the combination of wavelet thresholding with either the projection method or the least squares wavelet decomposition technique.

5.3.1 Projection and Wavelet Thresholding

For denoising of irregular data we may first project the data onto a regular grid, then take the wavelet transform of the regular but projected data and finally apply any of the thresholding procedures discussed above. However, the denoising performance obtained is open to question and only acceptable in special situations. This is primarily because we no longer deal with the original data set directly. The coefficient dependent thresholding method of Kovac-Silverman (2000) developed for dealing with spatially variable noise is in theory well suited for extension to the irregular data setting.

The original irregular data $\mathbf{y}$ is projected onto a uniform grid to produce a regular data set $\mathbf{y}'$:

$$
\mathbf{y}' = \mathbf{R} \mathbf{y}
$$

(5.34)

where $\mathbf{R}$ represents the projection matrix. For an irregular data set with i.i.d. normal noise, the covariance matrix is $\sigma^2 \mathbf{I}$ and we can estimate the covariance matrix of the projected regular data as

$$
\Sigma' = \sigma^2 \mathbf{R} \mathbf{(R')^T}
$$

(5.35)
However, $\Sigma''$ is no longer diagonal since the projection matrix $R''$ is not necessarily orthogonal. A similar situation is encountered in dealing with regular data with spatially variable and correlated noise (Kovac and Silverman, 2000). It is sufficient to set the covariance matrix of the scaling coefficients at the highest resolution $L$ to that of the projected data $\Sigma'' = \Sigma'', we can then use the coefficient dependent thresholding technique described in section 5.2.4 with no further modification.

Figure 5.9 compares the denoising performance of universal and coefficient dependent thresholding for a set of 1024 randomly spaced samples of the Heavisine function contaminated with i.i.d. Gaussian noise with a spread of $\sigma = 0.5$. The irregular data set is first linearly interpolated onto a regular grid to produce 1024 uniformly spaced samples before taking the wavelet transform. It is surprising that the simple universal thresholding performs better than the coefficient dependent thresholding in terms of both the number of basis functions retained (i.e. complexity) and the TMSE. A small simulation was conducted to ensure that this is not accidental and confined to a particular example. Table 5.2 summarizes the average statistics obtained on repeating the above exercise 100 times for different randomly spaced noisy data sets. The statistics obtained confirm that for the denoising of the Heavisine function the simpler universal thresholding method gives a less complex estimate with a smaller TMSE.

**Table 5.2** Average statistics (100 repetitions) for denoising of 1024 irregular samples of a Heavisine function using symlet8 wavelet for linear projection coupled to different thresholding methods

<table>
<thead>
<tr>
<th>Thresholding Method</th>
<th>$N$ (average)</th>
<th>TMSE (average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal (hard)</td>
<td>21</td>
<td>0.0311</td>
</tr>
<tr>
<td>Coefficient Dependent (hard)</td>
<td>61</td>
<td>0.0451</td>
</tr>
<tr>
<td>Universal (soft)</td>
<td>21</td>
<td>0.0363</td>
</tr>
<tr>
<td>Coefficient Dependent (soft)</td>
<td>61</td>
<td>0.0429</td>
</tr>
</tbody>
</table>
Figure 5.9 Comparison of universal and coefficient dependent thresholding methods for linear projection of randomly spaced data using Symlet8 wavelets

No definite general conclusion can be drawn on the basis of a single example. It may well be possible to construct other examples where the coefficient dependent thresholding technique has a better (average) performance compared to the simpler universal thresholding methods. In particular, performance of the coefficient dependent thresholding method is intimately linked to the type of projection used and the form of the basis function employed. We also note that the coefficient dependent thresholding method proposed by Kovac and Silverman (2000) is based on the variance of the wavelet coefficients and does not use the available covariance information. Another difficulty is associated with extending the projection method to two and higher dimensions. The most severe criticism of projection methods,
however, lies in the fact that we no longer deal with the original data set. In our opinion a more reliable denoising technique should deal with the irregular data set directly. The least squares wavelet decomposition technique offers such an opportunity and is considered next. To our knowledge, this approach has not been previously reported in the literature.

5.3.2 Least Squares Decomposition and Wavelet Thresholding

Consider a data set with \( i.i.d \) Gaussian noise of variance \( \sigma^2 \), the determination of the wavelet coefficients by least squares decomposition involves the minimization of

\[
\mathcal{R}_{\text{emp}}(d) = (y - Ad)^T (y - Ad)
\]

where \( A \) is the design matrix with elements \( A_{ij} = \theta_j(x_i) / \sigma \). This minimization reduces to the solution of the linear system

\[
(A^T A) d = A^T y,
\]

and the covariance matrix of the computed coefficients \( d \) is given by

\[
\Sigma^d = (A^T A)^{-1}
\]

In the case of a regular data set and an orthogonal wavelet system, the matrix \( A \) is orthogonal and \( (A^T A) = \frac{M}{\sigma^2} I \) so that

\[
\Sigma^d = \frac{\sigma^2}{M} I
\]

In other words, the coefficients are \( i.i.d \) distributed with variance \( \sigma^2 / M \). For an irregular data set, however, the matrix \( A \) is no longer orthogonal. Therefore, the diagonal elements of the covariance matrix \( \Sigma^d \) are not equal and its non-diagonal elements may be non-zero. This means that the wavelet coefficients are correlated and do not have the same variance: the variance of a wavelet coefficient located in a dense data region is smaller than that of a coefficient located in a sparse data region. Coefficient dependent thresholding, which is based on the variance of the individual coefficients, can therefore be naturally coupled to the least squares decomposition.
For comparison purposes, the Heavisine example used in the previous section was denoised using the least squares decomposition and wavelet thresholding. The results for universal and coefficient dependent thresholding are compared in Figure 5.10. The statistics for a small simulation (involving 100 repetitions) is also shown in Table 5.3. It is clear that in this case the complexity of the estimated function (i.e. the number of basis function retained) is practically the same for either thresholding method. The difference in the TMSE obtained for various methods is also smaller when compared to similar data based on projection, see Table 5.2. The advantage of not manipulating the original data is brought out by comparing the statistics in Table 5.2 and 5.3: the complexity and the TMSE for coefficient dependent hard thresholding using the least squares decomposition (i.e. N=17 and TMSE=0.03) compares favourably with those obtained using linear projection (i.e. N=51 and TMSE=0.04).

Figure 5.10 Comparison of universal and coefficient dependent thresholding methods for least squares decomposition of randomly spaced data using symlet8 wavelet
Table 5.3 Average statistics (100 repetitions) for denoising of 1024 irregular samples of a Heavisine function, using least squares decomposition with Symlet8 wavelets and different thresholding methods

<table>
<thead>
<tr>
<th>Thresholding Method</th>
<th>N (average)</th>
<th>TMSE (average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal (hard)</td>
<td>18</td>
<td>0.0349</td>
</tr>
<tr>
<td>Coefficient Dependent (hard)</td>
<td>16</td>
<td>0.0390</td>
</tr>
<tr>
<td>Universal (soft)</td>
<td>18</td>
<td>0.0697</td>
</tr>
<tr>
<td>Coefficient Dependent (soft)</td>
<td>16</td>
<td>0.0928</td>
</tr>
</tbody>
</table>

Irrespective of the decomposition method employed, denoising in the wavelet domain using current thresholding algorithms has a major limitation in the case of irregular data. The information about the noise is implicitly transferred into the covariance matrix of the wavelet coefficients. In applying the thresholding methods, advantage is taken of the variance of the coefficients but no use is made of the covariance information. To improve the reliability of the denoising procedure, we should ideally make use of the full covariance structure of the wavelet coefficients. This may be possible within the wavelet domain but leads inevitably to increasingly more complex and cumbersome thresholding algorithms. It is in some ways easier to take full advantage of the noise information by performing the denoising operation in the data domain. The remainder of this chapter is devoted to the development of a new data domain denoising technique and its comparison with available methods.

5.4 Data Domain Denoising: A New Method Based on the Local Goodness of Fit (LGF)

In wavelet domain denoising procedures a threshold value is applied to the wavelet coefficients to determine the number of basis functions retained for estimating the denoised function. Thresholding methods work well in cases where sampling is uniform but, in our experience, do not perform reliably in the case of irregular data. In this section, we develop a new denoising procedure which operates entirely in the data domain but takes advantage of the multiresolution structure and the properties of wavelets. To perform the calculations entirely in the data domain, we require some means of controlling the complexity of the estimated function. The
denoising method developed below uses the classical concept of “goodness of fit” applied “locally”. We refer to this method as the local goodness of fit denoising method or the LGF method for short.

5.4.1 The Local Goodness of Fit (LGF)

The goodness of fit is a measure traditionally used to test the validity of an assumed model in parametric modelling as discussed in Chapter 2. In denoising applications the form of the model is not known a priori but the goodness of fit criterion can be employed in a constructive manner. In other words, we can use the goodness of fit criterion to select a set of particular basis functions in order to control the complexity of the estimated function. The locality of first generation wavelets and the inherent characteristics of a regular multiresolution structure enable us to use a local measure of the goodness of fit for constructing the estimate of the true function. Consider the structure of a multiresolution wavelet system shown in Figure 5.11. We shall refer to each tile within this structure as a block addressed by its dilation \( j \) and translation \( k \), \( B(j, k) \). Each block \( B(j, k) \) has an associated basis function \( \psi_{j,k}(x) \), whose effective support is the width of the block.

![Figure 5.11 Blocks (B) identified on a dyadic multiresolution structure](image)

138
In order to define a local measure of goodness of fit, we shall assume that an estimate of the function at resolution \( j, F_j(x) \), is available. Let \( M_L(j, k) \) denote the number of data points falling in the block \( B(j, k) \) and define the following vectors:

- \( x_{j,k} \) = abscissa of data within block \( B(j, k) \)
- \( y_{j,k} \) = ordinate of data within block \( B(j, k) \)
- \( \sigma_{j,k} \) = standard deviation of noise for data within block \( B(j, k) \)
- \( \hat{y}_{j,k} \) = vector of estimates within block \( B(j, k) \) at resolution \( j \)
  
  i.e. \( \hat{y}_{j,k}(i) = F_j(x_i) \) for \( x_i \in x_{j,k} \)

The local chi-square within block \( B(j, k) \) can be computed from:

\[
\chi^2_L(j, k) = \sum_{i=1}^{M_L(j, k)} \left( \frac{y_{j,k}(i) - \hat{y}_{j,k}(i)}{\sigma_{j,k}(i)} \right)^2
\]

(5.40)

To develop a local estimate for the goodness of fit we shall assume that \( \chi^2_L(j, k) \) obeys a chi-square distribution with \( \nu_L(j, k) \) local degrees of freedom given by

\[
\nu_L(j, k) = M_L(j, k) - N_L(j, k)
\]

(5.41)

where \( N_L(j, k) \) is the effective number of basis functions which have contributed to the estimate \( F_j(x) \) within block \( B(j, k) \). We note that all basis functions, starting from the lowest resolution \( J \) up to and including resolution \( j-1 \) can make a contribution to the estimated function at resolution \( j \). For the regular multiresolution structure of Figure 5.11, \( N_L(j, k) \) is simply given by

\[
N_L(j, k) = \frac{\text{total number of basis functions up to level } j-1}{\text{Number of blocks at level } j} = \frac{2^j - 1}{2^j} = 1
\]

(5.42)

and the local degree of freedom is therefore,

\[
\nu_L(j, k) = M_L(j, k) - 1
\]

(5.43)

A local measure of the goodness of fit \( q_L(j, k) \) at block \( B(j, k) \) is obtained by evaluating the complement of the incomplete Gamma function \( Q(.) \) as (see equation 2.52):
\[ q_{L}(j,k) = Q\left(\frac{\sigma_{L}(j,k)}{2}, \frac{\chi^{2}_{L}(j,k)}{2}\right) \] (5.44)

and may be used to accept or reject the local estimate within block \( B(j, k) \). A very low value of \( q_{L}(j,k) \rightarrow 0 \) signifies local over-smoothing and indicates the data in block \( B(j, k) \) can support a higher resolution. A very high value of \( q_{L}(j,k) \rightarrow 1 \) signifies local over-fitting and the data in block \( B(j, k) \) can not support resolution \( j+1 \). The local goodness of fit criterion can therefore be used to control the local complexity of the estimated function with all the calculations performed entirely in the data domain. This opens the way for a unified treatment of regularly and irregularly spaced data.

### 5.4.2 The LGF algorithm

We are now in a position to present the LGF denoising algorithm, which is summarized in Figure 5.12. To apply this algorithm we must first construct the multiresolution structure shown in Figure 5.11 and populate it with a particular wavelet system. There is no restriction in the choice of the wavelet system other than they must be dyadic dilation and integer translation of a mother wavelet. The key property exploited is locality of the basis functions. Strict orthogonality, although helpful, is not essential. We also note that unlike the wavelet domain thresholding methods, all calculations are performed entirely in the data domain and the LGF algorithm is equally applicable to uniformly and non-uniformly spaced data. Given a populated multiresolution structure and a data set \((x, y)\) contaminated with Gaussian noise with spread \( \sigma \), the first step is to calculate the scaling coefficients \( c \) and wavelet coefficients \( d \) using any suitable decomposition method. The LGF algorithm is in fact independent of the method used to calculate the wavelet coefficients. For example, with regularly spaced data we can take advantage of the highly efficient FWT decomposition. For irregularly spaced data we can either use projection methods or resort to the least squares wavelet decomposition. As we shall see later, the LGF denoising method is particularly well matched with the least squares wavelet decomposition discussed in section 4.4.
Figure 5.12 Summary of the LGF denoising algorithm

- Set a threshold value for the LGF criterion $q_{low}$
- Initialize by deselecting all wavelets: $select(j, k) \leftarrow 0$ and $\hat{d}_{j,k} \leftarrow 0$ for all $j$ and $k$
- Select the wavelets at the coarsest resolution $J$ and find the coarsest estimate and detail
  $select(J, k) \leftarrow 1$ and $\hat{d}_{j,k} \leftarrow d_{j,k}$  $\forall k$
  $$F_j(x_i) = \sum_{k=1}^{k=2^j} c_{j,k} \varphi_{j,k}(x_i) \quad \text{and} \quad f_j(x_i) = \sum_{k=1}^{k=2^j} \hat{d}_{j,k} \psi_{j,k}(x_i)$$

  For $j = J: 1: L - 1$  
  * start of Resolution Loop

  - Find the estimate at level $j+1$
    $$F_{j+1}(x_i) = F_j(x_i) + f_j(x_i), \quad \text{TerminateFlag} \leftarrow 1$$
  For $l = 1: 1: 2^j$
    **start of Block Loop

    If $select(j, l) = 1$ Then
      For $k = 2l - 1, 2l$
        - cast the data to block $B(j+1, k)$:
          $$M_L(j+1, k) \leftarrow \text{number of data in } B(j+1, k)$$
          $$x_{j+1,k}, y_{j+1,k}, \sigma_{j+1,k} \leftarrow \text{data falling in } B(j+1, k)$$
          $$\hat{y}_{j+1,k} \leftarrow \hat{y}_{j+1,k}(i) = F_{j+1}(x_{j+1,k}(i)), \text{estimate in } B(j+1, k)$$
          - calculate the local variables in block $B(j+1, k)$
            $$\chi_L^2(j+1, k) \leftarrow \sum_{i=1}^{M_L(j+1, k)} \left( \frac{y_{j+1,k}(i) - \hat{y}_{j+1,k}(i)}{\sigma_{j+1,k}(i)} \right)^2$$
            $$v_L(j+1, k) \leftarrow M_L(j+1, k) - 1$$
            $$q_L(j+1, k) \leftarrow Q \left( \frac{v_L(j+1, k)}{2}, \frac{\chi_L^2(j+1, k)}{2} \right)$$
          - test the local goodness of fit in block $B(j+1, k)$
            if $q_L(j+1, k) < q_{low}$ then
              $select(j+1, k) \leftarrow 1, \hat{d}_{j+1,k} \leftarrow d_{j+1,k}$, and TerminateFlag = 0
            Endif
        Endfor
    Endif
  Endfor
  **end of Block Loop

- Test for TERMINATION
  If TerminateFlag=1 then
    TERMINATE
  Else
    $$f_{j+1}(x_i) = \sum_{k=1}^{k=2^j} \hat{d}_{j+1,k} \psi_{j+1,k}(x_i)$$
  Endif
**end of Resolution Loop

141
In a multiresolution structure, the estimate at resolution \( j+1 \), \( F_{j+1}(x) \), is built by adding the wavelets (detail) at resolution \( j \), \( f_j(x) \), to the estimate (coarse content) at resolution \( j \), \( F_j(x) \).

\[
F_{j+1}(x) = F_j(x) + f_j(x),
\]

(5.45)

The key idea behind the LGF denoising algorithm is to test the local goodness of fit \( q_L(j+1,k) \) within each block at resolution \( j+1 \) and terminating the estimation locally when appropriate. To test the LGF criterion in block \( B(j, k) \), we must first decide on an acceptable range for the local goodness of fit by specifying a probability \( q_{low} \) such that \( q_{low} \leq q_L \leq 1 - q_{low} \). This is in fact the only parameter of the LGF denoising method and its choice will be critically examined subsequently.

The LGF denoising procedure starts by reconstructing the function at the lowest resolution and testing the LGF criteria in all blocks. In the blocks where the LGF criterion is satisfied \( q_L \geq q_{low} \), the fit is deemed acceptable and the estimation procedure is terminated locally at that resolution. The next higher resolution basis functions will only be added in the blocks where the LGF criterion is not satisfied \( q_L < q_{low} \). This procedure proceeds to higher resolutions until the LGF become acceptable at all blocks. In general, the estimate of the function at lower resolutions is far from the true function, which results in a large \( \chi_L^2 \) and a small \( q_L < q_{low} \). The low resolution wavelets are therefore normally all selected and we could in fact start the reconstruction from a suitably chosen intermediate resolution with potential saving in computation. Note also that since the construction proceeds from low to high resolutions, we do not need to test the upper bound on the the LGF criterion \( q_L \), \( q_{low} \leq q_L \leq 1 - q_{low} \).

It may be worthwhile to present the LGF algorithm in a pseudo-code form. To this end we assign an indicator \( select(j, k) \) to each block \( B(j, k) \). This indicator is set to unity, \( select(j, k) \leftarrow 1 \), if the basis function associated with \( B(j, k) \), \( \psi_{j,k} \), is selected and to zero otherwise, \( select(j, k) \leftarrow 0 \). We also define a vector of selected wavelet coefficients \( \hat{d} \), which is identical to the full wavelet coefficient vector \( d \) except that
the coefficients corresponding to the non-selected basis functions are set to zero. Figure 5.12 summarizes the major steps in the LGF denoising algorithm in pseudo code form. The procedure is started by selecting all the scaling functions and wavelets at the coarsest resolution $J$ and deselecting all wavelets at higher resolutions. This is reasonable because the estimated function at the coarsest resolution is usually far from the true function, which results in a large value of $\chi^2$ and a very small $q_L < q_{low}$ within all blocks. The procedure proceeds up the resolution levels terminating locally within any block satisfying the lower bound on the LGF criterion. The entire procedure is terminated at a resolution where all the blocks meet the LGF criterion.

5.5 Results and discussion

5.5.1 Selection of the Probability Limit $q_{low}$

The probability limit $q_{low}$ is the only user defined parameter in the LGF denoising algorithm and its selection must be carefully considered. Figure 5.13 shows the performance of the LGF denoising algorithm for a noisy Heavisine function for a wide range of values from $q_{low} = 10^{-d}$ to $q_{low} = 0.5$. The data consist of 1024 regular samples contaminated with i.i.d. Gaussian noise with zero mean and standard deviation $\sigma = 0.5$. It is clear that for $q_{low} = 10^{-d}$ substantial over-smoothing of the true function (shown in red) occurs. At the other extreme, with $q_{low} = 0.5$, too many high resolution basis functions are selected and the noise is not filtered effectively. The true function is reconstructed well and the noise is effectively filtered for $q_{low} = 0.01$ and $q_{low} = 0.1$, in both cases the TMSE is around 0.02 and sharp local features are captured well. For this example therefore a suitable range for $q_{low}$ appears to be between 0.01 and 0.1.
Figure 5.13 The influence of the probability limit $q_{low}$ on the performance of the LGF denoising algorithm (Heavisine function and Symlet8 wavelet system)

From a conceptual viewpoint, the value of $q_{low}$ cannot be selected close to 0.5. Setting $q_{low} = 0.5$ implies that there is a 50% probability that we may reject the true function within a block and erroneously proceed to higher resolutions only to capture noise. From a global perspective, this means that with $q_{low} = 0.5$, we may over-fit in about 50% of the blocks at the highest resolutions. A lower value of $q_{low}$ must be used to increase the probability of accepting the true function in each block. Choosing a very low value, however, increases the probability of accepting many other (smoother) functions on a par with the true function. This would cause premature
termination of the LGF algorithm at a low resolution resulting in an over-smoothed estimate. Such heuristic arguments indicate that a suitable range for $q_{\text{low}}$ is between 0.001 and 0.1. To confirm this, the sensitivity of the LGF denoising algorithm to the value of $q_{\text{low}}$ was examined for a variety of test functions. Figure 5.14 shows noisy versions of the four most frequently used test functions employed in wavelet studies. Between them these test functions cover many of the features encountered in practical signals. In each case the data is 1024 regular samples contaminated with i.i.d. Gaussian noise with zero mean and $\sigma = 0.5$.

![Figure 5.14 Standard test functions used in wavelet studies](image)

A detailed simulation was conducted to denoise each of the above functions using the LGF algorithm for 20 different values of $q_{\text{low}}$ in the range $10^{-6}$ to 0.2. The estimation was repeated 100 times at each value of $q_{\text{low}}$ with different randomly selected Gaussian noise patterns to give a statistically reliable estimate of the (average) TMSE. Figure 5.15 shows a graph of the average TMSE versus $q_{\text{low}}$ for each of the four test functions. In all cases the curves show a shallow minimum in the range of 0.001 to 0.1, suggesting that a value of $q_{\text{low}} = 0.01$ is a good value for a wide
range of functions. The quality of the denoised test functions obtained using the LGF algorithm with $q_{low} = 0.01$ is shown in Figure 5.16. The performance for the Blocks function could be substantially improved if the Haar wavelet system is used instead of the Symlet8 system. Similarly, the sampling frequency (1024 points) limits the performance for the Bump and Doppler functions.

**Figure 5.15** The average TMSE as a function of $q_{low}$ for test functions of Figure 5.14

**Figure 5.16** Typical examples of the test functions denoised by the LGF technique with $q_{low} = 0.01$ (1024 uniform samples, Least squares wavelet decomposition with Symlet8)
The next question to answer is whether the same value of $q_{low}$ should be used at all resolutions. We should also confirm that the LGF algorithm works well with irregularly spaced data. A little reflection shows that a level dependent $q_{low}$ falls naturally within the multiresolution framework of the LGF algorithm. The wavelet components of the true function are usually concentrated at the lower resolutions while the higher resolution wavelets mainly represent the noise in the data. This suggests using a “larger” value of $q_{low}$ at low resolution so that low resolution wavelets are more easily selected. Conversely, a “smaller” value of $q_{low}$ should be employed at the higher resolutions, so that wavelets capturing noise are less readily selected. A simple recipe is to choose a level dependent probability $q_{low}$ as

$$q_{low}(j) = \min\left(\frac{c}{2^j}, \frac{1}{2}\right), \quad (5.46)$$

Here $2^j$ is the number of basis functions at resolution $j$ and $c$ is a constant. The value of $c$ affects the level of smoothness of the estimated function.

A simulation was conducted to test the performance of the level dependent $q_{low}(j)$ for the Blocks and Heavisine test functions using the Haar and Symlet8 wavelet systems respectively. The noisy data was synthesized by adding i.i.d. Gaussian noise of mean zero and $\sigma = 0.5$ to 1024 irregular (randomly spaced) data in each case. The estimation was repeated 100 times at 32 separate values for $c$ in the range 0.005 to 200 and Figure 5.17 shows the TMSE statistics obtained. A value of $c$ in the range of 1 to 10 seems an appropriate choice. A smaller value of $c$ favours a smoother function and for this reason we shall adopt the level dependent prescription,

$$q_{low}(j) = \min\left(\frac{1}{2^j}, \frac{1}{2}\right) \quad (5.47)$$

for the remainder of this chapter. It is of course possible to formulate more complex methods for selecting $q_{low}(j)$ but this will not be pursued further in this study. Figure 5.18 shows the Blocks and Heavisine function denoised by the LGF algorithm using equation (5.47) for $q_{low}(j)$. 

147
Figure 5.17 The TMSE statistics as a function of constant $c$ in level dependent thresholding.

Figure 5.18 The Heavisine and Blocks functions denoised by LGF technique with Level dependent $q_{low}(j)$ (1024 randomly spaced samples, Least squares wavelet decomposition with Symlet8)
5.5.2 The Comparison of the LGF Method with Conventional Wavelet Thresholding

It is instructive at this point to compare the performance of the proposed data domain LGF algorithm with well-established denoising methods based on wavelet thresholding. Conventional thresholding techniques are only applicable to regularly spaced data and invariably employ the highly efficient FWT decomposition. The LGF algorithm can be used with any suitable decomposition technique and is equally applicable to regularly and irregularly spaced data. Our aim in this section is to demonstrate through examples that the performance of the LGF algorithm is similar (and often superior) to that of the conventional thresholding methods. To this end we shall apply the LGF algorithm using the FWT decomposition to give a more direct comparison with the wavelet thresholding methods. A large simulation was conducted to compare the performance of the LGF algorithm with universal hard and soft thresholding and the level dependent SURE thresholding techniques for a noisy Heavisine function. The estimation was repeated 10000 times for 1024 regularly spaced data contaminated with different Gaussian noise patterns with zero mean and \( \sigma = 0.5 \). The FWT decomposition with periodic extension and the Symlet8 orthogonal wavelet system was used in all cases. The level dependent threshold probability \( q_{\text{low}}(j) \) was selected using equation (5.47). Figure 5.19 shows a comparison between the various methods for a particular noise pattern and the detailed statistics obtained from 10000 runs are summarized in Table 5.4. The second and third columns in Table 5.4 show the statistical mean of the number of selected basis functions \( N \) and the TMSE for each denoising method. The LGF estimate is on the whole sharper than the universal soft threshold estimate, smoother than the SURE estimate and similar to the universal hard threshold estimate.
**Figure 5.19** Comparison between performance of the LGF algorithm and various wavelet thresholding methods (1024 uniform samples, FWT with Symlet8 wavelets)

**Table 5.4** Detailed statistics for 10000 runs for denoising of the Heavisine function with various denoising techniques. (1024 uniform samples, FWT with Symlet8 wavelets)

<table>
<thead>
<tr>
<th>Denoising method</th>
<th>N  (average)</th>
<th>TMSE (average)</th>
<th>TMSFDE (average)</th>
<th>TMSSDE (average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGF</td>
<td>23</td>
<td>0.0219</td>
<td>0.0069</td>
<td>0.0155</td>
</tr>
<tr>
<td>Universal hard</td>
<td>15</td>
<td>0.0293</td>
<td>0.0090</td>
<td>0.0207</td>
</tr>
<tr>
<td>Universal soft</td>
<td>15</td>
<td>0.0442</td>
<td>0.0074</td>
<td>0.0155</td>
</tr>
<tr>
<td>SURE</td>
<td>79</td>
<td>0.0227</td>
<td>0.0120</td>
<td>0.0297</td>
</tr>
</tbody>
</table>
Up to this point the performance of the LGF algorithm has been assessed by considering the statistical average of the TMSE and the visual inspection of the denoised estimates for particular cases. The visual inspection is based largely on comparison between local sharpness and curvature of the estimated and true functions, which in turn depend on the first and second derivatives that can be readily computed using finite differences. We can therefore define the difference between derivatives of the estimated and the true functions as:

First Difference Error: \[(FDE_i) = \left| \frac{d\hat{f}(x)}{dx} \right| - \left| \frac{df(x)}{dx} \right| \]

Second Difference Error: \[(SDE_i) = \left| \frac{d^2\hat{f}(x)}{dx^2} \right| - \left| \frac{d^2f(x)}{dx^2} \right| \]

Small values of \(FDE_i\) and \(SDE_i\) indicate that the estimated and true function have close sharpness and curvature at point \(x_i\). Mean squared values of the point errors over a dense data set containing \(M\) regular data points:

True Mean Square First Difference Error \[TMSFDE = \frac{1}{M} \sum_{i=1}^{M} (FDE_i)^2\]

True Mean Square Second Difference Error \[TMSSDE = \frac{1}{M} \sum_{i=1}^{M} (SDE_i)^2\]

can be used to compare the overall smoothness properties obtained from various denoising methods. The fourth and fifth columns of Table 5.4 compare the values of the TMSFDE and TMSSDE for the LGF and wavelet thresholding methods averaged over 10000 repetitions. The average TMSFDE and TMSSDE of the LGF method are close to the values for the soft thresholding method and less than the values for hard and SURE thresholding methods. The price paid for the enhanced smoothness associated with soft thresholding is an increase in the (average) TMSE. It is notable that the TMSE obtained with the LGF algorithm is closer to that of the hard thresholding technique and the improved smoothness is not obtained at the expense of

151
an increased TMSE. We do not of course expect a better TMSE than the SURE
denoising technique which forgoes smoothness properties to achieve a low TMSE.

From the above simulation it appears that the LGF algorithm combines the
best features of hard and soft thresholding without suffering from the major
shortcomings of the SURE method with regards to smoothness properties. The key to
the performance of the LGF algorithm is the way the basis functions are selected at
various resolutions. This is best demonstrated by considering the number of basis
functions selected at each resolution for the Blocks function using various algorithms.
Figure 5.20 illustrates the estimation of the Blocks function from 1024 noisy ($\sigma = 1$)
regular samples using the universal hard thresholding, the SURE thresholding, and the
LGF denoising algorithm.

The number of basis functions selected at each resolution by the LGF method
is close to that of hard thresholding but is substantially smaller than the SURE
method. There are some small differences at intermediate resolutions between the
LGF and the hard thresholding methods, largely because the basis functions
associated with small wavelet coefficients are excluded by the keep or kill recipe of
hard thresholding. It is also notable that the LGF algorithm terminates at resolution 9
with no wavelets selected at the 10th resolution. In hard thresholding, however, some
wavelets are retained at resolution 10, which cause the spikes observed in the
estimated function. The LGF denoising method, although retaining a larger number of
basis functions, operates more conservatively by terminating at a lower resolution and
delivering a smoother estimate.
In many practical situations the characteristics of the noise source contaminating the true underlying signal is non-stationary. Put another way the noise may be spatially variable and exhibit different spreads over various parts of the signal domain. This may occur in practice when the behaviour of the measurement device and/or transmission media is subject to variation. For spatially variable Gaussian noise, the wavelet coefficients are no longer i.i.d. distributed and do not exhibit a constant variance. Conventional thresholding methods rely on such properties and perform inadequately for spatially variable noise. Kovac and Silverman (2000) developed a coefficient dependent wavelet thresholding method to deal with non-
stationary Gaussian noise (see Section 5.2.4). The LGF algorithm makes no assumptions about the variance of the wavelet coefficients and can therefore deal with spatially variable noise directly. In fact, the LGF algorithm operates entirely in the data domain and uses the point-wise variance information \( \sigma_i^2 \) directly to formulate the local chi-square within each block \( B(j, k) \):

\[
\chi^2_i(j, k) = \sum_{t=1}^{M_{(j,k)}} \left( \frac{Y_{f,k}(t) - \hat{Y}_{f,k}(t)}{\sigma_i} \right)^2
\]

c.f. (5.40)

We shall employ an example to bring out the substantial superiority of the LGF algorithm in dealing with spatially variable noise. Figure 5.21 shows the Heavisine function contaminated with a Gaussian noise with zero mean and a spread which varies with the square of the distance from the origin:

\[
\sigma_i = x_i^2, \quad x_i = \frac{i}{2^{10}}, \quad i = 1, \ldots, 2^{10}
\]

c.f. (5.33)

The coefficient dependent thresholding method was used to denoise this function using both the universal hard and soft thresholding. The same data was denoised using the LGF algorithm combined with FWT and the level dependent \( q_{lev}(f) \) of equation (5.47). The estimation was repeated 2000 times in each case and the average statistics are given in Table 5.5. Typical results obtained for a particular spatially variable noise pattern are shown on Figure 5.21.

**Table 5.5** Detailed statistics for 2000 runs for denoising of the Heavisine function with various denoising techniques. (1024 samples with spatially variable Gaussian noise, FWT with Symlet8 wavelets)

<table>
<thead>
<tr>
<th>Denoising methods</th>
<th>N (average)</th>
<th>TMSE (average)</th>
<th>TMSFDE (average)</th>
<th>TMSSDE (average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGF</td>
<td>70</td>
<td>0.0160</td>
<td>0.0058</td>
<td>0.0141</td>
</tr>
<tr>
<td>Coef. dep. univ. hard</td>
<td>62</td>
<td>0.0471</td>
<td>0.0231</td>
<td>0.0565</td>
</tr>
<tr>
<td>Coef. dep. univ. soft</td>
<td>62</td>
<td>0.0474</td>
<td>0.0188</td>
<td>0.0445</td>
</tr>
</tbody>
</table>
Figure 5.21 Denoising of the Heavisine function contaminated with spatially variable noise using the LGF and coefficient dependent thresholding (1024 uniform samples, FWT with Symlet8 wavelets)

It is evident that the LGF estimate is superior to the coefficient dependent thresholding estimate in all respects. In particular, a similar number of basis functions are selected, the (average) TMSE of the LGF algorithm is about a third of that for the coefficient dependent thresholding methods and its smoothness properties are also far superior. This is simply because the LGF method uses the local information about the noise directly. In contrast, the coefficient dependent wavelet thresholding method does not use the noise variance information directly and translates this information into a covariance matrix for the wavelet coefficients. The method proposed by Kovac and Silverman (2000) also puts aside the covariance information and only uses the variance information (the diagonal elements of the covariance matrix) for threshold selection and complexity control. While it may be possible to develop more complex thresholding methods that use the off-diagonal covariance information, we believe
that operating entirely within data domain offers a more direct route for dealing with spatially variable noise.

5.5.3 Denoising of Irregular Data: the LGF Algorithm with the Least Squares Wavelet Decomposition

The LGF denoising algorithm developed in this study is rather general. The only requirement is a set of regular (first generation) wavelets and the associated dyadic multiresolution structure. The LGF algorithm operates entirely in the data domain, does not put any restriction on data spacing and can be used with any suitable method for wavelet decomposition. For example, in the case of regularly spaced data we can take full advantage of the highly efficient FWT algorithm. In the case of irregularly spaced data, we can first project the data onto a regular grid and then apply the FWT algorithm to the projected data. This, however, has the serious drawback of not dealing with the original irregular data set directly and the denoising performance will inevitably depend on the projection method employed. The simultaneous least squares wavelet decomposition, examined in detail in Chapter 4, enables the direct application of the LGF algorithm to the original irregularly spaced data.

The LGF denoising algorithm and the least squares wavelet decomposition have a number of inherent synergies and are naturally well matched. In particular, the least squares wavelet decomposition is based on chi-square minimization and the LGF algorithm is also based on evaluating and testing of the local chi-square. In the application of the LGF algorithm we could first pre-compute and store the wavelet coefficients at all resolutions to form the vector $\mathbf{d}$. We can then test the LGF criterion starting from the coarsest resolution $J$ and proceed level by level, setting the coefficients of the non-selected basis functions to zero, to give the modified vector of selected coefficients $\mathbf{d}'$ for reconstruction. This presents little difficulty in the case of regular data where we can use the highly efficient FWT algorithm. In the case of irregular data, however, computing all the wavelet coefficients using the least squares wavelet decomposition may become computationally expensive. This is simply because it is necessary to solve a very large least squares problem. In a regular multiresolution structure, the number of basis functions at all resolutions is roughly
equal to the number of data points. So that with as few as 1024 data points, we may have to deal with a 1024\times 1024 matrix in the least squares minimization. The synergy between the LGF algorithm and the least squares wavelet decomposition enables us to break this large problem into the solution of a sequence of smaller least squares minimizations.

The selection of basis functions in the LGF algorithm is carried out entirely in the data domain and requires no information about the wavelet coefficients. Consequently, we can first select the basis functions at a given resolution and then compute the wavelet coefficients only for the selected wavelets. Put another way, the coefficients for the selected wavelet may be computed on the fly as we proceed up the resolution levels. We start by selecting all the scaling functions and wavelet at the coarsest resolution \( J \), the vector of corresponding coefficients \( \hat{\alpha}_J^T = (\hat{\alpha}_J^T \mid \hat{\alpha}_J^T) \) may be then computed by solving the linear system:

\[
(A_J^T A_J) \hat{\alpha}_J = A_J^T y
\]  

(5.52)

Where the design matrix \( A_J \) is given in partitioned form by

\[
A_J = [G_J \mid H_J]
\]  

(5.53)

and \( G_J \) and \( H_J \) are the design matrices corresponding to scaling function and wavelets at resolution \( J \), see Section 4.4.2. The estimated function at the next higher resolution \( J+1 \) is then simply,

\[
\hat{\mathbf{y}}_{J+1} = A_J \hat{\alpha}_J
\]  

(5.54)

and may be used to form the local chi-squares in each block at resolution \( J+1 \) in order to test the LGF criterion and select the appropriate basis function. The basis functions selected at resolution \( J+1 \) may then be appended to those previously selected to form the design matrix and coefficient vector

\[
A_{J+1} = [G_J \mid H_J \mid H_{J+1}]
\]  

(5.55)

\[
\hat{\alpha}_{J+1}^T = (\hat{\alpha}_J^T \mid \hat{\alpha}_J^T \mid \hat{\alpha}_{J+1}^T)
\]  

(5.56)

The coefficients for the selected basis function up to resolution \( J+1 \) can then be obtained by solving the linear system

\[
(A_{J+1}^T A_{J+1}) \hat{\alpha}_{J+1} = A_{J+1}^T y
\]  

(5.57)
This procedure can be continued level by level to higher resolutions and is terminated at a level when no additional wavelets are selected. The problem is therefore broken down to the sequential formulation and solution of a series of smaller simultaneous least squares problems.

A simple example serves to highlight the selection of the basis functions and the size of the simultaneous least squares sub-problems encountered at each level. Figure 5.22 shows the estimate obtained by the on the fly application of the LGF algorithm to 1024 randomly spaced noisy samples of the Blocks function using the Haar wavelets and least squares decomposition. The number of basis functions selected at each resolution is also shown in Figure 5.22.

![Graph showing true signal and noisy signal with denoising results and selected basis functions](image)

**Figure 5.22** The estimate of a noisy Blocks function and the number of basis functions at each resolution

The total number of basis functions available at resolution $j$ is $2^j$ and it is evident that only a small fraction of the available basis functions are selected. It is also instructive to see the actual location of wavelets selected at each resolution for the test problem under consideration. Figure 5.23 shows the true function and the
location of the Haar wavelets selected at each resolution ranging for $J=0$ to $L=9$. It is noteworthy that a very small fraction of the high resolution wavelets are selected, which correspond to locations where the true underlying function has sharp discontinuities.

Figure 5.23 The true Blocks function and the location of Haar wavelets selected at each resolution for example of Figure 5.22

Table 5.6 gives the size of the successive least squares problems solved at each resolution. The numbers in brackets show the (cumulative) number of available basis functions at each resolution. Pre-computing all the wavelet coefficients at once
requires the solution of a single least squares problem of roughly $1024 \times 1024$ size. It is evident that applying the LGF algorithm on the fly requires far less computational effort to carry out the successive least squares minimizations. We also note that applying the LGF algorithm on the fly stabilizes the calculations. Pre-computing all the wavelet coefficients at once is prone to over-selection of high resolution wavelets, which is reflected in spurious sharp oscillations in the estimated function. With the on the fly application of the LGF algorithm, fewer and fewer wavelets are selected as we proceed up the resolution levels. This minimizes the risk of unwanted contributions from improperly selected high resolution wavelets.

Table 5.6 The cumulative number of basis functions and the size of least squares sub-problems for the example shown in Figure 5.22

<table>
<thead>
<tr>
<th>Resolution $j$</th>
<th>Cumulative No. of selected basis</th>
<th>Size of least squares sub-problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2 (2)</td>
<td>$1024 \times 2$</td>
</tr>
<tr>
<td>1</td>
<td>4 (4)</td>
<td>$1024 \times 4$</td>
</tr>
<tr>
<td>2</td>
<td>8 (8)</td>
<td>$1024 \times 8$</td>
</tr>
<tr>
<td>3</td>
<td>13 (16)</td>
<td>$1024 \times 13$</td>
</tr>
<tr>
<td>4</td>
<td>19 (32)</td>
<td>$1024 \times 19$</td>
</tr>
<tr>
<td>5</td>
<td>26 (64)</td>
<td>$1024 \times 26$</td>
</tr>
<tr>
<td>6</td>
<td>34 (128)</td>
<td>$1024 \times 34$</td>
</tr>
<tr>
<td>7</td>
<td>42 (256)</td>
<td>$1024 \times 42$</td>
</tr>
<tr>
<td>8</td>
<td>46 (512)</td>
<td>$1024 \times 46$</td>
</tr>
<tr>
<td>9</td>
<td>49 (1024)</td>
<td>$1024 \times 49$</td>
</tr>
</tbody>
</table>

The next example is chosen to demonstrate the robustness of the LGF algorithm with least squares decomposition for strongly irregular data. The data set consists of 1024 samples of the Heavisine function located at

$$x_i = \left(\frac{i}{2^{10}}\right)^2, \quad i = 1, \ldots, 2^{10}$$

and contaminated with random noise drawn from a Gaussian distribution with zero mean and $\sigma = 0.5$. Note that the data spacing is highly non-uniform and varies as the
square of distance from the origin. Figure 5.24 compares the estimates obtained using universal hard thresholding and the LGF algorithm. In both cases, all wavelet coefficients were pre-computed using the least squares wavelet decomposition and the Mexican hat wavelet system. Thresholding the wavelet coefficients using the universal hard criterion produces a very poor estimate. In contrast, the LGF algorithm delivers an excellent reproduction of the true underlying function for this highly irregular data set.

![Figure 5.24](image)

**Figure 5.24** Estimates of the Heavisine function recovered from highly irregular samples using the LGF algorithm and universal hard thresholding (Least squares decomposition with Mexican hat wavelets)

The poor performance of universal hard thresholding is not too surprising, since this technique is designed for regularly spaced data. The next example is constructed to give a thorough comparison of the LGF algorithm with the coefficient dependent thresholding method of Kovac and Silverman (2000) for an irregular data set with stationary noise pattern. Figure 5.25 compares the denoising of a Heavisine function using the LGF algorithm and the least squares wavelet decomposition with
coefficient dependent hard, soft, and SURE thresholding using linear interpolation and the FWT algorithm. The data consists of 1024 randomly spaced samples of the Heavisine function contaminated with Gaussian noise with $\sigma = 1$. The statistics obtained for 2000 repetitions of this test case is given in Table 5.7. The LGF estimate is substantially better than all the coefficient dependent thresholding methods in every respect. In particular, the LGF estimate is less complex with a much smaller number of selected wavelets; it also has the smallest TMSE and at the same time delivers the best smoothness properties measured in terms of the first and second derivative errors.

Figure 5.25 Estimate of a Heavisine function contaminated with unit variance Gaussian noise using the LGF algorithm and coefficient dependent universal hard, soft, and SURE thresholding
Table 5.7 Detailed statistics for 2000 runs for denoising of the Heavisine function with various denoising techniques. (1024 randomly samples contaminated with Gaussian noise σ=1.)

<table>
<thead>
<tr>
<th>Denoising method</th>
<th>N (average)</th>
<th>TMSE (average)</th>
<th>TMSFDE (average)</th>
<th>TMSSDE (average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGF</td>
<td>19</td>
<td>0.0528</td>
<td>0.0083</td>
<td>0.0165</td>
</tr>
<tr>
<td>Coef. Dep. Univ. hard</td>
<td>55</td>
<td>0.1016</td>
<td>0.1233</td>
<td>0.3940</td>
</tr>
<tr>
<td>Coef. Dep. Univ. soft</td>
<td>55</td>
<td>0.0723</td>
<td>0.0347</td>
<td>0.1056</td>
</tr>
<tr>
<td>Coef. Dep. Univ. SURE</td>
<td>360</td>
<td>0.1494</td>
<td>0.2607</td>
<td>0.8152</td>
</tr>
</tbody>
</table>

The last example is chosen to highlight the substantial superiority of the LGF algorithm for irregular data contaminated with spatially variable noise, see Figure 5.26. The data set considered is 1024 randomly spaced samples of the Heavisine function contaminated with a Gaussian noise whose spread varies with the square of distance from the origin:

\[ \sigma_i = x_i^2, \quad x = \text{sort} \{ \text{rand}(1024) \} \]  \hspace{1cm} (5.59)

The coefficient dependent thresholding method was used to denoise this data set using both the universal hard and soft thresholding. The same data were denoised using the LGF algorithm and least squares decomposition with the level dependent threshold \( q_{\text{low}}(j) \) of equation (5.47). The estimation was repeated 2000 times in each case and the average statistics are given in Table 5.8. The LGF estimate is again far superior in terms of complexity, TMSE and smoothness properties.

Table 5.8 Detailed statistics for 2000 runs for denoising of the Heavisine function with various denoising techniques. (Data: 1024 randomly spaced samples with spatially variable Gaussian noise, FWT algorithm with Symlet8)

<table>
<thead>
<tr>
<th>Denoising method</th>
<th>N (average)</th>
<th>TMSE (average)</th>
<th>TMSFDE (average)</th>
<th>TMSSDE (average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGF</td>
<td>47</td>
<td>0.0204</td>
<td>0.0073</td>
<td>0.0155</td>
</tr>
<tr>
<td>Coef. dep. univ. hard</td>
<td>97</td>
<td>0.0633</td>
<td>0.0421</td>
<td>0.1163</td>
</tr>
<tr>
<td>Coef. dep. univ. soft</td>
<td>97</td>
<td>0.0574</td>
<td>0.0201</td>
<td>0.0467</td>
</tr>
</tbody>
</table>
Figure 5.26 Estimates of the Heavisine function recovered from 1024 randomly spaced samples contaminated with spatially variable noise using the LGF algorithm and coefficient dependent universal hard and soft thresholding.

5.5.4 Extension to Higher Dimensions

A major motivation for the development of the least squares wavelet decomposition in Chapter 4 and the LGF algorithm in this chapter was the possibility of developing a general wavelet denoising technique for irregular data in higher dimensions. Current denoising methods are largely derived for image processing applications and deal with regularly spaced two-dimensional data (see Chambolle, et al 1998). A limited number of attempts have been made for denoising of irregular two-dimensional data using the lifting scheme. The methods suggested, however, are
not sufficiently general and rely on the highly specialized 2-D Haar wavelet system (see Bonneau, 1998; Schröder and Sweldens, 1995-a, 1995-b). In principle, the projection method and coefficient dependent thresholding could possibly be extended to higher dimensions. However, to our knowledge, no such extensions have been reported; most probably because of the severe difficulties associated with interpolation of sparse high dimensional data. There is therefore strong incentive for the development of a general multidimensional denoising algorithm capable of dealing with irregularly spaced data. The combination of the least squares wavelet decomposition and the LGF algorithm provides a suitable framework for such development. The least squares wavelet decomposition method is readily extended to higher dimensions as detailed in Chapter 4. There is also no assumption in the LGF algorithm to complicate its application to two and higher dimensions. A multidimensional multiresolution structure is easily constructed and multidimensional regular wavelet systems can also be readily constructed as the tensorial products of 1-D basis functions. We shall only demonstrate the LGF denoising of irregular 2-D data sets, but there is nothing in the development to preclude its extension to any number of dimensions.

Consider the 2-D multiresolution structure shown in figure 5.27. We shall refer to each “cube” in this structure as block B(j, k, l) addressed by its dilation index j and translation indices k and l. Each block B(j, k, l) contain three 2-D wavelets constructed as tensor products of 1-D scaling functions and wavelets:

![Figure 5.27 Blocks B(j, k, l) identified on a 2-D pyramidal multiresolution grid](image-url)

**Figure 5.27** Blocks B(j, k, l) identified on a 2-D pyramidal multiresolution grid
Here $\psi_{j,k,l}(x_1,x_2)$ and $\psi_{j,l}(x_1)$ are dilations and translations of a single 1-D mother wavelet $\psi(\cdot)$. Similarly $\varphi_{j,k}(x_1)$ and $\varphi_{j,l}(x_2)$ are dilations and translations of the corresponding 1-D scaling function $\varphi(\cdot)$. We also note that the coarsest representation of the function involves no wavelets and is constructed using 2-D scaling functions which are themselves a tensorial product of 1-D scaling functions

$$\varphi_{J,k,l}(x_1,x_2) = \varphi_{J,k}(x_1)\varphi_{J,l}(x_2)$$

The 2-D approximation at level $j$ can therefore be represented as:

$$F_j(x_1,x_2) = \sum_{k=1}^{N_j} \sum_{l=1}^{N_j} c_{j,k,l} \varphi_{j,k,l}(x_1,x_2)$$

$$+ \sum_{m=1}^{j-1} \sum_{k=1}^{N_j} \sum_{l=1}^{N_j} \left( d_{m,k,l}^1 \psi_{m,k,l}^1(x_1,x_2) + d_{m,k,l}^2 \psi_{m,k,l}^2(x_1,x_2) + d_{m,k,l}^3 \psi_{m,k,l}^3(x_1,x_2) \right)$$

With the above definitions in place, the LGF algorithm remains the same as that given in Figure 5.12, except that the local chi-square for block $B(j,k,l)$ is calculated from

$$\chi^2_{j,k,l} = \sum_{i=1}^{M_{j,k,l}} \left( \frac{y_{j,k,l}(i) - \hat{y}_{j,k,l}(i)}{\sigma_{j,k,l}(i)} \right)^2$$

where

$\chi^2_{j,k,l}$ = abscissa of data within block $B(j,k,l)$

$y_{j,k,l}$ = ordinate of data within block $B(j,k,l)$

$\sigma_{j,k,l}$ = standard deviation of noise for data within block $B(j,k,l)$

$\hat{y}_{j,k,l}$ = vector of estimates within block $B(j,k,l)$ at resolution $j$

i.e. $\hat{y}_{j,k,l}(i) = F_j(x_i)$ for $x_i \in x_{j,k,l}$
The effective number of basis functions which have contributed to the 2-D estimate \( F_j(x_1, x_2) \) within block \( B(j, k, l) \) is given by

\[
N_L(j, k, l) = \frac{\text{total number of basis functions up to level } j-1}{\text{Number of blocks at level } j} = \frac{2^{2j}}{2^j} = 1 \quad (5.66)
\]

The local degree of freedom is therefore as before

\[
v_L(j, k, l) = M_L(j, k, l) - 1 \quad (5.67)
\]

where \( M_L(j, k, l) \) is the number of data points falling in block \( B(j, k, l) \). The local measure of the goodness of fit \( q_L(j, k, l) \) is obtained from:

\[
q_L(j, k, l) = Q\left( \frac{v_L(j, k, l)}{2}, \frac{\chi^2_L(j, k, l)}{2} \right) \quad (5.68)
\]

Selection of the threshold limit \( q_{low} \) also remains unchanged and we may use the level dependent probability \( q_{low}(j) \) of equation (5.47).

As the first 2-D test case for the LGF denoising algorithm we shall assume that the true underlying function is composed of two separate parts representing distinct local and global features:

\[
F(x_1, x_2) = F_{local}(x_1, x_2) + F_{global}(x_1, x_2) \quad (5.69)
\]

The local feature will be taken as,

\[
F_{local}(x_1, x_2) = 100(1 - 1.100(x_1 + 1) + 0.967(x_1 + 1)^2) \exp\left(-\left(\frac{x_1 - 0.5}{0.75}\right)^2\right) \times (1 - 1.100(x_2 + 1) + 0.967(x_2 + 1)^2) \exp\left(-\left(\frac{x_2 - 0.5}{0.75}\right)^2\right) \quad (5.70)
\]

and has virtually no response outside the range \( 1/3 \leq x_1, x_2 \leq 2/3 \). The global feature is taken as

\[
F_{global}(x_1, x_2) = \exp(-0.4(x_1 + 1)) + 0.02 \exp(-1.667(x_2 + 1)) \quad (5.71)
\]

and has a response over the entire input domain \( 0 \leq x_1, x_2 \leq 1 \). The true underlying function obtained on combining the above local and global features is shown on Figure 5.28. A 2-D noisy data was synthesized by adding random noise drawn from a normal distribution with zero mean and a spread of \( \sigma = 0.2 \) to 1600 randomly spaced samples of the underlying function within the unit square. The location of the samples
and the noisy data set are shown on Figures 5.28 for comparison purposes. The LGF algorithm and the least squares wavelet decomposition were employed to denoise this noisy data set. Comparison of the denoised estimate and the true underlying function validates the applicability of the LGF algorithm in a 2-D setting. The minor differences observed around the edges of the data domain could possibly be reduced by incorporating specialized treatment of edge effects.

Figure 5.28 Denoising an irregular 2-D data set contaminated with Gaussian noise using the 2-D least squares wavelet decomposition and the LGF algorithm

The global feature in the previous example was deliberately chosen so that there was no input interaction in the global trend. The next example is chosen to confirm that the LGF algorithm can handle strong input interactions in both the local
and global features. The local feature remains as equation (5.70) but an interacting term is added to the global feature,

\[ F_{\text{global}}(x_1, x_2) = \exp(-0.4(x_1 + 1)) + 0.02 \exp(-1.667(x_2 + 1)) - 0.167(x_1 + 1)(x_2 + 1) \]  

(5.72)

interaction term

The true underlying function and a set of 3600 randomly spaced samples contaminated with the Gaussian noise with \( \sigma = 0.2 \) are shown in Figure 5.29. Comparison of the denoised estimate and the true underlying function confirm that the LGF algorithm is capable of handling local and global input interactions.

Figure 5.29 Denoising of a function with global interaction from irregular 2-D samples contaminated with Gaussian noise using the 2-D least squares wavelet decomposition and the LGF algorithm.
For a final 2-D example, we consider a case where the data is contaminated with spatially variable Gaussian noise. The true function is taken as,

$$F(x_1, x_2) = \sin(2\pi x_1) \sin(2\pi x_2)$$  \hspace{1cm} (5.73)

over the unit square $0 \leq x_1, x_2 \leq 1$ and is shown in Figure 5.30. The noisy data set was synthesized by selecting 2500 randomly spaced samples and adding a normal noise whose spread varied as:

$$\sigma_{i,j} = 0.1 x_i(i) x_j(j)$$  \hspace{1cm} (5.74)

The location of the samples and the noisy data set are shown on Figure 5.30. Comparison of the denoised estimate and the true function demonstrate that the capability of the LGF algorithm for handling non-stationary noise extends to two dimensions.

**Figure 5.30** Denoising a 2-D noisy data set contaminated with spatially variable Gaussian noise using the least squares wavelet decomposition and the LGF algorithm
5.6 Conclusions

Real data are invariably contaminated with a certain measure of uncertainty or noise. The determination of the true trend underlying a set of noisy data is therefore a recurrent problem in diverse areas. The wavelet representation of typical functions is sparse, the essential features of the signal are captured by a few large coefficients whereas the measurement noise is distributed amongst all the wavelet coefficients. Consequently, transforming the data into the wavelet domain and setting the small wavelet coefficients to zero (thresholding) provides an effective means for denoising. A wide range of thresholding techniques have been proposed and were critically reviewed in this chapter. The majority of such development is, however, restricted to denoising of uniformly spaced data contaminated with Gaussian noise and the well known wavelet thresholding techniques perform poorly with non-uniformly spaced data or non-Gaussian noise. The primary objective of this chapter was to develop a denoising technique capable of handling arbitrarily spaced multidimensional data contaminated with Gaussian noise.

A non-uniformly spaced data set can of course be projected onto a regular grid to open the way for the use of some variant of the conventional thresholding techniques. The best developed technique in this area is due to Kovac and Silverman (2000) who combined linear projection with a sophisticated coefficient dependent thresholding procedure. Results obtained in this chapter indicate that such a technique does not perform well on several examples. The most serious objection to this procedure lies in the fact that we no longer deal with the original data set. The projection based methods are also inevitably sensitive to the type of projection employed. Dealing with projected data correlates the wavelet coefficients and they are no longer independent. In the Kovac and Silverman (2000) implementation the thresholding is based on the variance of the individual wavelet coefficients and the covariance of the wavelet coefficients is not considered. Including the covariance information may improve the denoising performance but complicates the thresholding further. Major difficulties must also be faced in extending the projection based methods to two and higher dimensions, this is because interpolation of
multidimensional irregularly spaced data is a complex problem. Performing the
denoising operation in the data domain, rather than the wavelet domain, enables us to
deal with the irregular data directly and delivers methods that are readily extended to
higher dimensions.

In this chapter we developed a new data domain denoising technique, the LGF
algorithm, which employs a local measure of the traditional goodness of fit for
controlling the complexity of the estimated function. The LGF denoising algorithm
can handle regular and irregular data sets seamlessly and can be coupled to any
suitable wavelet decomposition method. For example, with regular data we can
combine the LGF algorithm with the FWT algorithm. For irregular data, the LGF
algorithm has inherent synergies with the least squares wavelet decomposition and
their combination provides a computationally efficient denoising procedure. The
application of the LGF denoising algorithm requires the selection of a single
parameter, the probability limit $q_{low}$. The selection of this tunable parameter was also
considered and a simple level dependent recipe for its selection was proposed.

A critical comparison of the performance of the proposed LGF algorithm and
wavelet domain thresholding methods was undertaken for carefully constructed
illustrative examples. The results obtained indicate that the LGF algorithm is
competitive with thresholding methods for regularly spaced data. It can, however,
deliver a substantially better denoising performance for irregularly spaced data and
non-stationary Gaussian noise. This superior performance was confirmed by several
large and small simulation exercises to arrive at meaningful statistical conclusions
regarding the number of basis functions retained, the true mean squared error
delivered and the smoothness properties of the estimated functions. The proposed
LGF algorithm is also readily extended to higher dimensions without sacrificing its
effectiveness.

The data domain LGF algorithm offers an effective means for denoising of
arbitrarily spaced multidimensional data contaminated with Gaussian noise, it can also
handle non-stationary or spatially variable Gaussian measurement errors. The
performance of the LGF denoising algorithm, in common with all other denoising
techniques considered so far, may deteriorate substantially when the data set contains significant "outliers" or the noise distribution has a longer tail than the compact Gaussian distribution. The next chapter is devoted to developing an algorithm for denoising of arbitrarily spaced multidimensional data contaminated with arbitrary noise.
Chapter 6

Multiresolution Denoising with Non-Gaussian Noise: Local Balance of Fit (LBF) Technique

6.1 Introduction

The highly efficient classical wavelet thresholding techniques are all developed on the understanding that the noise is independently and identically drawn (i.i.d.) from a Gaussian distribution. The new data domain LGF denoising method developed in Chapter 5 retains the Gaussian assumption but can be used for non-identically distributed or spatially variable noise. In many practical measurements, however, the data is contaminated by off points or outliers caused by instrument failure or human error. Outliers cannot be handled using the compact Gaussian distribution and force the use of a distribution with a much wider tail. In other cases, the measurement noise may be inherently non-Gaussian. For example, measurements based on counting exhibit a Poisson rather than normal noise distribution. Outliers and long-tail noise can severely compromise the performance of denoising procedures developed on the assumption of Gaussian noise. Wavelet decomposition is a linear transformation of the data and the outliers exert an unbounded influence on the wavelet coefficients. Consequently, wavelet shrinkage denoising methods or the data domain LGF denoising technique do not perform adequately in the presence of outliers. Direct handling of outliers or data contaminated with non-Gaussian noise has critical importance in non-stationary applications like tracking, signal and image processing, navigation and fault detection. Our aim in this chapter is to develop a data domain denoising method that is effectively distribution free and can handle arbitrarily spaced data contaminated with independent but arbitrarily distributed noise.

Multiresolution denoising is developed on the premise that the essential features of the signal are captured by the lower resolution coefficients and the noise is
captured by the higher resolution coefficients. Both the classical discrete wavelet decomposition and the least squares wavelet decomposition are designed to preserve the local mean of the signal across the resolutions. Outliers have a strong influence on the mean and can therefore distort the lower resolution coefficients obtained through a mean-preserving decomposition. Consequently, the denoised signals recovered can become severely distorted in the presence of outliers. Much better denoising performance can be achieved through a multiresolution decomposition aimed at preserving the local median rather than the local mean of the signal. This is simply because the median is far less affected by outliers; a median preserving decomposition is “robust” in the sense that it limits the influence of outliers to the higher resolutions and prevents its leakage to the lower resolution. Median preserving decompositions have been examined by a number of authors, in particular by Donoho and Yu (1997) for regularly spaced data under fairly restrictive conditions. In this chapter we develop a median preserving decomposition, named Interpolated Block Median Decomposition (IBMD), which removes such restrictions and can be used with irregularly spaced data. The robust IBMD decomposition opens the way for the development of a completely new robust multiresolution data domain denoising algorithm, named the Local Balance of Fit (LBF) algorithm. The LBF algorithm is inspired by the seemingly effortless yet effective way a human operator is able to draw a smooth curve through a noisy data set. An individual drawing such a curve does not care about data spacing, handles the outliers automatically, performs no discernable calculations for noise estimation and needs no information about the noise distribution.

We start this chapter with a few examples which clearly highlight the failure of mean preserving decompositions and associated denoising methods in the presence of outliers. The concept of maximum likelihood estimation is employed in Section 6.3 to establish the robustness of the median compared to the mean in the presence of outliers. This is followed by a review of the median preserving decompositions reported in the literature for regularly spaced data with particular emphasis on the seminal work of Donoho and Yu (1997). The new Interpolated Block Median Decomposition (IBMD) algorithm is presented in section 6.5 with a discussion of its characteristics. Section 6.6 presents the development of the new data domain Local Balance of Fit (LBF) denoising algorithm. This algorithm has a single user defined
parameter and a universally applicable recipe for its selection is also presented. Section 6.7 is devoted to a comprehensive comparison of the IBMD-LBF denoising procedure for several examples which can also be handled using other denoising techniques. Such examples serve to confirm the robustness and excellent denoising performance of the proposed IBMD-LBF denoising method. A number of complex examples involving arbitrary spaced data contaminated with arbitrary noise are also presented that cannot be handled by other denoising methods but are dealt with seamlessly and effectively by the proposed IBMD-LBF method. There is nothing in either the IBMD algorithm or the LBF algorithm to prevent their extension to multidimensional denoising applications and Section 6.7.6 presents a number of illustrative two dimensional examples. The examples presented serve to confirm the IBMD-LBF combination as a distribution free method that can be confidently used for all types of denoising applications; the only requirement is that the noise is independent and uncorrelated.

6.2 Failure of Conventional Decompositions in the Presence of Outliers

The denoising methods described so far share a common assumption: the data is contaminated with noise drawn from an independent and identically distributed Gaussian distribution. In many practical applications the noise deviates strongly from normal distribution, either because the data is contaminated with a few outliers or because the noise has a broader tail than the compact Gaussian distribution. Outliers are frequently observed because of instrument failure, power flickers or human error in practical measurements. The noise associated with many measurement devices is also inherently non-Gaussian, for example measurements based on counting often exhibit a Poisson rather than normal distribution. In other applications, the data may have come from different sources with different noise distributions and the noise is then not identically distributed. Handling of data containing outliers, long tail noise distributions or non-identically distributed noise by the method discussed so far is problematic. A few examples highlight the shortcoming of conventional techniques for such complex situations.
The left hand panel in Figure 6.1 shows the Heavisine function contaminated with a Gaussian noise with zero mean and $\sigma = 0.5$. The right panel shows the same data but with a few additional outliers. The two data sets are denoised using the wavelet domain hard thresholding and the data domain LGF technique developed in Chapter 5. It is clear that the presence of outliers deteriorates the performance of both denoising techniques and special methods are needed to deal with outliers contaminating the data.

![Figure 6.1 Denoising of data contaminated with Gaussian noise (with and without outliers) using the universal hard thresholding and LGF methods](image)

**Figure 6.1** Denoising of data contaminated with Gaussian noise (with and without outliers) using the universal hard thresholding and LGF methods

Similar difficulties are encountered when the noise is not Gaussian. Figure 6.2 shows the Heavisine function contaminated with noise drawn from the longer tail double sided exponential distribution. Once again the universal hard thresholding and the LGF methods, which assume normal noise, perform poorly with spurious spikes contaminating the recovered signal.
Figure 6.2 Denoising of data contaminated with double sided exponential noise using the universal hard thresholding and LGF methods

The problem encountered in Figure 6.2 is severely exaggerated when the noise has even a longer tail. This is demonstrated in Figure 6.3, which shows the Heavisine function this time contaminated with noise drawn from the Cauchy distribution. The much longer tail of the Cauchy distribution generates data points that deviate significantly from the true function so much so that the shape of the underlying true function is obscured. The universal hard thresholding and the LGF method both fail drastically in this case and the recovered function bears no resemblance to the Heavisine function.
6.3 Robust Estimation: Median versus Mean Estimation

The basic reason for the drastic failure observed in the above examples is the influence of the outliers on the wavelet decompositions employed which are designed to preserve the local mean. Figure 6.4 illustrate the location and the amplitude of the wavelet coefficients at different resolutions for 1024 regularly spaced samples of the Heavisine function contaminated with Cauchy noise. The presence of the "outliers" is felt not only in the coefficients at the highest resolutions but also affects the coefficients at the lower resolutions. The consequence is that the reconstructed function is severely distorted even at the lower resolutions. This is clearly shown on Figure 6.4, which shows the (normalized) reconstructed functions at each resolution. In particular, the low resolution approximations bear no resemblance to the underlying function.
In an ideal world we can repeat the measurements at each spatial point a very large number of times, $m$. The data obtained at each location will be distributed according to some distribution depending on the nature of the measurement noise. Figure 6.5 gives a schematic for 100 repetitions at three different locations for Gaussian distribution. Focusing on a single location, say $x_k$, we seek a single estimate $\alpha$ that can serve as the estimate of the true underlying function at $x_k$. 

Figure 6.4 Location and amplitude of the wavelet coefficients and the reconstructed functions at different resolutions.
The single estimate $\alpha$ which minimize the sum of squared deviations

$$G = \sum_{i} (y_i(x_k) - \alpha)^2$$  \hfill (6.1) 

is simply the mean of the data

$$\alpha_{\text{mean}} = \frac{1}{m} \sum_{i=1}^{m} y_i(x_k) = y(x_k)$$  \hfill (6.2) 

The minimizer of squared deviations at point $x_k$ is therefore often referred to as the \textit{mean} estimator. It is clear that the mean estimator is very sensitive to the presence of outliers, a single "accidentally large" data point is sufficient to alter the mean significantly. The mean estimator is not robust in the presence of outliers.

A more robust estimate of the true value at location $x_k$ is obtained by seeking to minimize the sum of absolute deviations

$$G = \sum_{i=1}^{m} |y_i(x_k) - \alpha|$$  \hfill (6.3) 

It is easy to show that the estimator which minimizes the above objective is the median of data. For an ordered data set, $y_1 \leq y_2 \leq \cdots \leq y_m$, the sample median is given by:
The median estimator is inherently more robust in the presence of outliers. Unlike the mean, the median is not altered drastically by the presence of outliers. The conclusion is that a robust denoising technique may be forthcoming using a multiresolution decomposition which attempts to preserve the (local) median rather than the (local) mean of the data across the resolutions.

An introduction to the subject of robust estimation using the maximum likelihood concept was presented in Chapter 2. Table 6.1 shows the loss function and the empirical risk for the Gaussian, double sided exponential and Cauchy distributions. Minimizing the empirical risk provides the maximum likelihood estimate in each case. The maximum likelihood estimator reduces to the minimization of the sum of squared deviations for Gaussian noise (i.e. a mean estimator) whereas for the double sided exponential the sum of absolute deviation is minimized (i.e. a median estimator). The robustness of median estimator to outliers is readily demonstrated by a simple example.

**Table 6.1 Loss functions and the empirical risk functional for the Gaussian, double sided exponential and Cauchy distributions**

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Gaussian</th>
<th>Double sided exp.</th>
<th>Cauchy</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho(\delta_i)$</td>
<td>$\exp(-\frac{1}{2} \delta_i^2)$</td>
<td>$\exp(-</td>
<td>\delta_i</td>
</tr>
<tr>
<td>Loss function</td>
<td>$\frac{1}{2} \delta_i^2$</td>
<td>$</td>
<td>\delta_i</td>
</tr>
<tr>
<td>Empirical risk</td>
<td>$\mathcal{R}_{\text{emp}}(\delta_i)$</td>
<td>$\sum_{i=1}^{M} \delta_i^2$</td>
<td>$\sum_{i=1}^{M}</td>
</tr>
</tbody>
</table>
Figure 6.6 shows data obtained by adding noise drawn from a Cauchy distribution to a single constant line $y=1$. Attempts to recover this constant assuming Gaussian, double sided exponential and Cauchy distributions are also shown on this figure. It is clear that the outliers skew the mean estimator (for Gaussian distribution) significantly giving $\hat{y} = 0.1351$ rather than 1. The median estimator (for double sided exponential) gives a reasonable estimate of the constant at $\hat{y} = 0.938$. The best result is of course obtained from minimization of the empirical risk related to the Cauchy distribution. From this simple example, we may conclude that median estimation provides a sound basis for development of robust multiresolution denoising methods. In effect each tile in a multiresolution structure can be handled in much the same way as Figure 6.6. A few attempts at developing multiresolution decomposition and denoising procedures based on preserving the median have been reported and are reviewed next.

![Figure 6.6 Recovering a constant from 1024 samples contaminated with Cauchy noise, based on three different a priori assumptions for the noise distribution.](image)

183
6.4 Review of Robust Multiresolution Estimation Methods

In view of the frequent occurrence of outliers in practical measurements, a number of robust denoising procedures have been reported. The simplest way of dealing with outliers is to detect them so that they can be excluded from the data and allow the use of mean estimators developed on the basis of Gaussian noise distribution. For example, Kovac and Silverman (2000) suggest tabulating the absolute difference between adjacent data points. The variance \( \hat{\sigma} \) of the noise can then be estimated from the median of the tabulated data. A running median smoother with a proper window size (typically 10 points) is then applied to each data point to produce a smoother estimate of the signal. Any data point deviating from the smooth estimate by more than 1.96 \( \hat{\sigma} \) is considered as an outlier and is excluded from the original data set. The remaining data points, which will be irregularly spaced, can then be denoised using the projection method combined with the coefficient dependent thresholding technique detailed in section 5.3.1. Little is known about the performance characteristics of this approach, which may be sensitive to selection of a proper window size. A similar approach has been used by Chan and Zhang (2004). More direct methods for handling outliers and long tail noise distributions are based on down-weighting the effect of outliers on the low resolution wavelet coefficients. Bruce et al (1994) use a fast O(M) wavelet decomposition called robust smoother-cleaner, which is robust towards outliers. This method down weights the effect of outliers leaking in the estimation of wavelet coefficients at coarse levels. Bruce et al (1994) used the “B-spline” biorthogonal wavelets combined with median filters (of length 5 to 7) and a tuning parameter called the robust residual threshold.

The two techniques described above are based on estimation of the wavelet coefficients using a decomposition which is aimed at preserving the mean of the signal across the resolutions. Sardy et al [2001] proposed a robust wavelet based estimator using a robust loss function. The estimated coefficients are found as the solution to a regularization problem with a robust loss function:

\[
\min_{\hat{c}} \| s - \phi \hat{c} \| + \lambda \| \hat{c} \|
\]  

(6.5)
where \( \|w\|_p = \sum_{i=1}^{N} p(\omega_i) \) and \( p \) is a hybrid loss function which use the \( L^2 \) norm for small residuals and the \( L^1 \) norm for large residuals,

\[
p(\omega) = \begin{cases} 
\frac{\omega^2}{2} & |\omega| < \tau \\
\tau |\omega| - \tau^2 / 2 & |\omega| > \tau 
\end{cases}
\] (6.6)

This method requires solving a non-trivial optimization problem and selecting both a smoothing parameter \( \lambda \) and a robustness parameter \( \tau \), which is not a simple task.

A significant contribution to the development of robust multiresolution denoising is due to Donoho and Yu [1997]. These authors introduced a robust multiresolution decomposition which is aimed at preserving the median (rather than the mean) of the signal. Such a decomposition can limit the influence of outliers and prevent the corruption of the wavelet coefficients at lower resolutions. The proposed decomposition can only be used for a regularly spaced data set containing \( M=3^L \) individual data points. The \( M=3^L \) original data are divided into non-overlapping blocks \( B(j, k) \) which have a triadic pyramidal structure. Each block in a resolution \( j \) is divided into three blocks at the next higher resolution \( j+1 \). Using this triadic structure, Donoho and Yu [1997] succeeded in deriving an analytical expression for a unique quadratic whose median in each block is exactly the same as the median of the data in that block. The name “Median-Interpolating Pyramidal Transform” (MIPT) is used by Donoho and Yu [1997] for their median preserving decomposition.

The non-linear MIPT decomposition proposed by Donoho and Yu [1997] involves the following steps:

1- Block Median Calculation at resolution \( j \)

The data is divided into \( 3^j \) blocks of equal width and the median \( m_{j,k} \) of each block \( B(j, k) \) is calculated.

\[
m_{j,k} = \text{median}(y_i : y_i \in B(j, k))
\] (6.7)

2- Median-Interpolation at resolution \( j \)
A unique quadratic $P_j(x)$ is found whose median in each block $B(j, k)$ is the same as $m_{j,k}$ calculated at step 1.

$$\text{median} \left( P_j(x), x \in B(j,k) \right) = m_{j,k} \quad (6.8)$$

3- Median prediction at resolution $j+1$

Estimates (predictions) of the medians in the $3^{j+1}$ blocks at resolution $j+1$, $\hat{m}_{j+1,k}$, are calculated using polynomial $P_j(x)$ obtained in step 2.

$$\hat{m}_{j+1,k} = \text{Median} \left( P_j(x), x \in B(j+1,k) \right) \quad (6.9)$$

The above steps are repeated for all resolutions ranging for the coarsest resolution $J$ to the highest resolution $L$. The transform coefficient $d_{j,k}$ in each block is the difference between the median of samples located in the block $m_{j,k}$ and the estimate of block median obtained from median prediction $\hat{m}_{j,k}$.

$$d_{j,k} = m_{j,k} - \hat{m}_{j,k} \quad (6.10)$$

The conventional universal and SURE thresholds, which are based on the assumption of Gaussian noise, are inappropriate for removing non-Gaussian noise from the MIPT coefficients. Donoho and Yu (1997) proposed a method to compute proper level dependent threshold values for thresholding the MIPT coefficients for symmetric noise distributions. Given the cumulative distribution of the noise $P(.)$, the threshold at resolution $j$, $\tau_j$, is calculated from

$$\tau_j = \sqrt{3^{j-j}} \cdot \frac{1}{2} \cdot \sqrt{1 - \left( \frac{1}{2L3^j} \right)^2} \quad (6.11)$$

where $P^{-1}(.)$ is the inverse of $P(.)$ and $L$ is the finest triadic resolution considered.

From equation (6.11), it can be concluded that the magnitude of $\tau_j$ at low resolutions is governed by the behaviour of $P^{-1}(1/2)$. For smooth symmetric distributions with continuous derivatives around zero, the behavior of $P^{-1}(1/2)$ depends very little on the detailed nature of the distribution and similar thresholds are obtained at low resolutions for all distributions. At the highest resolutions, however, the behaviour of $P^{-1}(.)$ is strongly influenced by the tail of the distribution $\rho(.)$ and radically different threshold values are obtained depending on the tail of the noise distribution. Table 6.2 shows the level dependent thresholds.
for $M=3^8$ samples of Gaussian and Cauchy distributions calculated using procedures provided by Donoho and Yu (1997).

Table 6.2 MIPT thresholds at different resolutions calculated for $M=3^8$ samples of Gaussian and Cauchy white noise

<table>
<thead>
<tr>
<th>j</th>
<th>$\tau_j$ (Gaussian)</th>
<th>$\tau_j$ (Cauchy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4.2187</td>
<td>4.2557</td>
</tr>
<tr>
<td>3</td>
<td>4.8811</td>
<td>5.0563</td>
</tr>
<tr>
<td>4</td>
<td>5.4041</td>
<td>6.1683</td>
</tr>
<tr>
<td>5</td>
<td>5.8760</td>
<td>9.6588</td>
</tr>
<tr>
<td>6</td>
<td>6.3822</td>
<td>4.5599 x 10^1</td>
</tr>
<tr>
<td>7</td>
<td>6.9829</td>
<td>1.5879 x 10^4</td>
</tr>
<tr>
<td>8</td>
<td>7.5477</td>
<td>1.1470 x 10^13</td>
</tr>
</tbody>
</table>

As an illustrative example, Figure 6.7 shows the performance of the MIPT thresholding method for $6561 (=3^8)$ regular samples of the Heavisine function contaminated with Cauchy noise ($k=0.5$),

$$\rho(\varepsilon) = \frac{k}{\pi(k + \varepsilon^2)}$$

(6.12)

The recovered functions using the threshold values for Cauchy noise with $k=0.1$, $k=0.5$, and $k=1.0$ are compared in Figure 6.7. It is evident that for $k=0.5$, the outliers are effectively eliminated and an excellent estimation is obtained. Generally, this method has good performance when the data is regular and triadic, the noise is independent and identically distributed and the type and parameters of the noise distribution are known accurately.
Figure 6.7 Performance of the MIPT denoising method for data contaminated with Cauchy noise

### 6.5 Interpolated Block Median Decomposition (IBMD)

The MIPT algorithm of Donoho and Yu (1997) has the distinct advantage of guaranteeing that the median of the estimated function is the same as the median of the data in each block of the triadic multiresolution structure. The price paid, however, is restriction to a regularly spaced data set with a size which is an exact integer power of 3 and a triadic structure. Finding an analytical median interpolating polynomial is not straightforward for other than a triadic structure, a point emphasized by Donoho and Yu (1997). In practice, however, a simple polynomial interpolation of the block medians approximates well the unique median-interpolating quadratic. This approach has been used by Melnik et al (2001) who employed a triadic polynomial structure of blocks with simple polynomial interpolation of the block medians. The procedures of Donoho and Yu (1997) and Melnik et al (2001) (also Averkamp and Houdre, 2003) are restricted
to regularly spaced data and impose a restriction on the size of the data set and the noise distribution. Pang and Yu (2004) uses general continuous M-estimators (instead of median) for pyramid transform; but their analysis is restricted to one dimensional continuous functions. We shall now develop a robust decomposition which employs a simple dyadic multiresolution structure, attempts to preserve the local median, can be used with an arbitrary sized irregularly spaced data set and is easily extended to multidimensional applications. We shall refer to this new procedure as the Interpolated Block Median Decomposition or IBMD for short.

Consider the dyadic multiresolution structure shown in Figure 6.8. We shall refer to each tile within this structure as a block $B(j, k)$ addressed by its dilation $j$ and translation $k$ and assume that a basis function is centred within each block.

![Figure 6.8 Blocks B(j, k) identified on a pyramidal multiresolution grid](image)

Let us define the following vectors based on data points falling in a block $B(j, k)$:

- $x_{j,k} =$ abscissa of data within block $B(j, k)$
- $y_{j,k} =$ ordinate of data within block $B(j, k)$
The steps in the IBMD algorithm can be stated as follows:

1- Start at the lowest resolution \( j = J \)

2- Find the median of the samples in all blocks at resolution \( J \)

\[
m_{j,k} = \text{median}(y_{j,k}) \quad \text{for all } k
\]

(6.13)

3- Form the ordered block median set \( S_D_j \) whose elements are the block median \( m_{j,k} \) located at the centre of each block,

\[
\text{center}_{j,k} = \frac{k - \frac{1}{2}}{2^j}
\]

(6.14)

\[
S_D_j = \{(\text{center}_{j,k}, m_{j,k}) \text{ for all } k\}
\]

(6.15)

4- Interpolate the set \( S_D_j \) using cubic spline interpolation to find the approximation at resolution \( j \), \( F_j(x) \),

\[
F_j(x) = \text{interp}(SD_j)
\]

(6.16)

5- Use the approximation \( F_j(x) \) to predict the medians at the centres of blocks at resolution \( j+1 \).

\[
\hat{m}_{j+1,k} = F_j(\text{center}_{j+1,k})
\]

(6.17)

6- Find the sample medians in blocks at resolution \( j+1 \)

\[
m_{j+1,k} = \text{median}(y_{j+1,k}) \quad \text{for all } k
\]

(6.18)

and set the transform coefficients at resolution \( j+1 \) to the difference

\[
d_{j+1,k} = m_{j+1,k} - \hat{m}_{j+1,k} \quad \text{for all } k
\]

(6.19)

(Note: \( d_{j,k} = m_{j,k} \quad \text{for all } k \) at the coarsest resolution)

7- Repeat steps 3 to 6 for the next higher resolution up to resolution \( L-1 \)

Reconstruction can be performed for any arbitrary given set of locations \( x_t \) using the following steps:
1- Start from the coarsest resolution $J$ and form the ordered set $SR_J$

$$SR_J=\{(\text{center}_{j,k}, d_{j,k}) \text{ for all } k\}, \quad (6.20)$$

find

$$F_j(x) = \text{interp}(SR_J) \quad (6.21)$$

2- Form the ordered set $SR_{J+1}$

$$SR_{J+1}=\{(\text{center}_{j+1,k}, d_{j+1,k}) \text{ for all } k\}, \quad (6.22)$$

3- Find the detail function $f_j(x)$ at resolution $j$ by interpolating the set $SR_{J+1}$

$$f_j(x) = \text{interp}(SR_{J+1}) \quad (6.23)$$

4- Evaluate the approximation at resolution $j+1$ by adding the detail $f_j(x)$

$$F_{j+1}(x) = F_j(x) + f_j(x) \quad (6.24)$$

5- Repeat steps 2 and 4 up to resolution $L-1$.

The name Interpolated Block Median Decomposition (IBMD) for the above algorithm is chosen since an attempt is made to preserve the median of the data within the blocks as we move across the resolutions of a dyadic multiresolution structure. It is this property of the IBMD which limits the influence of outliers to the highest resolutions and prevents leakage to the lower resolutions. Figure 6.9 illustrates the locations and amplitudes of the IBMD coefficients at different resolutions for 1024 regularly spaced samples of the Heavisine function contaminated with Cauchy noise ($\kappa=0.5$). The reconstructed functions at different resolutions are also included and show clearly that preserving the median (rather than the mean) limits the effects of outliers to the highest resolutions coefficients. This is the key feature of a multiresolution median estimator that allows the development of effective denoising algorithms for non-Gaussian noise.
Figure 6.9 Locations and amplitude of the IBMD transform coefficients at different resolutions

The IBMD algorithm can be used for *irregularly spaced* samples with *arbitrary* size because the algorithm is based on finding the median of samples that lie in a block of the dyadic multiresolution grid. Number and regularity of samples does not limit the algorithm. In the irregular case, the samples distribution is dense in some regions and sparse in other regions. Some of the blocks, especially at higher resolutions, may not contain any samples and they can be
simply dropped out of the pyramid. The coefficients related to empty blocks are considered as zero during the decomposition and reconstruction stages.

6.6 Data Domain Denoising: A New Method Based on the Local Balance of Fit (LBF)

Having developed a "median preserving" transform through the IBMD algorithm we now turn to denoising applications. This can be achieved by operating in the transform domain and seeking to develop a suitable thresholding procedure for the IBMD coefficients. Alternatively, we can attempt to develop a data denoising procedure with all calculations performed entirely in the data domain rather than the transform domain. Development of an effective thresholding procedure for irregularly spaced data and non-Gaussian noise is by no means straightforward. The well known wavelet thresholding technique depends heavily on the regularity of the data and inherent properties of the Gaussian distribution. Donoho and Yu (1997) have succeeded in developing a median preserving decomposition and an effective thresholding procedure for regular data contaminated with a known symmetric noise distribution. Their algorithm, however, imposes a restriction on the size of the data set and employs a triadic pyramidal structure. Ideally, we require a multiresolution denoising procedure that can handle an irregularly spaced data set of an arbitrary size contaminated with an unknown arbitrary noise distributions, such a procedure is more readily forthcoming by denoising in the data domain rather than the transform domain.

To develop a general denoising method capable of handling arbitrarily spaced data contaminated with arbitrary noise, we shall first explore the denoising style carried out by a human operator. The human denoising style is very general as well as very simple. We start by posing a simple question to the reader of this thesis: given the noisy data shown on the left panel of Figure 6.10, draw the best curve through the data set. Having conducted this experiment with several individuals, we conclude that the curves obtained are all remarkably similar. The individual drawing this curve, handles the outliers automatically, needs no information about the distribution of the noise, performs no discernable calculation for noise estimation, and does not care
about the data spacing; yet the curves produced by different individuals do not differ much. It appears that the most basic concept used is to pass the curve as far as possible in between the data points. Incidentally the data in Figure 6.10 was generated by adding Cauchy noise (k=0.02) to 1024 samples of the Heavisine function.

![Figure 6.10 Noisy samples and manually drawn curve](image)

The human observer may miss some of the fine details, but captures the overall shape of the underlying curve very efficiently even in the presence of outliers. This manual curve fitting approach can be considered as an effective "robust" denoising method. In the manual curve fitting procedure we do not use any a priori information about the probability density function of the noise, there is therefore no need to know or guess the noise distribution and its parameters such as its variance. This is of immense importance since, in practical applications, often the only available information is the data and no a priori knowledge of the noise distribution and its parameters is available. It is this highly desirable property which has motivated us to develop a synthetic algorithm to mimic the human denoising style as much as possible. The effortless yet highly effective human denoising style appears to rest on two important principles:
1- Based on the zooming in and out capability of the human eye-brain combination, both the global trend and the local features hidden in the noisy data are recognized and reconstructed simultaneously using wide-smooth and narrow-sharp pieces of curves. This can be considered as a natural multiresolution analysis and reconstruction procedure. To mimic this natural behaviour, we should use a multiresolution system of basis functions.

2- Humans appear to pass the curve through the median of the data points by balancing the number of data points above and below the curve in all local regions. To follow the general trend embedded in the data, however, the degree of balance is adjusted automatically to suit the local interval under consideration. Over a wide interval with large number of data an attempt is made to match the number of points above and below the curve more or less precisely. Over a narrow interval with fewer data points, however, a greater degree of imbalance is tolerated. To mimic this for function estimation from noisy samples, an appropriate "balance criterion" should be defined within a multiresolution structure. This balance criterion should also be gradually modified while proceeding to higher resolutions.

Inspired by the above observations, we develop a new denoising method using the combination of a dyadic multiresolution structure and a local balancing criterion and refer to it as the Local Balance of Fit (LBF) denoising method.

6.6.1 The Local Balance Criterion

The form of the approximating function is not known a priori in denoising applications, the balance criterion must therefore be employed in a constructive manner. In other words, we must use the balance criterion to select a set of particular basis functions in order to control the complexity of the estimated function. The characteristics of the multiresolution structure enable us to use a local measure of balance for constructing the estimate of the true function. Consider the regular dyadic multiresolution structure shown in Figure 6.8 and assume that an estimate of the function at resolution $j$, $F_j(x)$, is available. Let $M_x(j,k)$ denote the number of data points falling in block $B(j,k)$ and define the following vectors:
\( x_{j,k} \) = abscissa of data within block \( B(j, k) \)

\( y_{j,k} \) = ordinate of data within block \( B(j, k) \)

\( \hat{y}_{j,k} \) = vector of estimates within block \( B(j, k) \) at resolution \( j \)

\( i.e. \hat{y}_{j,k}(i) = F_j(x_i) \quad \text{for} \quad x_i \in x_{j,k} \)  \( (6.25) \)

We define a Balance Ratio \( BR(j, k) \) within each block \( B(j, k) \) as follows:

\[
BR(j, k) = \frac{\text{Minimum number of samples located on one side of the estimated function in block } B(j, k)}{\text{Total number of samples in block } B(j, k)}
\]  \( (6.26) \)

To find the balance ratio in each block, the number of samples on the upper side of the curve, \( M_u(j, k) \), can be computed as:

\[
M_u(j, k) = \text{number of points in block } B(j, k) \text{ with } \hat{y}_{j,k} > y_{j,k}
\]  \( (6.27) \)

The minimum number of samples located on one side of the estimated function is therefore,

\[
M_{\min}(j, k) = \min(M_u(j, k), M(j, k) - M_u(j, k))
\]  \( (6.28) \)

The balance ratio within block \( B(j, k) \) can therefore be computed as:

\[
BR(j, k) = \frac{M_{\min}(j, k)}{M(j, k)}
\]  \( (6.29) \)

By definition, the balance ratio varies in a range between 0 and 0.5. \( BR(j, k)=0.5 \) indicates “full” balance and \( BR(j, k)=0.0 \) indicates “no” balance between the points lying on either side of the approximated curve. To explore the characteristics of the balance ratio in the multiresolution structure of blocks, we consider an ideal case where the estimated function coincides with the true underlying function and compute the balance ratio within all blocks for an illustrative example. Figure 6.11 shows the balance ratios at resolutions 0 to 4 for the data of Figure 6.10 based on the number of points falling above and below the underlying Heavisine function.
The first observation is that the balance ratios are non-zero, indicating that there is a degree of balance within each block. The next observation is that the balance ratio is close to 0.5 at lower resolutions, indicating that near full balance is achieved at the lower resolutions. This is to be expected since the number of blocks is doubled in going to a higher resolution but the number of points falling within a block is halved

\[ M(j, k) = \frac{M}{2^j} \]  

(6.30)

The low resolution blocks contain a large number of data points and the probability of an equal number falling on either side of the estimated curve is high. The narrower high resolution blocks, however, contain fewer data points and the probability of an imbalance between the numbers of points on either side of the curve is increased accordingly. For example, for a block with only two data points there is a 50\% chance that both points lie on the same side of the curve giving a balance ratio of zero. It appears, therefore, that the minimum balance ratio at a given resolution \( MBR(j) \)

\[ MBR(j) = \min_k (BR(j, k)) \quad \text{for all } k \]  

(6.31)
can serve as a useful measure of the local balance of fit. The minimum balance ratio at each resolution is shown in red on Figure 6.11. We observe that the maximum balance ratio is around 0.5 at all resolutions but the minimum balance ratio decreases as we go up the resolutions:

<table>
<thead>
<tr>
<th>Resolution j</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBR(j)</td>
<td>0.5</td>
<td>0.48</td>
<td>0.46</td>
<td>0.42</td>
<td>0.39</td>
</tr>
</tbody>
</table>

The balance ratio data is also shown graphically in Figure 6.12, again with the minimum value indicated in red.

![Minimum balance ratio versus resolution](image)

**Figure 6.12** The minimum balance ratio verses resolutions for data of Figure 6.8

Let us assume that we can *somehow* assign a (level dependent) Minimum Acceptable Balance Ratio, $M_{ABR}(j)$, for each resolution. We can then consider that the local balance criterion is satisfied within a block $B(j, k)$ for which

$$BR(j, k) \geq M_{ABR}(j)$$  \hspace{1cm} (6.32)

A local balance ratio $BR(j, k)$ which is smaller than the specified $M_{ABR}(j)$ signifies local over-smoothing and indicates that the data falling in $B(j, k)$ can support a higher
resolution. This simple local balance criterion can be used in a constructive procedure for accepting or rejecting the estimated function in each block of the multiresolution structure. The critical issue of course is the specification of a level dependent MABR(j) curve which is as close as possible to the MBR(j) curve for the (unknown) true underlying function.

6.6.2 Selecting the Minimum Acceptable Balance Ratio (MABR)

Appropriate selection of the MABR is a fundamental issue in the proposed local balance of fit (LBF) approach. Ideally, the specified MABR(j) should correspond to the MBR(j) of the true function. Since the true function is unknown, we shall introduce a procedure for selecting MABR based on a statistical discussion of the true and sample medians. In particular, we shall reduce the selection of the MABR(j) curve to the specification of a single probability computed from a discrete binomial distribution. The proposed selection procedure will be justified through illustrative examples and simulations. For regularly spaced samples, the number of samples in all blocks at a given resolution is the same; a single MABR(j) value then suffices for all blocks at the same resolution. For unequally spaced samples, however, the number of samples in the blocks of a given resolution may differ widely. In general, therefore, the MABR must be chosen based on the number of samples falling in a block. This would unify the treatment of regularly and irregularly spaced data at the expense of seeking a block dependent MABR(j, k).

In robust median estimation it is assumed that samples of the true underlying function \( f(x) \) are contaminated with noise drawn from a distribution with zero median, that is

\[
y_i = f(x_i) + \varepsilon_i \quad \text{for } i = 1, \ldots, M
\]

and

\[
\text{median}\{\varepsilon_i\} = 0 \quad \text{as } M \to \infty
\]

Consider a random variable \( Y \) drawn from a continuous probability density function, \( \rho(y) \), and let \( P(.) \) denote the cumulative probability distribution

\[
P(y) = \int_{-\infty}^{y} \rho(y) \, dy
\]
The continuous median of $Y$, $y_{med}$, is by definition such that:

$$
\int_{-\infty}^{y_{med}} \rho(y) \, dy = \int_{y_{med}}^{\infty} \rho(y) \, dy = P(y_{med}) = \frac{1}{2}
$$

(6.35)

We shall refer to the continuous median as the "true median" and note that its value can only be evaluated if we either know $\rho(y)$ or an infinite number of samples are available. In practice, only a finite number of samples (measurements) of the random variable $Y$ are available and we can only compute the median of the available samples. If we arrange the $M$ available samples in increasing order, $y_1 \leq y_2 \leq \cdots \leq y_M$, the "sample median" is defined as:

$$
y_{sm} = \begin{cases} 
\frac{y_{M+1}}{2} & \text{odd } M \\
\frac{y_M + y_{M+1}}{2} & \text{even } M 
\end{cases}
$$

(6.36)

The sample median $y_{sm}$ is the measured value that is as frequently exceeded as not and is not necessarily equal to the true median $y_{med}$; the two values coincide only if an infinite number of samples is available. Consequently, a deviation between $y_{med}$ and $y_{sm}$ can be expected and any sample $y_r$ may actually be closer to the true median $y_{med}$ than $y_{sm}$. A question can therefore be raised about the probability with which any sample $y_r$ best represents the true median $y_{med}$ of the random variable $Y$. Put another way, given $M$ independent ordered samples drawn from a distribution, we seek the probability that $r$ samples are smaller and the remaining $M-r$ samples are larger than the (inaccessible) true median. This is actually a trial for which only two outcomes (success or failure) are possible and is referred to as the binomial or Bernoulli trial. The probability that $M$ binomial trials will yield exactly $r$ successes and $M-r$ failures is given by the discrete binomial distribution (Meyer, 1975),

$$
P_{M,r} = \binom{M}{r} \alpha^r \beta^{M-r} = \frac{M!}{r!(M-r)!} \alpha^r \beta^{M-r}
$$

(6.37)

where $\alpha$ is the probability for success and $\beta$ is the probability for failure in a given trial. The expected value (mean) and the variance for the discrete binomial distribution are given by
\[ E(r) = \alpha M \]  \hspace{1cm} (6.38)  
\[ \text{var}(r) = \alpha \beta M \]  \hspace{1cm} (6.39)

In the case of the true median, we may set,

\[
\alpha = \Pr(y_i < y_{med}) = \frac{1}{2}
\]

\[
\beta = \Pr(y_i > y_{med}) = \frac{1}{2}
\]  \hspace{1cm} (6.40)

and the binomial distribution (6.37) specialises to

\[
P_{M,r} = \binom{M}{r} \left( \frac{1}{2} \right)^r \left( \frac{1}{2} \right)^{M\!-\!r} = \binom{M}{r} \frac{1}{2^M}
\]  \hspace{1cm} (6.41)

Figure 6.13 shows the discrete binomial distribution (6.41) evaluated for \( M=16 \) independent samples.

![Bionomial Probability Distribution, M=16, alpha=0.5](image)

**Figure 6.13** The binomial distribution of \( M=16 \) independent samples with \( \alpha = \beta = 1/2 \)

In practice we are interested in the probability \( p \) with which at least \( s \) samples lie on one side and at most \( M-s \) samples lie on the other side of the true median \( y_{med} \). This probability can be calculated from the discrete binomial distribution (6.41) using
\[ p = \Pr\left[y_s \leq y_{med} \leq y_{M-s}\right] = \sum_{r=0}^{M-s} P_{M,r} = \sum_{r=0}^{M-s} \binom{M}{r} \frac{1}{2^M} \]  

(6.42)

and can be related to minimum balance ratio MBR. If the estimated value coincides with the true median \( y_{med} \) and \( y_s \leq y_{med} \leq y_{M-s} \), the balance ratio will lie in the range \( s/M \leq BR \leq 1/2 \) and the minimum balance ratio is therefore \( MBR=s/M \). We can also view (6.42) as the probability for the balance ratio BR falling between \( s/M \) and \( 1/2 \) or as the probability for the minimum balance ratio \( MBR=s/M \):

\[ p = \Pr\left[\frac{s}{M} \leq BR \leq \frac{1}{2}\right] = \Pr[MBR = \frac{s}{M}] = \sum_{r=0}^{M-s} \binom{M}{r} \frac{1}{2^M} \]

(6.43)

For a specified number of samples \( M \), a larger \( p \) is equivalent to a wider range for BR or a smaller value of MBR.

We shall find it more convenient to work with the complement of the probability \( p \)

\[ q = 1-p = \Pr[BR < s/M] = 1 - \sum_{r=0}^{M-s} \binom{M}{r} \frac{1}{2^M} = 2 \sum_{r=0}^{M-s} \binom{M}{r} \frac{1}{2^M} \]

(6.44)

which is simply the probability that the balance ratio is less than \( s/M \). Note that a smaller \( q \) corresponds to a smaller minimum balance ratio \( MBR=s/M \). We can assign a likely value to \( q \), say \( q_{low} \), and consider the corresponding \( s/M \) value obtained by inverting the cumulative binomial distribution (6.44) as the Minimum Acceptable Balance Ratio MABR. The MABR value can then be compared with the balance ratio directly computed from the data and the current estimate to check the balance criterion.

The above discussion was focused on the discrepancy of the true and sample medians for a large number of data points all at the same spatial location. In practice we do not have enough samples at each spatial location and the measured values are distributed over the entire input domain. We shall employ the above analysis for seeking out the true underlying function in block \( B(j, k) \) of the multiresolution structure) based on the following premise: if the current estimated function in block
B(j, k) is identical to the true underlying function, the minimum number of data points falling above and below the estimate has the same distribution as (6.44). Let \( M_{\text{min}}(j, k) \) denote the minimum number of samples located on one side of the estimated function in block B(j, k), the corresponding probability \( q_L(j, k) \) can be computed as:

\[
q_L(j, k) = 2 \sum_{r=0}^{M_{\text{min}}(j, k)-1} \left( \frac{M(j, k)}{r} \right) \frac{1}{2^M(j,k)}
\]  

(6.45)

We shall find it more convenient to assign a value for probability \( q_L \), say \( q_{\text{low}} \), and finding the corresponding \( M_{\text{min}}(j, k) \) by inverting the cumulative binomial distribution (6.45). The minimum acceptable balance ratio in block B(j, k) is then simply

\[
M_{\text{ABR}}(j, k) = \frac{M_{\text{min}}(j, k)}{M(j, k)}
\]  

(6.46)

This MABR(j, k) can be compared with the balance ratios BR(j, k)

\[
BR(j, k) = \frac{M_{\text{min}}(j, k)}{M(j, k)}
\]  

c.f. (6.29)

computed from the data and the current estimated function to test the balance criterion in Block B(j, k). If BR(j, k) < MABR(j, k) we may conclude that the balance criterion is not satisfied and the current estimate in B(j, k) is not a good approximation to the true underlying function.

The relationship between \( q_{\text{low}} \) and MABR(j, k) is non-linear but monotonic so that a smaller probability \( q_{\text{low}} \) yields a smaller MABR(j, k). We note here that for a given \( q_{\text{low}} \), the computed \( M_{\text{min}}(j, k) \) decreases super-linearly as \( M(j, k) \) is decreased linearly. The consequence is that for a given \( q_{\text{low}} \), MABR(j, k) decreases as the number of data points in a block \( M(j, k) \) is decreased. This means that a constant \( q_{\text{low}} \) value leads to a decreasing MABR(j, k) as we go up the resolution levels. Figure 6.14 shows that the MABR(j, k) curves generated for \( q_{\text{low}} = 0.1, 0.01, \text{and } 0.001 \) follow the same trend as the minimum balance ratios of Figure 6.12, computed directly from the true Heavisine function and the noisy data of Figure 6.10.
6.6.3 The LBF Algorithm

We are now in a position to present the LBF denoising algorithm. To apply this algorithm we need to perform a suitable dyadic multiresolution decomposition to find the multiresolution coefficients $d$. The LBF algorithm is in fact independent of the decomposition method used to find the multiresolution coefficients. For example, for regularly spaced data contaminated with Gaussian noise we can take full advantage of the highly efficient FWT decomposition. For irregularly spaced data contaminated with Gaussian noise we can resort to least squares wavelet decomposition discussed in Chapter 4. For non-Gaussian noise or data contaminated with outliers, we can use a robust dyadic multiresolution decomposition such as the IBMD method developed in this chapter. As we shall see later, the LBF denoising method is particularly well matched with the IBMD decomposition technique discussed in section 6.5. Without loss of generality, we shall present the LBF denoising algorithm for the IBMD decomposition method. We also note that all
calculations are performed entirely in the data domain and the LBF denoising algorithm is equally applicable to uniformly and non-uniformly spaced data. More significantly, the LBF denoising algorithm is *distribution free* and can be used for any data set contaminated with *independent* but *not necessarily identically* distributed noise.

The key idea behind the LBF denoising algorithm is to compare the local balance ratio $BR(j, k)$ in a block $B(j, k)$ with the minimum acceptable balance ratio $MABR(j, k)$ obtained for a given probability $q_{low}$ to see if the balance criterion is satisfied. However, the satisfaction of the balance criteria in a particular block $B(j, k)$ does not guarantee that the estimated function is locally balanced within the domain of $B(j, k)$ at higher resolutions. To insure that this is the case, we must also check the balance criterions in all the higher resolutions blocks contained within the domain of $B(j, k)$. To this end we shall define a column of blocks $C(j, k)$ as the block $B(j, k)$ and all overlapping higher resolution blocks:

$$C(j, k) = \{ B(m, p) | m = j : L - 1; \ p = 2^{m-j}(k-1) + 1 : 2^{m-j} \}$$  \hspace{1cm} (6.47)

Figure 6.15 illustrates the dyadic multiresolution structure with column $C(2,2)$ and $C(3,7)$ identified by dotted lines. To terminate the estimation locally in block $B(j, k)$ we must first ensure that the balance criterion is satisfied in all blocks of the corresponding column $C(j, k)$. If the balance criterion is not satisfied in any of the blocks belonging to column $C(j, k)$, we proceed locally to the next higher resolution. This procedure is continued until the balance criterion is satisfied in all columns.

Failure to perform the column test can have serious consequences. This is demonstrated in Figure 6.16 which shows the denoising of a data set consisting of 2048 noisy samples of the Heavisine function with added Gaussian noise ($\sigma = 0.5$). The function estimated by checking the balance criterion in blocks only over-smoothes the true underlying function significantly. In contrast, an excellent estimate is obtained when the balance criterion is checked in all blocks of each column.
Figure 6.15 Blocks $B(j, k)$ and columns $C(j, k)$ in a dyadic multiresolution structure

Figure 6.16 Denoising performance of the LBF algorithm with block tests alone and with column tests (2048 samples with Gaussian noise with $\sigma = 0.5$)
The pseudo-code for the LBF denoising algorithm is shown on Figure 6.17. It is assumed that the decomposition and reconstruction are performed using the IBMD method but any other decomposition method can be easily substituted. The multiresolution coefficient vector \( \hat{d} \) is obtained using the IBMD method and a suitable value for probability \( q_{low} \) is specified. We shall also define a vector of selected coefficients \( \hat{d} \) by assigning an indicator \( \text{select}(j, k) \) to each block \( B(j, k) \). This indicator is set to unity, \( \text{select}(j, k) \leftarrow 1 \), if the coefficient associated with \( B(j,k) \), \( d_{j,k} \), is selected and to zero otherwise \( \text{select}(j, k) \leftarrow 0 \). The vector \( \hat{d} \) is therefore identical to the full coefficients vector \( d \) except that the non-selected coefficients are set to zero. The procedure is started by selecting all the coefficients at the coarsest resolution \( J \) and deselecting all the higher resolutions coefficients. The coarsest approximation \( F_{J}(x) \) is obtained using the selected coefficients vector \( \hat{d} \). A resolution loop is then entered and the balance of fit for this estimate is checked locally within an inner column test loop. Within the inner loop we first test to see if a block \( B(j, l) \) at resolution \( j \) is selected. For each selected block \( B(j, l) \), a test is performed in the two higher resolution columns \( C(j+1, 2l-1) \) and \( C(j+1, 2l) \) overlapping the domain of \( B(j,l) \). This is to see if blocks \( B(j+1, 2l-1) \) and \( B(j+1, 2l) \) at resolution \( j+1 \) should be selected or not. The balance of fit calculations and test carried out in a block \( B(m, p) \) are shown by the dotted rectangle on Figure 6.17. The algorithm proceeds to the next higher resolution if at least one block is selected at the higher resolution. The next higher resolution estimate is obtained by adding the detail to the coarse content

\[
F_{j+1}(x) = F_{j}(x) + f_{j}(x)
\]

(6.48)

The algorithm terminates when none of the blocks in the next higher resolution is selected.
Figure 6.17: The IBMD-LBF denoising algorithm

1. Perform IBMD to find coefficients $\hat{d}$ and set a value for $q_{\text{low}}$.
2. Initialize by deselecting all coefficients: $\text{select}(j, k) \leftarrow 0$ and $\hat{d}_{j,k} \leftarrow 0$ for all $j$ and $k$.
3. Select the coefficients at the coarsest resolution $J$ and find the coarsest estimate $\text{select}(J, k) \leftarrow 1$, and $\hat{d}_{j,k} \leftarrow d_{j,k}$ for all $k$.

\[
\text{SR}_J \leftarrow \{(\text{center}_{j,k}, \hat{d}_{j,k}) \text{ for all } k\}, \text{ and } \hat{F}_J(x_i) \leftarrow \text{interp}(\text{SR}_J)_{x_i}
\]

4. For $j = J:1:L-1$
   - Start of Resolution Loop
      - TerminateFlag $\leftarrow 1$
      - For $l = 1:1:2^j$
         - Start of Column Test Loop
            - If select$(j, l) = 1$ Then
               - For $k = 2l - 1, 2l$
                  - Test the balance of fit at all blocks of column $C(j+1, k)$
                  - For $m = j+1:1:L-1$
                     - For $p = 2^{-j}(k-1):1:2^{-j} k$
                        - Perform local balance calculations for block $B(m, p)$
                           $M_{\text{min}}(m, p) \leftarrow$ minimum number of data falling on
                           one side of estimate
                           $M_{\text{min}}^{q_{\text{low}}}(m, p) \leftarrow$ acceptable value of $M_{\text{min}}(m, p)$ for $q_{\text{low}}$
                           - Test the local balance in block $B(m, p)$
                           - If $M_{\text{min}}(m, p) < M_{\text{min}}^{q_{\text{low}}}(m, p)$ then
                              $\text{select}(j+1, k) \leftarrow 1$, $\hat{d}_{j+1,k} \leftarrow d_{j+1,k}$, and TerminateFlag $= 0$
                           Endif
            Endif
         End
      End
   End
   - Test for TERMINATION
   - If TerminateFlag $= 1$ then
      TERMINATE
   Else
      - Find the estimate at level $j+1$
      \[
      \text{SR}_{j+1} \leftarrow \{(\text{center}_{j+1,k}, \hat{d}_{j+1,k}) \text{ for all } k\}, \text{ and } \hat{f}_j(x_i) \leftarrow \text{interp}(\text{SR}_{j+1})_{x_i}
      \]
      $\hat{F}_{j+1}(x_i) \leftarrow \hat{f}_j(x_i) + f_j(x_i)$,
   Endif
End

**end of Resolution Loop**
6.6.4 Selection of the Probability Limit $q_{low}$

The probability limit $q_{low}$ is the only user defined parameter in the LBF denoising algorithm and must be carefully selected. Figure 6.18 shows the performance of the IBMD-LBF denoising algorithm for a noisy Heavisine function for values ranging from $q_{low} = 10^{-6}$ to $q_{low} = 0.5$. The data consists of 4096 regular samples contaminated with i.i.d. Cauchy noise with $k=0.5$. It is clear that for $q_{low} = 10^{-6}$ substantial over-smoothing of the true function (shown in red) occurs. At the other extreme, with $q_{low} = 0.5$, too many high resolution coefficients are selected and the noise is not filtered effectively. For this example, the true function is reconstructed well and the noise is effectively filtered for a $q_{low}$ in the range 0.01 to 0.1.

From a conceptual viewpoint, the value of $q_{low}$ cannot be selected close to 0.5. Setting $q_{low} = 0.5$ implies that there is a 50% probability that we may reject the true function within a block and erroneously proceed to higher resolutions only to capture noise. Taking a global viewpoint, this means that with $q_{low} = 0.5$, we may over-fit in about 50% of the blocks at the highest resolutions. A lower value of $q_{low}$ must therefore be used to increase the probability of accepting the true function in each block. Choosing a very low value, however, increases the probability of accepting many other (smoother) functions on par with the true function. This would cause the premature termination of the LBF algorithm at a low resolution resulting in an over-smoothed estimate.
Figure 6.18 Influence of the probability limit $q_{\text{low}}$ on the performance of the IBMD-LBF denoising algorithm

The above example and heuristic arguments suggest that a suitable range for $q_{\text{low}}$ is probably between 0.01 and 0.1. To confirm this, the performance of the IBMD-LBF denoising algorithm for $q_{\text{low}} = 0.05$ was examined for a variety of test functions. Figure 6.19 shows noisy versions of the four most frequently used test functions employed in denoising studies, between them these four test functions cover many of the features encountered in practical signals. In each case the data is 4096 regular samples contaminated with i.i.d. Cauchy noise with $k$=0.2. The quality of the
denoised test functions obtained using the LBF algorithm with $q_{low} = 0.05$ is also shown on Figure 6.19. The performance of the IBMD-LBF denoising algorithm with $q_{low} = 0.05$ is impressive in all cases considering the nature of the data and the significant outliers present. The performance for the Blocks function can be improved if piecewise constant interpolation is used instead of cubic spline interpolation. The performance for the Bumps and Doppler functions is limited by the sampling rate (1/4096) and may be improved with a higher sampling rate.

Figure 6.19 Performance of the IBMD-LBF denoising algorithm for four test functions with $q_{low} = 0.05$. Data is generated by adding Cauchy noise ($k=0.2$) to the true function shown in red on the right hand panel.
To confirm the suitability of a $q_{\text{low}}$ in the range $0.01 \leq q_{\text{low}} \leq 0.1$, a detailed simulation was conducted to denoise the Heavisine functions by the IBMD-LBF algorithm for 20 different values of $q_{\text{low}}$ in the range $10^{-5}$ to 0.2 and three different sample sizes. The estimation was repeated 100 times with different randomly selected Cauchy noise patterns at each value of $q_{\text{low}}$ to give a statistically reliable estimate of the (average) TMSE. Figure 6.20 shows a graph of the average TMSE versus $q_{\text{low}}$ for each sample size. The curves show a shallow minimum in the range of 0.01 to 0.1, suggesting that a value of $q_{\text{low}} = 0.05$ is a good value for a wide range of sample sizes. Similar results were obtained for the other three test functions and a $q_{\text{low}} = 0.05$ appears to be a good choice for a wide range of functions.

![Figure 6.20](image)

**Figure 6.20** The average TMSE as a function of $q_{\text{low}}$ for denoising of the Heavisine function by the IBMD-LBF algorithm.

The next question to answer is whether the same value of $q_{\text{low}}$ should be utilized at all resolutions. We should also confirm that the LBF algorithm works well with irregularly spaced data. A little reflection shows that a level dependent value
for \( q_{low} \) falls naturally within the multiresolution framework of the IBMD-LBF algorithm. The essential features of the true function are usually captured by the coefficients concentrated at the lower resolutions while the higher resolution coefficients mainly represent the noise in the data. This suggests using a “larger” value of \( q_{low} \) at low resolution so that low resolution coefficients are more easily selected. Conversely, a “smaller” value of \( q_{low} \) should be employed at the higher resolutions, so that the coefficients capturing noise are less readily selected. A simple recipe is to choose a level dependent probability \( q_{low} \) as

\[
q_{low}(j) = \min\left(\frac{c}{2^j}, \frac{1}{2}\right),
\]

(6.49)

Here \( 2^j \) is the number of blocks at resolution \( j \) and \( c \) is a constant, the value of \( c \) determines \( q_{low}(j) \) and affects the level of smoothness of the estimated function.

A simulation was conducted to test the performance of the level dependent \( q_{low}(j) \) for the Blocks and Heavisine test functions using the IBMD-LBF algorithm. The noisy data was synthesized by adding i.i.d. Cauchy noise \( \mathcal{C}(k = 0.5) \) to 4096 randomly spaced data in each case. The estimation was repeated 100 times at 32 separate values for \( c \) in the range 0.005 to 200. Figure 6.21 shows the TMSE statistics obtained as a function of \( c \) and a value of \( c \) in the range of 1 to 100 seems an appropriate choice. A smaller value of \( c \) favours a smoother function and for this reason we shall adopt a conservative level dependent prescription with \( c=1 \),

\[
q_{low}(j) = \min\left(\frac{1}{2^j}, \frac{1}{2}\right),
\]

(6.50)

for the remainder of this chapter. Figure 6.22 shows a typical example of the Blocks and Heavisine functions denoised by the IBMD-LBF algorithm using equation (6.50) for setting the probability limit \( q_{low}(j) \).
Figure 6.21 The TMSE statistics for $q_{low}(j) = \min(c/2^j, 1/2)$ for the IBMD-LBF algorithm (4096 randomly spaced data with additional Cauchy noise, $k=0.5$).

Figure 6.22 The Heavisine and Blocks functions denoised by IBMD-LBF technique with level dependent $q_{low}(j) = 1/2^j$ (4096 randomly spaced data with Cauchy noise ($k=0.5$) added. Samples outside the [-50, 50] range are not shown.)
The LBF algorithm is very general and provides a *distribution free* denoising technique. This suggests that the constant and level dependent prescriptions for selection of the probability $q_{low}$ given above should remain valid for different functions, different noise distributions and different sample sizes. We are unable to provide a rigorous mathematical proof for this claim but can support it through statistical simulations. We can in fact confine our simulations to a zero true function and (pure) Gaussian noise without loss of generality. This is simply because the balance ratio in each block, and hence the minimum balance ratio at each resolution, depend only on the sign of the noise and are independent of its magnitude or distribution.

The results obtained from 100 repetition of 1024 samples of pure Gaussian noise are compressed into a single Figure 6.23. The yellow dots in this figure show the balance ratios obtained in the blocks at each resolution with the minimum balance ratios (MBR) identified in red. The thin blue lines connect the MBR at different resolutions for each of the 100 repetitions. The thick green curve is the MBR curve corresponding to a constant $q_{low} = 0.01$ and the thick black curve is the MBR curve obtained for a level dependent $q_{low} = 1/2^j$. It is clear that both the constant and level dependent recipes for $q_{low}$ provide good representation of the actual MBR curves obtained from simulations. In particular, the level dependent prescription $q_{low} = 1/2^j$ provides a notably better representation at the higher resolutions.

To show that the simple $q_{low}$ recipes works well for different sample sizes, the above simulation was repeated for samples with $M=1024$ and $M=16384$ data points. The results obtained are shown in Figure 6.24 with the blue curves excluded for clarity. It is clear that both prescriptions $q_{low} = 0.01$ and $q_{low} = 1/2^j$ give good representation of the actual MBR curves irrespective of the sample size. Once again the simple level dependent $q_{low} = 1/2^j$ is superior at the higher resolutions. From these simulations we conclude that the level dependent $q_{low} = 1/2^j$ can be used with confidence in all denoising applications.
Figure 6.23 MBR results for 100 repetitions of 1024 samples of pure Gaussian noise

- BR(j, k), - MBR(j), - MBR curve for each repetition
- MBR curve for \( q_{low} = 0.01 \), - MBR curve for \( q_{low} = 1/2^j \)

Figure 6.24 MBR results for 100 repetitions of pure Gaussian noise for M=1024 and M=16384 samples (Legend as in Figure 6.23)
6.7 Comparison of the IBMD-LBF denoising method with other techniques

In this section, we explore the performance of the IBMD-LBF denoising technique through several illustrative examples. To demonstrate the versatility of the proposed denoising method and also to provide a performance comparison with other well-established techniques we shall use examples involving both regularly and irregularly spaced data contaminated with Gaussian as well as non-Gaussian noise. The combination of the IBMD and LBF algorithms can handle all these situations seamlessly with no *a priori* information concerning the distribution of the noise or its variance.

6.7.1 Regularly Spaced Data Contaminated with Gaussian Noise

As the first example we compare the performance of the IBMD-LBF method with methods designed to handle dyadic regularly spaced data contaminated with Gaussian noise. Figure 6.25 compares the typical performance of the IBMD-LBF denoising method with those of the universal hard thresholding and the LGF method combined with FWT decomposition. The data set is 4096 regularly spaced samples contaminated with Gaussian noise with \( \sigma = 0.5 \). The IBMD decomposition employs cubic spline interpolation and the FWT uses the Symlet8 wavelet system. We have used the level dependent \( q_{low}(j) = 1/2^j \) for both the LGF and LBF algorithms. This example was repeated 2000 times with different noise patterns and the detailed statistics are shown in Table 6.3. The qualities of the FWT hard thresholding and the FWT-LGF estimates are slightly better than the IBMD-LBF estimate. This is not surprising because the LBF method does not use any information about the noise while the other methods use the *a priori* known variance of the noise. The TMSE of IBMD-LBF estimation is not much larger than the other methods and the number of basis functions is similar to that for the LGF method.
Figure 6.25 Comparison of the IBMD-LBF method with FWT-universal hard thresholding and FWT-LGF (Data: 4096 regular samples with Gaussian noise with $\sigma = 0.5$)

Table 6.3 Statistics for IBMD-LBF, FWT-universal hard thresholding, and FWT-LGF denoising methods (2000 repetitions)

<table>
<thead>
<tr>
<th>Denoising method</th>
<th>N</th>
<th>TMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>mean</td>
</tr>
<tr>
<td>FWT-hard thresholding</td>
<td>22</td>
<td>27</td>
</tr>
<tr>
<td>FWT-LGF</td>
<td>25</td>
<td>32</td>
</tr>
<tr>
<td>IBMD-LBF</td>
<td>26</td>
<td>34</td>
</tr>
</tbody>
</table>

The presence of a few outliers, however, is sufficient to destroy the performance of denoising methods developed specially for Gaussian noise (mean estimators). In contrast, the performance of the proposed IBMD-LBF method does not change appreciably. The data in Figure 6.26 is the same data set used for Figure 6.25 except that 5 samples were replaced by outliers. The thresholding and LGF estimates suffer from drastic failure while the IBMD-LBF estimate is hardly altered.
6.7.2 Irregular Data Contaminated with Gaussian Noise

We now consider the case of irregularly spaced data contaminated with Gaussian noise, such data can not be handled by simple hard or soft thresholding techniques and we are forced to use either the projection method combined with coefficient dependent thresholding (Kovac and Silverman, 2000) or the LGF method combined with the least squares decomposition. The proposed IBMD-LBF method can handle irregularly spaced data directly. Figure 6.27 compares the performance of the IBMD-LBF denoising method with the LGF and coefficient dependent universal hard thresholding for 1000 randomly spaced samples of the Heavisine function contaminated with Gaussian noise with $\sigma = 0.5$. This is a situation ideally suited to the LGF denoising method and, as expected, the LGF estimate with an \textit{a priori} known $\sigma = 0.5$ provides a better estimate than the other techniques. The IBMD-LBF method, which uses no \textit{a priori} information about the noise, provides an estimate which is not substantially inferior to that obtained from the LGF denoising. This is important since
in practice the variance of the noise must itself be estimated and the LGF performance depends on the estimated variance.

![Graphs showing comparisons between different denoising methods](image)

**Figure 6.27** Comparison of the IBMD-LBF method with coefficient dependent universal hard thresholding and least squares-LGF denoising techniques (Data: 1000 randomly spaced data with Gaussian noise with $\sigma = 0.5$)

Once again, replacing 5 samples of the data set used in Figure 6.27 by outliers deteriorates the thresholding and LGF estimates substantially while the IBMD-LBF estimate is largely unaffected (See Figure 6.28). Estimation of noise variance is more difficult when the data is irregularly spaced or the noise is spatially variable. In addition, the presence of outliers is a frequent practical occurrence. The distribution free IBMD-LBF denoising method offers a robust alternative to other techniques under such conditions.
Figure 6.28 Comparison of the IBMD-LBF method with coefficient dependent universal hard thresholding and least squares-LGF denoising techniques (Data: as in Figure 6.27 but with 5 data points replaced with outliers)

6.7.3 Regular data contaminated with long tail noise

To handle outliers directly, we are forced to assume a long tail distribution for the noise and attempt to preserve the median rather than the mean of the signal for robust estimation. A median preserving procedure was developed by Donoho and Yu (1997) in their MIPT algorithm coupled with specialized thresholding for a symmetric but not necessarily Gaussian noise distribution. It is therefore important to give a careful comparison of our IBMD-LBF denoising method with the MIPT-thresholding technique. The latter technique is restricted to a regular data set with a size which is an integer power of 3 and utilizes a triadic pyramidal structure. In contrast the IBMD-LBF procedure uses a dyadic pyramid, places no restriction on the size of the data set, and can be used with both regularly and irregularly spaced data. More significantly, the thresholds proposed by Donoho and Yu (1997) must be specifically developed for any given symmetric noise distribution. In contrast, the LBF denoising procedure is
effectively *distribution free* and requires no *a priori* assumptions regarding the noise. The top left hand panel in Figure 6.29 shows a data set consisting of \( M=3^8 \) regularly spaced samples of the Heavisine function contaminated with Cauchy noise with \( k=0.5 \). The wide tail of the Cauchy distribution generates data points which deviate significantly from the underlying Heavisine curve, so much that the shape of the underlying function is obscured. The top right panel of Figure 6.29 shows the noisy data falling in the \([-10, 10]\) range. The estimates recovered by the method of Donoho and Yu (1997) and by the IBMD-LBF method are compared in Figure 6.29. It is clear in this particular example that the distribution free IBMD-LBF algorithm provides an estimation which is better than the specialized MIPT thresholding algorithm.

![Image of Figure 6.29](image)

**Figure 6.29** Comparison of the performance of the IBMD-LBF denoising method with the MIPT-thresholding method (Data: \( M=3^8 \) regularly spaced samples with Cauchy noise with \( k=0.5 \)).

The superiority of the proposed IBMD-LBF algorithm can not be justified on the basis of single anecdotal example. To this end the example of the Figure 6.29 was repeated 2000 times with different randomly drawn noise patterns to provide a meaningful comparison. The statistics obtained is summarized in Table 6.4 and is very revealing.
The statistical average TMSE of the MIPT-thresholding method is 0.0236 but the maximum TMSE observed in the 2000 repetitions is extremely large at 2.516. Careful examination of the results for the MIPT-thresholding approach reveals that in 19 cases the TMSE is greater than 0.1, in 3 cases TMSE>0.5, and in 2 cases TMSE>1. In contrast, the TMSE of the IBMD-LBF estimate falls in a very narrow band 0.0112<TMSE< 0.0287 with a statistical average of 0.0188. This confirms the superior stability and robustness of the IBMD-LBF denoising method developed in this study. As an illustration, the estimates recovered for a particular case, where the IMPT-thresholding fails badly, are shown in Figure 6.30.

Table 6.4 Comparison of the statistics for IBMD-LBF method with the MIPT-thresholding method of Donoho and Yu (1997)(2000 repetitions)

<table>
<thead>
<tr>
<th>Denoising method</th>
<th>N</th>
<th>TMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>mean</td>
</tr>
<tr>
<td>MIPT-thresholding</td>
<td>25</td>
<td>33</td>
</tr>
<tr>
<td>IBMD-LBF</td>
<td>32</td>
<td>43</td>
</tr>
</tbody>
</table>

Figure 6.30 Failure of the MIPT-thresholding method for a particular Cauchy noise pattern
6.7.4 Robustness against Outlier Patches

Robustness of a denoising algorithm against outliers is a critical issue in practical applications. In general, "outliers" do not necessarily obey a particular distribution. Methods, such as the MIPT algorithm of Donoho and Yu (1997), which are based on the \textit{a priori} assumption of a particular distribution for the noise, can fail outright in the presence of \textit{arbitrary} outliers. The next example is aimed at emphasizing this point and clearly highlighting the excellent robustness properties of the proposed IBMD-LBF procedure developed in this chapter. The data in figure 6.31 is identical to that in Figure 6.29 except that 3 adjacent outliers (an outlier patch of length 3) are added. The MIPT-thresholding method fails miserably for this data set but the IBMD-LBF estimate is unaffected.

![Figure 6.31 Breakdown of the MIPT thresholding method in the presence of 3 adjacent outliers](image)

Resistance to the presence of "outlier patches" of large length has attracted considerable interest in the literature (Bruce et al, 1994; Davies and Kovac, 2001, and
Kovac, 2002). This is because intermittent failure of measuring devices often leads to "off" points which are adjacent, hence the name "outlier patch". Figure 6.32 compares the performance of the IBMD-LBF procedure with that of the MIPT-thresholding for an outlier patch of length 14 added to the data set of Figure 6.29. The latter technique fails outright whereas the IBMD-LBF procedure is hardly affected, confirming the excellent robustness of the proposed technique.

![Figure 6.32 Robustness of the IBMD-LBF denoising method against an outlier patch of length 14.](image)

6.7.5 Arbitrarily Spaced Data Contaminated with Arbitrary Noise

We now turn to a number of situations which can not be directly handled by the available robust multiresolution denoising techniques such as MIPT-thresholding approach. In particular, we consider examples in which an arbitrary number of data points with arbitrary irregular spacing are contaminated with arbitrary noise. In fact the noise may be spatially variable or different data points within the data set may even be contaminated with noise drawn from quite different distributions. The IBMD-LBF method does not require that the noise is identically distributed; it only requires
that the noise samples are statistically independent. Formally speaking, this reduces the popular assumption of independent and identically distributed (i.i.d) noise to the much weaker assumption of independently distributed (i.d) noise and allows the application of the IBMD-LBF method to a much broader class of denoising problems.

Figure 6.33 shows the application of the IBMD-LBF method for denoising of irregularly spaced data using 5000 randomly spaced samples of the Heavisine function contaminated with Cauchy noise with \(k=0.5\). This data set can not be directly handled by any of the other denoising methods discussed so far.

![Figure 6.33](image)

**Figure 6.33** Application of the IBMD-LBF method for denoising 5000 randomly spaced samples of the Heavisine function contaminated by Cauchy noise \((k=0.5)\)

Figure 6.34 demonstrates the capability of the IBMD-LBF denoising method for handling spatially variable long tail noise with irregularly spaced samples: 5% of the data in Figure 6.34 were selected randomly and contaminated with noise drawn from a Gaussian distribution with \(\sigma = 20\), the remaining 95% of the data were contaminated with a Gaussian noise with \(\sigma = 0.5\). The magnitude of the noise was also weighted with the distance of the data points from the origin, giving a spatially variable noise pattern. The IBMD-LBF estimate for this highly complex situation is shown in Figure 6.34 and provides a good reproduction of the true underlying Heavisine function.
Figure 6.34 Application of the IBMD-LBF method for denoising of a randomly spaced data set contaminated with spatially variable long tail noise.

The proposed IBMD-LBF denoising method can deal directly with even more complex situations. For a final example we consider 6000 regularly spaced samples of the Heavisine signal contaminated with double sided exponential noise with standard deviation $\sigma = 0.5$ in the range $[0, 1/3]$, Gaussian noise with $\sigma = 0.5$ in the range $[1/3, 2/3]$, and Cauchy noise with $k=0.5$ in the range $[2/3, 1]$. The entire data set and an expanded views of the data in the range [-10, 10] are shown in Figure 6.35.

Figure 6.35 Samples of the Heavisine signal contaminated with noise drawn from double sided exponential, Gaussian, and Cauchy distributions in separate regions.
The estimate obtained by applying the IBMD-LBF method with \( q_{\text{low}}(j) = 1/2^j \) to this complicated data set is shown on Figure 6.36 and is a faithful reproduction of the true underlying Heavisine function. The IBMD-LBF denoising method is distribution free, does not require any \emph{a priori} information about the noise, places no restriction on the size of the data set, and can handle regularly and irregularly spaced data seamlessly. These properties, which are not shared by any of the previously reported denoising methods, ensure that the IBMD-LBF denoising method approaches the simple and yet effective human denoising style. In addition, a simple universal prescription for the only adjustable parameter of the IBMD-LBF denoising method is available in \( q_{\text{low}}(j) = 1/2^j \). The IBMD-LBF denoising method is also easily extended to higher dimensions as we shall now demonstrate.

![Figure 6.36](image-url)

**Figure 6.36** The IBMD-LBF estimation of the Heavisine signal from the complex data set of Figure 6.35 using the level dependent prescription \( q_{\text{low}}(j) = 1/2^j \)

### 6.7.6 Extension of the IBMD-LBF Denoising Method to Higher Dimensions

There are no assumptions either in the interpolated block median decomposition (IBMD) algorithm or in the local balance of fit (LBF) algorithm that complicate their extension to two or higher dimensions. We shall only demonstrate the extension of the IBMD-LBF procedure to 2-D data sets, but extension to higher dimensions remains straightforward. The essential point in the development of the 2-D IBMD-LBF procedure is the construction of the 2-D multiresolution structure
shown in Figure 6.37. We shall refer to each “cube” in this structure as block $B(j,k,l)$ addressed by its dilation index $j$ and translation indices $k$ and $l$. The only change necessary for the 2-D IBMD algorithm is that all computations such as median calculations and refinements are performed within 2-D blocks instead of 1-D blocks.

![Image of 3D grid with blocks B(j,k,l)]

**Figure 6.37** Blocks $B(j,k,l)$ identified on a 2-D dyadic multiresolution grid

Interpolation of the medians at each level $j$ is straightforward because the medians are assigned to the centres of the blocks of a regular 2-D structure:

$$F_j(x) = \text{interp2D} (SD_j)$$  \hspace{1cm} (6.51)

Here, the elements of the set $SD_j$ are given by

$$SD_j = \{ (\text{center}_{j,k,l}, m_{j,k,l}) \text{ for all } k \text{ and } l \}$$ \hspace{1cm} (6.52)

with $m_{j,k,l}$ denoting the median of the data within the 2-D block $B(j,k,l)$ with its centre at

$$\text{center}_{j,k,l} = \left( \frac{k-1/2}{2^j}, \frac{l-1/2}{2^l} \right)$$ \hspace{1cm} (6.53)

To extend the LBF algorithm to 2-D data, we need to extend the concept of a column to a 2-D situation. Any block $B(j,k,l)$ with all overlapping higher resolution blocks up to resolution $L-1$ are referred to as a column of blocks $C(j,k,l)$. 

---

229
\[ C(j,k,l) = \{ B(m,p,q) \mid m = j : L - 1; \quad p = 2^{m-j}(k-1) + 1 : 2^{m-j}k; \quad q = 2^{m-j}(l-1) + 1 : 2^{m-j}l \} \] (6.54)

With the above definition in place, the LBF calculations remain the same as that given in Figure 6.17 except that the minimum number of points falling on one side of the 2-D surface, \( M_{\text{min}}(j,k,l) \) is calculated for each 2-D blocks \( B(j,k,l) \).

For the first 2-D example, we shall use a 2-D true function obtained by adding the following local and global features

\[ F(x_1, x_2) = F_{\text{local}}(x_1, x_2) + F_{\text{global}}(x_1, x_2) \]

\[ F_{\text{local}}(x_1, x_2) = 100(1 - 1.100(x_1 + 1) + 0.967(x_1 + 1)^2) \exp \left[ -\left( \frac{x_1 - 0.5}{0.75} \right)^2 \right] \]
\[ \times (1 - 1.100(x_2 + 1) + 0.967(x_2 + 1)^2) \exp \left[ -\left( \frac{x_2 - 0.5}{0.75} \right)^2 \right] \]

\[ F_{\text{global}}(x_1, x_2) = \exp(-0.4(x_1 + 1)) + 0.02 \exp(-1.667(x_2 + 1)) \]
\[ - 0.167(x_1 + 1)(x_2 + 1) \] (6.55)

The same function was used in Chapter 5 in connection with the LGF denoising procedure. A 2-D noisy data set was synthesized by adding noise drawn from a Gaussian distribution with \( \sigma = 0.2 \) to 10000 randomly spaced samples of the underlying function within the unit square. The true function, the locations of the noisy samples, the noisy surface and the estimate recovered using the 2-D IBMD-LBF procedure are shown on Figure 6.38 for comparison purposes. It is clear that the underlying function is captured well except for minor differences around the edges of the unit square. Such differences could possibly be reduced by incorporating specialized treatment of edge effects, which was not pursued in this study.
Figure 6.38 Application of the IBMD-LBF procedure to a 2-D noisy data set (Data: 10000 random samples contaminated with Gaussian noise with $\sigma = 0.2$)

For the next 2-D example, we consider the following simple surface over the unit square

$$F(x_1, x_2) = \sin(2\pi x_1)\sin(2\pi x_2)$$  \hspace{1cm} (6.57)

and contaminate it with spatially variable noise. The noisy data set was generated by selecting 10000 randomly spaced samples and adding a normal noise, whose spread varied as,

$$\sigma_{i,j} = 0.1 x_j(i)$$  \hspace{1cm} (6.58)
Figure 6.39 shows the sample locations, the true function, the noisy data set, and the estimate recovered by the 2-D IBMD-LBF procedure. It is again clear that the true function is recovered well even in the presence of spatially variable noise.

For the final example, we shall use the same true function as in Figure 6.39 but contaminate it with long tail noise (outliers). The noisy data consists of 10000 randomly spaced samples of the true surface: 5% of the data were selected randomly and contaminated with Gaussian distribution with $\sigma = 2.0$, the remaining 95% of the data was corrupted with Gaussian noise with $\sigma = 0.05$. Figure 6.40 shows the sample locations, the true function, the noisy data set and the 2-D IBMD-LBF estimate.
recovered. Once again, the true function is captured well except for minor differences around the edges of the unit square. These few examples are sufficient to confirm that the advantages of the robust and distribution free IBMD-LBF denoising technique extend to two and higher dimensions.

Figure 6.40 Application of the IBMD-LBF procedure to a 2-D noisy data set contaminated with long tail noise (Data: 10000 randomly spaced samples, 5% of the data were selected randomly and contaminated with Gaussian noise with $\sigma = 2.0$, the remaining 95% of the data was corrupted with Gaussian noise with $\sigma = 0.05$)
6.8 Conclusions

The wavelet thresholding techniques are developed on the understanding that the noise is drawn independently and identically from a Gaussian distribution. The new data domain LGF denoising method developed in Chapter 5 retains the Gaussian assumption but can be used for non-identically distributed or spatially variable normal noise. In many applications the noise is inherently non-Gaussian and wavelet thresholding and the LGF method are inappropriate in such cases. The MIPT-thresholding algorithm of Donoho and Yu (1997) is aimed at handling regularly spaced data contaminated with noise drawn independently and identically from an a priori known symmetric distribution under restrictive conditions.

The LBF criterion developed in this chapter is a very general distribution free denoising method that can be used for data contaminated with noise drawn independently from any arbitrary unknown distribution. There is indeed no need for the noise to be identically distributed and the LBF algorithm can even handle data contaminated with noise drawn from several different distributions. This is simply because the LBF criterion depends only on the sign of the added noise at each point and is independent of its magnitude or distribution. The LBF denoising procedure effectively deals with the number of data points falling above and below the current estimate within each block of the multiresolution structure. It is this property which allows the successful application of the LBF denoising technique to data involving outliers, non-Gaussian noise and even non-identically distributed noise. The IBMD-LBF denoising method is easily extended to two and higher dimensions and provides a robust distribution free denoising procedure for a wide class of practical denoising applications.

The LBF algorithm is in fact independent of the decomposition method used to find the multiresolution coefficients. For example, for regularly spaced data contaminated with Gaussian noise we can take full advantage of the highly efficient FWT decomposition. For irregularly spaced data contaminated with Gaussian noise we can resort to the least squares wavelet decomposition discussed in Chapter 4. For non-Gaussian noise or data contaminated with outliers, we can use a robust dyadic
multiresolution decomposition such as the IBMD method developed in this Chapter. The LBF denoising method is particularly well matched with the IBMD decomposition technique, and their combination offers a robust denoising technique for a very wide class of denoising applications.
Chapter 7

Conclusions and Suggestions for Future Work

7.1 Conclusions

Real data are invariably contaminated with a certain measure of uncertainty or noise. The determination of the true trend underlying a set of noisy data is therefore a recurrent problem in diverse areas of science and engineering. The development of denoising procedures has been actively pursued in various fields, in particular signal and image processing and modelling and identification, using a variety of approaches, for example regression and smoothing in the statistics literature, learning theories in the neural network literature and filtering theory in the electrical engineering literature. The majority of previous developments have, however, been aimed at denoising of uniformly spaced data contaminated with independent and identically distributed (i.i.d.) Gaussian noise. Practical data often suffer from data drop out, multirate sampling, sampling jitter or, more generally, may be arbitrarily spaced. The assumption of i.i.d. Gaussian noise, motivated in part by the nice mathematical features of the compact normal distribution and in part by the Central Limit Theorem of statistics, is often poorly realized. Practical data sets often contain “off-points” or “outliers”, caused by instrument or human failure, which are seriously at odds with the compact Gaussian distribution. In other cases, the measurement noise may be inherently non-Gaussian; for example measurements based on counting exhibit a Poisson rather than normal distribution of errors. Furthermore, many of the proposed denoising techniques cannot be easily extended to cover the two and higher dimensional data sets frequently encountered. The ultimate objective of this study was to develop effective denoising procedures capable of dealing with arbitrarily spaced multidimensional data contaminated with arbitrary noise.

The development of denoising procedures has been pursued using diverse but related approaches, for example non-parametric regression in statistics and the learning theories in neural networks. Function estimation provides a general umbrella
that unifies the various theoretical approaches to denoising. The essential elements of function estimation were briefly introduced in general terms in Chapter 2. In particular, the need for the definition of a structured function space, the selection of suitable bases for the structured space, the definition and minimisation of a suitable metric to measure “closeness” to available data, and finally the necessity for a method to control the complexity of the estimated function within a structured space were emphasized.

The multiresolution analysis (MRA), first proposed by Mallat (1989), provides an efficient structured space for function estimation and was introduced in Chapter 3. Wavelets can lend themselves naturally to multiresolution analysis and allow the construction of a suitable structured space for function estimation from a noisy data set. The multiresolution wavelet representation of typical functions is sparse: the essential features of the signal are captured by a few large coefficients whereas the measurement noise is distributed amongst all the wavelet coefficients. It is the combination of properties of wavelets as basis functions and their ability to fit naturally in MRA that has led to an explosion in their application to data compression and denoising. The role of wavelets and scaling functions in Mallat’s multiresolution analysis and the desirable properties of wavelets for time-frequency analysis were considered in Chapter 3. In the case of uniformly spaced dyadic data, the wavelet transform can be obtained through the highly efficient Fast Wavelet Transform (FWT), with obvious advantages in real time applications.

The highly efficient FWT algorithm cannot be used for irregularly spaced data and the special decomposition procedures required were critically reviewed in Chapter 4. Projection of the irregular data onto a regular grid enables the use of the efficient FWT algorithm. Such procedures, however, do not deal with the original data set and their performance is sensitive to the type of projection employed. The lifting scheme, using data adapted second generation wavelets, can deal with irregular data directly but the basis functions employed are no longer simple dilates and translates of a mother wavelet. In addition, the extension of the second generation lifting scheme to two and higher dimension is not straightforward. The least squares wavelet decomposition method offers a procedure that retains the simplicity of the
first generation wavelets, deals with the original irregularly spaced data directly and is readily extended to higher dimensions.

The properties of the least squares wavelet decomposition was thoroughly investigated in Chapter 4 in terms of its reconstruction capability. The assumption of discrete orthogonality is inherent in the commonly employed level by level least squares wavelet decomposition. Discrete orthogonality is lost with irregularly spaced data, even when the wavelet system employed is itself strictly orthogonal. In particular, it was clearly demonstrated that the commonly employed level by level least squares wavelet decomposition can suffer from gross interpolation error in the case of strongly irregular data sets. To our knowledge, this has not been pointed out in previous literature employing the level by level least squares decomposition (e.g. Bakshi and Stephanopoulos 1993, Safavi, 1995, Ford and Etter, 1998), most probably because the data sets employed were only slightly irregular. The gross interpolation error tendency of the level by level least squares wavelet decomposition can seriously compromise the quality of the reconstruction achieved.

An alternative formulation of the least squares wavelet decomposition, where all the scaling and wavelet coefficients are determined simultaneously rather than level by level, was developed in Chapter 4. The simultaneous least squares wavelet decomposition does not suffer from gross interpolation error in the case of highly irregular data sets. Its performance at the higher resolutions, however, is sensitive to the wavelet selection criteria employed. In cases where the ultimate objective is accurate function reconstruction, hybrid decomposition can be employed to combine the high resolution stability of the level by level decomposition with the low resolution accuracy of the simultaneous decomposition. In this study we favour the simultaneous least squares wavelet decomposition for denoising applications. This is because noise is usually high frequency and the denoising algorithm is aimed at removing the high frequency noise. This can be exploited at the same time to reduce the sensitivity of the simultaneous least squares wavelet decomposition to erroneous wavelet selection.

A variety of highly effective denoising procedures have been developed that take full advantage of the sparse representation in the wavelet domain. Transforming
the data into the wavelet domain and setting the small wavelet coefficients to zero (thresholding) offers a simple but powerful method for denoising. A critical review of the more important wavelet thresholding techniques was undertaken in Chapter 5. The majority of reported thresholding procedures are, however, restricted to denoising of uniformly spaced data contaminated with Gaussian noise. Conventional thresholding techniques developed for uniform data perform poorly with non-uniformly spaced data or non-Gaussian noise. The primary objective of Chapter 5 was to develop a denoising technique capable of handling arbitrarily spaced multidimensional data contaminated with Gaussian noise.

A non-uniformly spaced data set can of course be projected onto a regular grid to open the way for the use of some variant of the conventional thresholding techniques. The best developed technique in this area is due to Kovac and Silverman (2000) who combined linear projection with a sophisticated coefficient dependent thresholding procedure. Results obtained in Chapter 5 indicate that such a technique does not perform well on several illustrative examples. The most serious objection to this procedure lies in the fact that we no longer deal with the original data set. The projection based methods are also inevitably sensitive to the type of projection employed. Dealing with projected data correlates the wavelet coefficients and they are no longer independent. In the Kovac and Silverman (2000) implementation, the thresholding is based on the variance of the individual wavelet coefficients and the covariance of the wavelet coefficients is not considered. Including the covariance information may improve the denoising performance but complicates the thresholding further. Major difficulties must also be faced in extending the projection-based methods to two and higher dimensions, this is because interpolation of multidimensional irregularly spaced data is itself a complex problem. Performing the denoising operation in the data domain, rather than the wavelet domain, enables us to deal with the irregular data directly and delivers methods that are readily extended to higher dimensions.

A new data domain denoising technique, the Local Goodness of Fit (LGF) algorithm, was developed in Chapter 5, which employed a local measure of the traditional goodness of fit for controlling the complexity of the estimated function. The LGF denoising algorithm can handle regular and irregular data sets seamlessly.
and can be coupled to any suitable wavelet decomposition method. For example, with uniformly spaced data we can combine the LGF algorithm with the FWT algorithm. For irregular data, the LGF algorithm has inherent synergies with the least squares wavelet decomposition and their combination provides a computationally efficient denoising procedure. The application of the LGF denoising algorithm requires the selection of a single parameter, the probability limit \(q_{\text{low}}\). The selection of this tuneable parameter was also considered and a simple level dependent recipe for its selection was proposed.

A critical comparison of the performance of the proposed LGF algorithm and wavelet domain thresholding methods was undertaken for carefully constructed illustrative examples. The results obtained indicate that the LGF algorithm is competitive with thresholding methods for regularly spaced data. It can, however, deliver a substantially better denoising performance for irregularly spaced data and non-stationary Gaussian noise. This superior performance was confirmed by several large and small simulation exercises to arrive at meaningful statistical conclusions regarding the number of basis functions retained, the true mean squared error delivered and the smoothness properties of the estimated functions. The proposed LGF algorithm is also readily extended to higher dimensions without sacrificing its effectiveness.

The data domain LGF algorithm offers an effective means for denoising of arbitrarily spaced multidimensional data contaminated with stationary or non-stationary Gaussian noise. The application of the LGF algorithm, in common with all other reported denoising procedures, demands an accurate estimate of the variance of the Gaussian noise. Errors in estimation of the noise variance, which can be quite complex in a non-stationary setting, can compromise the denoising performance. More significantly, the performance of the LGF denoising algorithm, in common with all other denoising techniques considered so far, may deteriorate substantially when the data set contains significant “outliers” or the noise distribution has a longer tail than the compact Gaussian distribution. Chapter 6 of this study was devoted to developing an algorithm for denoising of arbitrarily spaced multidimensional data contaminated with an unknown and arbitrary noise distribution.
The conventional wavelet thresholding techniques are developed on the understanding that the data is uniformly spaced and the noise is drawn independently and identically from a Gaussian distribution. The new data domain LGF denoising method developed in Chapter 5 retains the Gaussian assumption but can be used for irregularly spaced data and identically or non-identically distributed (spatially variable) normal noise. In many applications either the noise is inherently non-Gaussian or the data contains significant outliers, both the wavelet domain thresholding techniques and the data domain LGF algorithm are inappropriate in such cases. The potential catastrophic failure of such techniques was clearly demonstrated through a variety of examples in Chapter 6. The roots for the observed failure, however, lie partly in the method of decomposition.

Both the classical discrete wavelet decomposition and the least squares wavelet decomposition are designed to preserve the local mean of the signal across the resolutions. Outliers have a strong influence on the mean and can therefore distort the lower resolution coefficients obtained through a mean-preserving decomposition. Much better denoising performance can be achieved through a multiresolution decomposition aimed at preserving the local median rather than the local mean of the signal. This is simply because the median is far less affected by outliers; a median preserving decomposition is "robust" in the sense that it limits the influence of outliers to the higher resolutions and prevents its leakage to the lower resolutions.

Multiresolution median preserving decompositions have been examined by relatively few authors. In particular, Donoho and Yu (1997) have reported a median preserving decomposition for regularly spaced data under fairly restrictive conditions. A significant undertaking in Chapter 6 was the development a median preserving decomposition, named the Interpolated Block Median Decomposition (IBMD), which can be used with arbitrarily spaced data. The robust IBMD decomposition opens the way for the development of a completely new robust multiresolution data domain denoising algorithm, named the Local Balance of Fit (LBF) algorithm. The LBF algorithm is inspired by the seemingly effortless yet effective way a human operator is able to draw a smooth curve through a noisy data set. An individual drawing such a
curve does not care about data spacing, handles the outliers automatically, performs no discernable calculations for noise estimation and needs no information about the noise distribution.

The intuitive human style denoising appears to be based on two simple inherent abilities:

(i) the multiresolution analysis capability of the human eye-brain system
(ii) the ability to recognize and adjust the local balance of fit

Humans appear somehow able to estimate the probability with which the noisy data points are distributed above and below the unknown true underlying curve. In a locality with a large number of data points, a low probability is assigned for a large fraction of the data points falling on one side of the curve; the true underlying curve is therefore placed close to the local median of the data. In a locality with a few data points, a high probability is assigned for all of the data points falling either above or below the true underlying curve. A large degree of imbalance between the number of data points straddling the underlying curve is therefore tolerated in the regions containing a limited number of data points.

The LBF denoising algorithm developed in Chapter 6 is a heuristic attempt at mimicking the intuitive human denoising style. In particular, through a comparison of the true and the sample medians, it was shown that the probability for a particular fraction of the data points falling on one side of the true underlying curve can be extracted from a discrete binomial distribution. Specifying an acceptable limit for this probability, \( q_{\text{low}} \), enables us to use the binomial distribution for complexity control in a median preserving multiresolution decomposition. This is analogous to the use of \( q_{\text{low}} \) and the chi-square distribution for the mean preserving decompositions considered in Chapter 5. The significant difference is that the median is based only on the sign of the noise and is independent of the details of the noise distribution. The probability limit \( q_{\text{low}} \) is the only tuneable parameter of the LBF algorithm and its specification was considered in some detail. A simple recipe for specification of a level dependent \( q_{\text{low}} \) was presented which, based on simulation results, can be used for general purposes.
A critical comparison of the performance of the IBMD-LBF procedure, classical thresholding methods, the LGF denoising algorithm and the robust denoising method of Donoho and Yu (1997) was undertaken in Chapter 6 for carefully constructed illustrative examples. Several large and small simulation exercises were undertaken to arrive at meaningful statistical comparison regarding the number of basis functions retained, the true mean squared error delivered and the smoothness properties of the estimated functions using the various denoising methods. The results obtained indicate that for regularly spaced data and Gaussian noise, the denoising performance of the proposed IBMD-LBF procedure is not too different from the classical thresholding techniques. A few outliers are, however, sufficient to destroy the performance of the classical thresholding methods; whereas the performance of the robust IBMD-LBF procedure is not affected.

In the case of irregularly spaced data contaminated with Gaussian noise, the performance of the IBMD-LBF procedure is superior to the coefficient dependent thresholding of Kovac and Silverman (2000) and only slightly inferior to that of the least squares-LGF procedure developed specifically for Gaussian noise in this study. Here again, a few outliers are sufficient to seriously compromise the LGF algorithm but the robust IBMD-LBF algorithm remains unaffected. In addition, unlike the thresholding methods and the LGF algorithm, the IBMD-LBF procedure does not require an estimate of the variance of the Gaussian noise. The IBMD-LBF procedure can also handle non-stationary Gaussian noise effectively.

Data contaminated with long-tail noise cannot be effectively denoised by methods developed on the assumption of normal measurement error; these include all the classical thresholding technique, the coefficient dependent thresholding procedure of Kovac and Silverman (2000) as well as the LGF denoising algorithm developed in this study. In the case of regularly spaced data, the robust MIPT-thresholding algorithm of Donoho and Yu (1997) can deal with symmetric long-tail noise, provided the noise distribution and its variance are known a priori. The robust IBMD-LBF procedure developed in this study can deliver a superior performance for long-tail noise, without any a priori knowledge of the long-tail noise distribution. The IBMD-LBF procedure is also substantially more robust against outlier patches,
which can badly compromise the robust denoising method of Donoho and Yu (1997). More significantly, several illustrative examples confirm the effectiveness of the IBMD-LBF procedure for dealing with arbitrarily spaced data contaminated with arbitrary noise; a task that cannot be handled by any of the available denoising procedures. Both the robust Interpolated Block Median Decomposition and the Local Balance of Fit denoising algorithm are readily extended to two and higher dimensions. Illustrative 2-D examples confirm the effectiveness of the combined IBMD-LBF procedure for dealing with high dimensional noisy data sets.

The LBF denoising algorithm developed in this study is a very general *distribution free* robust denoising method that can be used for multidimensional data contaminated with noise drawn independently from any arbitrary unknown distribution. There is indeed no need for the noise to be identically distributed and the LBF algorithm can even handle data contaminated with noise drawn from several different distributions. This is simply because the LBF criterion depends only on the sign of the added noise at each point and is independent of its magnitude or distribution. The LBF denoising procedure effectively deals with the number of data points falling above and below the current estimate within each block of the multiresolution structure. It is this property which allows the successful application of the LBF algorithm to data involving outliers, non-Gaussian noise and even non-identically distributed noise with virtually no *a priori* information on the noise characteristics. The IBMD-LBF denoising procedure provides a robust distribution free denoising technique that should find application in a wide class of practical denoising problems.

7.2 Suggestions for Future Work

A number of areas can be suggested for future studies based on the results obtained in Chapters 4, 5 and 6 of this thesis. The least squares wavelet decomposition considered in Chapter 4 may be worthy of further study for applications where accurate function reconstruction from arbitrarily spaced multidimensional samples, rather than denoising, is the primary objective. The objective in such applications is perfect data interpolation and the irregular lifting
scheme can, in theory, handle the interpolation problem precisely. However, the extension of the irregular lifting scheme to two and higher dimensions is not an easy task. The hybrid least squares decomposition, which is based on first generation wavelets and a regular multiresolution structure, could provide a technique for near perfect interpolation that is readily extended to scattered data in higher dimension.

Both the proposed LGF and LBF data domain denoising algorithms developed in this study require the specification of a probability limit $q_{low}$, which effectively controls the complexity of the denoised function. The level dependent prescriptions offered for setting $q_{low}$ were based on simulation results and appear to work well for a wide range of illustrative examples. Much better denoising performance could, however, be forthcoming if the selection of the probability limit $q_{low}$ is made data dependent. Further studies are needed to develop robust and reliable data dependent procedures for setting the probability limit $q_{low}$.

Perhaps the most fertile area for future work is the extension of the LGF and LBF algorithms to handle correlated noise directly. The fundamental assumption in the LGF algorithm is that the noise samples are independent and normally distributed. The restriction to Gaussian noise was dropped in the robust LBF denoising algorithm, which only requires that the noise samples are independent. However, neither the LGF nor the LBF algorithm can directly handle dependent (or correlated) noise. Further work in this direction seems promising because the chi-square formulation, which lies at the heart of both the least squares wavelet decomposition and the LGF denoising algorithm, can be generalized to correlated noise using the covariance matrix of the data. For the LBF algorithm, however, the correlation information (covariance matrix) can not be used directly. It seems likely, however, that correlations in the noise would affect the form of the discrete binomial distribution and could possibly be reflected in the selection of the probability limit $q_{low}$. It would be an interesting exercise to see if the only fundamental restriction the LBF algorithm, that of independent noise, can be relaxed.

It is worth pointing out that the LBF denoising algorithm is extremely general and can be used with the most suitable decomposition for the data at hand. For
example, with a uniformly spaced data set contaminated with an unknown and complicated spatially variable Gaussian noise distribution, we could employ the robust FWT-LBF procedures. For a non-uniformly sampled data set with spatially variable Gaussian noise, the LBF algorithm can be combined with the least squares decomposition. The case of a non-uniform data set containing outliers or long-tail noise is of course best handled by the IBMD-LBF procedure. There is in fact no restrictions to prevent the use of the LBF denoising algorithm with decompositions based on a non-regular multiresolution structures, for example that arising with the irregular lifting scheme. An attempt to combine the LBF denoising algorithm with the irregular lifting scheme, particularly in higher dimensions, is another area worthy of future study.

Finally, we wish also to acknowledge that the LGF and LBF denoising algorithms developed in this study were based on simple heuristic ideas that were supported on the whole by specific examples and simulations. Future research to provide rigorous mathematical proofs and bounds would undoubtedly increase the confidence in the results of this study.
Appendix

Mathematical Descriptions of Wavelets

Among the various types of designed wavelet basis functions only a limited number have explicit analytical descriptions. Many other wavelet basis functions are defined based on their corresponding filter coefficients and can be constructed using the inverse wavelet transform. The wavelet basis functions used in this thesis are given below. For more detailed information about wavelet basis functions see Daubechies (1992), and Burruse et al (1998).

Wavelet Systems with Explicit Analytical Description

1- Haar wavelet (Figure 3.4): Daubechies (1992)

Haar wavelet

\[ \psi(x) = \begin{cases} 
1 & 0 \leq x < 1/2 \\
-1 & 1/2 < x \leq 1 \\
0 & \text{otherwise} 
\end{cases} \]

Haar scaling function

\[ \phi(x) = \begin{cases} 
1 & 0 \leq x < 1 \\
0 & \text{otherwise} 
\end{cases} \]

2- Mexican hat wavelet (Figure 3.11): Daubechies (1992)

Second derivative of Gaussian

\[ \psi(x) = \frac{2}{\sqrt{3}} \pi^{-1/4} (1 - x^2)e^{-x^2/2} \]

Gaussian as scaling function

\[ \phi(x) = 2.3766 e^{-x^2/2} \]

3- Cubic B-spline wavelet (Figure 3.11): Unser et al (1992)

\[ \psi(x) = \frac{4a^4}{\sqrt{8\pi}} \sigma_w \cos(2\pi f_0(2x-1)) \exp\left(-\frac{(2x-1)^2}{8\sigma_w^2}\right) \]

\[ a = 0.697066, \quad f_0 = 0.409177, \quad \sigma_w^2 = 0.561145 \]

\[ \phi(x) = \begin{cases} 
\frac{x^2}{6} & 0 < x \leq 1 \\
\frac{[x^2(2-x) + x(x-1)(3-x) + (x-1)^2(4-x)]/6}{(4-x)^2/6} & 1 < x \leq 2 \\
\frac{[(x(3-x))^2 + (x-1)(3-x)(4-x) + (x-2)(4-x)^2]/6}{(4-x)^2/6} & 2 < x \leq 3 \\
\end{cases} \]
Wavelet Systems Defined by their Filter Coefficients

1- Daubechies wavelets (Figure 3.10): filter coefficients of daub2, Daubechies (1992)

\[ h = [ .482962913145 \quad .836516303738 \quad .224143868042 \quad -.129409522551 ] \]

2- Symlet wavelets (Figure 3.10): filter coefficients of symlet8, Daubechies (1992)

\[ h = [ \quad .002672793393 \quad -.000428394300 \quad -.021145686528 \quad .005386388754 \ldots \\
.069490465911 \quad -.038493521263 \quad -.073462508761 \quad .515398670374 \ldots \\
1.099106630537 \quad .680745347190 \quad -.086653615406 \quad -.202648655286 \ldots \\
.010758611751 \quad .044823623042 \quad -.000766690896 \quad -.004783458512 ] \]
References


