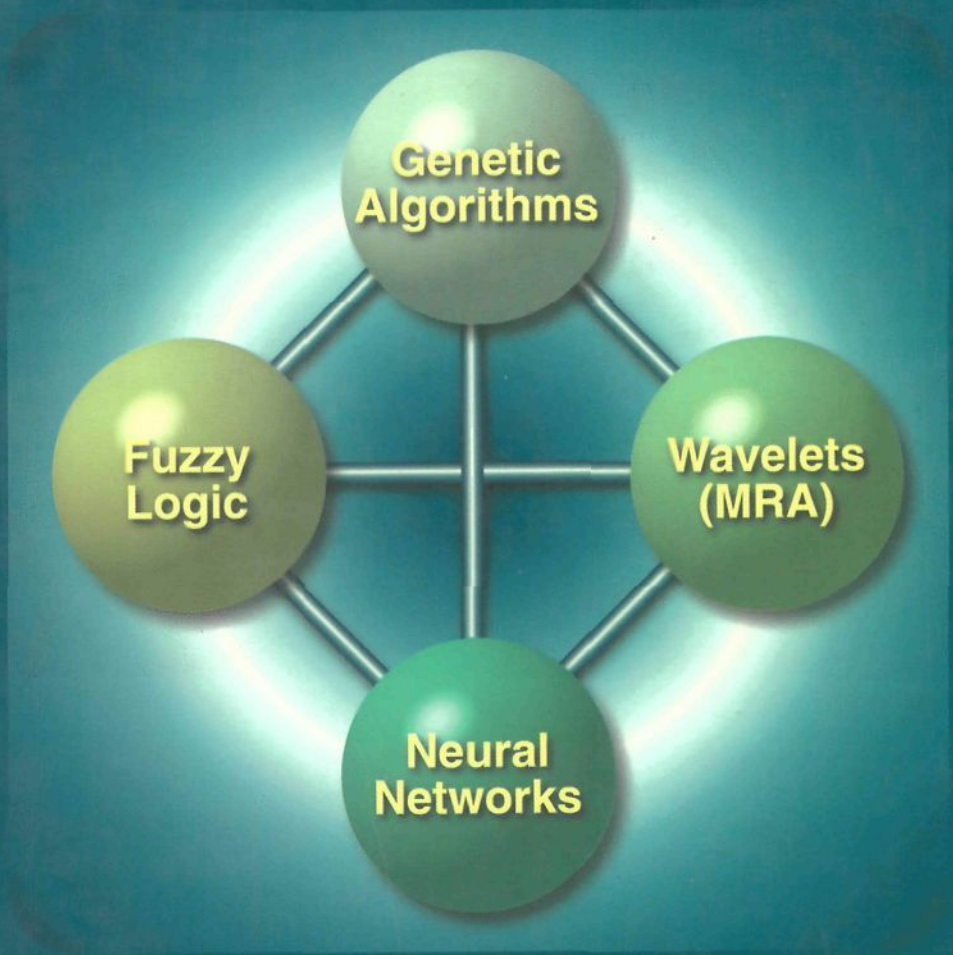
World Scientific Series in Robotics and Intelligent Systems – Vol. 25

WAVELETS IN SOFT COMPUTING

MARC THUILLARD



World Scientific

WAVELETS IN SOFT COMPUTING

WORLD SCIENTIFIC SERIES IN ROBOTICS AND INTELLIGENT SYSTEMS

Editor-in-Charge: C J Harris (*University of Southampton*)
Advisor: T M Husband (*University of Salford*)

Published:

Vol. 10: Cellular Robotics and Micro Robotic Systems
(*T Fukuda and T Ueyama*)

Vol. 11: Recent Trends in Mobile Robots (*Ed. Y F Zheng*)

Vol. 12: Intelligent Assembly Systems (*Eds. M Lee and J J Rowland*)

Vol. 13: Sensor Modelling, Design and Data Processing for Autonomous Navigation
(*M D Adams*)

Vol. 14: Intelligent Supervisory Control: A Qualitative Bond Graph Reasoning
Approach (*H Wang and D A Linkens*)

Vol. 15: Neural Adaptive Control Technology (*Eds. R Zbikowski and K J Hunt*)

Vol. 17: Applications of Neural Adaptive Control Technology (*Eds. J Kalkkuhl,
K J Hunt, R Zbikowski and A Dzielinski*)

Vol. 18: Soft Computing in Systems and Control Technology
(*Ed. S Tzafestas*)

Vol. 19: Adaptive Neural Network Control of Robotic Manipulators
(*S S Ge, T H Lee and C J Harris*)

Vol. 20: Obstacle Avoidance in Multi-Robot Systems: Experiments in Parallel
Genetic Algorithms (*M A C Gill and A Y Zomaya*)

Vol. 21: High-Level Feedback Control with Neural Networks
(*Eds. F L Lewis and Y H Kim*)

Vol. 22: Odour Detection by Mobile Robots
(*R. Andrew Russell*)

Vol. 23: Fuzzy Logic Control: Advances in Applications
(*Eds. H B Verbruggen and R Babuska*)

Vol. 24: Interdisciplinary Approaches to Robot Learning
(*Eds. J. Demiris and A. Birk*)

World Scientific Series in Robotics and Intelligent Systems – Vol. 25

WAVELETS IN SOFT COMPUTING

MARC THUILLARD
Siemens Building Technologies
Switzerland

World Scientific
Singapore • New Jersey • London • Hong Kong

Published by

World Scientific Publishing Co. Pte. Ltd.

P O Box 128, Farrer Road, Singapore 912805

USA office: Suite 1B, 1060 Main Street, River Edge, NJ 07661

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

WAVELETS IN SOFT COMPUTING
World Scientific Series in Robotics and Intelligent Systems — Vol. 25

ISBN 981-02-4609-9

Printed in Singapore by Uto-Print

To Claudia, Estelle and Xavier

This page is intentionally left blank

Foreword

The main goal of *Wavelets in Soft Computing* is to furnish a synthesis on the state of integration of wavelet theory into soft computing. Wavelet methods in soft computing can be classified into 5 main categories that form the backbone of the book:

-Preprocessing methods
-Automatic generation of a fuzzy system from data
-Wavelet networks
-Wavelet-based nonparametric estimation and regression techniques
-Multiresolution genetic algorithms and search methods.

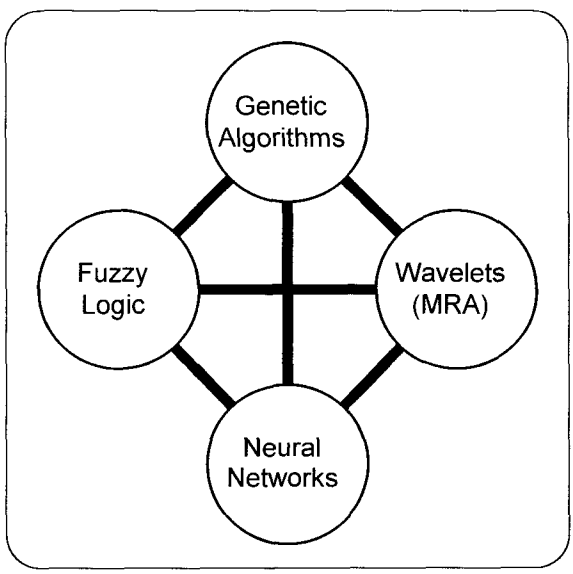
The main new contributions of *Wavelets in Soft Computing* to these topics are in the domain of the automatic generation of a fuzzy system from data (fuzzy-wavelet, fuzzy wavenets for on-line learning), wavelet networks and wavelet estimators (extension to biorthogonal wavelets) and multiresolution search methods. These new methods have been either implemented in commercial fire detectors or used during development. Despite the fact that over 2000 articles have combined elements of wavelet theory to soft computing, no book has been dedicated to that topic yet. The topic has grown to such proportions that it is not possible anymore to offer an exhaustive review. For that reason, the emphasis is placed on topics, that are not specifics to a particular application. A special place is given to methods that have been implemented in real world applications. This is especially the case of the different techniques combining fuzzy logic, neural networks to wavelet theory. These methods have been implemented during the development of several products and have found applications in intelligent systems, such as for instance in fire detection.

Industry is certainly one of the driving force behind soft computing. In many industrial products, very extensive computations are not feasible, either because it would make the product too costly, too slow, or sometimes the limitation may simply be the power consumption as for instance in devices powered with batteries. In spite of all this limitations, many products, such as for instance sensors, require complex algorithms for data processing. This is where soft computing finds one of its best field of applications.

Multiresolution analysis and wavelet theory are a natural complement to soft computing methods. Soft computing deals with solving computationally intensive problems with a limited amount of computing power and memory by giving up some of the precision. Multiresolution analysis can be used to determine how and

where to give up the precision. Also several standard methods in multiresolution analysis could be easily classified as being part of soft computing. This is the case of algorithms such as the *matching pursuit* or of some wavelet-based regression and denoising methods.

Multiresolution analysis is of central importance in the mechanisms of perception and decision. Humans are particularly good at such tasks. For instance, image processing in the brain relies heavily on the analysis of the signals at several levels of resolution. Extracting details of importance out of a flow of information is an essential part of any decision process. Soft computing covers a range of methods that are somewhat tolerant of imprecision, uncertainty and partial truth. Hybrid methods combining soft computing methods to wavelet theory have therefore the potential to accommodate two central elements of the human brain, the capability of selecting an appropriate resolution to the description of a problem and to be somewhat tolerant to imprecision.



The main goal of this book is to present the state of integration of wavelet theory and multiresolution analysis into soft computing, represented here schematically by three of its main techniques.

The success of wavelet theory and multiresolution analysis can be explained by different factors. Wavelet theory offers both a formal and practical framework to understand problems that require the analysis of signals at several resolutions. Despite the fact that several aspects of multiresolution analysis did precede the development of wavelet theory, wavelet theory furnishes an unifying platform to the discussion of multiresolutional signal processing. This is certainly one of the great merit of wavelet theory. From the point of view of applications, wavelet analysis possesses the very nice feature to be easily implementable.

There are a number of excellent books on wavelet theory (see for instance Chui, 1992; Daubechies, 1992; Kaiser, 1994; Mallat, 1998; Meyer, 1992; Vetterli, 1995; Wickerhauser, 1994). In this book, we have deliberately chosen the view of presenting wavelet theory quite pragmatically. Theory is limited to the minimum necessary to understand the ideas behind wavelet theory in order to apply them correctly. The same holds for the different soft computing techniques.

Learning is a central theme to that book. A significant development in recent years has been the recognition of the complementarity and similarities existing between neural network, wavelets analysis and fuzzy logic. The degree of maturity of the different hybrid techniques combining two or more soft computing methods is quite different. On the one hand, neurofuzzy has been used in numerous industrial projects. On the other hand, the combination of wavelet theory with fuzzy logic is emerging: only a few products using fuzzy-wavelet techniques are now commercialized. Excellent books on neurofuzzy techniques have been written (see for instance Brown and Harris, 1994; Babuska, 1998; Jang, 1997; Kosko, 1992; Nauck, 1992; Nie and Linkens, 1995; a large range of applications can be found in recent proceedings of EUFIT or TOOLMET). As our approach follows the line of the book by Brown and Harris (1994), we refer especially to that book for an introduction to neurofuzzy. A main theme is the integration of wavelet theory into neurofuzzy methods. These hybrid techniques are referred either as fuzzy wavenets or as fuzzy wavelet networks, depending on the details of the applied method.

OVERVIEW OF THE BOOK

Wavelet theory is presented in a self-contained manner, adding new concepts as they become necessary to the comprehension of the different hybrid methods combining wavelet theory to soft computing.

Part 1 presents wavelet theory from different complementary perspectives. It explains first wavelet theory in simple terms by discussing the differences and the similarities between Fourier analysis and wavelet theory. In particular, the short-time Fourier transform is compared to the wavelet transform. Fundamental definitions (wavelet, orthogonality, biorthogonality, multiresolution, nested spaces) are given. After having introduced multiresolution analysis from the mathematical perspective, the signal processing approach, also called subband coding, is presented. The most important algorithm, the fast wavelet decomposition algorithm, is then presented in the framework of filter theory. It is shown that a wavelet decomposition can be carried out using a cascade of filters. The final sections present a number of examples showing the power of wavelet analysis for data compression, data analysis and denoising. More recent developments of wavelet theory, for instance the lifting scheme or nonlinear wavelets, are presented gradually in the following parts.

The majority of publications on applications of wavelet analysis in soft computing are in the domain of preprocessing. Wavelet preprocessing has been

used in a large number of different applications, from signal denoising, feature extraction to data compression. Part 2 is dedicated to wavelet theory in preprocessing. The first sections focus on two central problems in signal processing: the curse of dimensionality and the complexity issue. The curse of dimensionality is an expression that characterizes the fact that the sample size needed to estimate a function grows often exponentially with the number of variables. The complexity issue refers to the increase of the computing power to solve hard problems with many inputs. In hard problems in many dimensions, both the curse of dimensionality and the complexity issues are relevant and one can speak of a double curse. Different methods for reducing the dimension of an input space are briefly presented. In particular, the classical dimension reduction based on the Karhunen-Loève transform is explained An important section discusses the contributions of wavelet theory to dimension reduction. The two main methods, the matching pursuit and the best basis are presented. Wavelet theory also finds applications to exploratory knowledge extraction to discover nonlinear interactions between variables or non-significant variables. In the last sections, a number of representative applications combining wavelet preprocessing and soft computing are reviewed with an emphasis on classification problems and applications to intelligent sensing.

Parts 3-6 are dedicated to wavelet-based methods that are suited to the automatic development of a fuzzy system from data. It introduces first the reader to off-line methods for data on a regular grid (fuzzy-wavelet). In subsequent parts, on-line learning schemes are explained in the framework of wavelet-based neural networks and nonparametric wavelet-based estimation and regression techniques. Part 3 gives an overview of wavelet-based spline approximation and compression algorithms. Part 3 is a pre-requisite to part 4-6 on learning. After an introduction on splines, the main families of spline-based wavelet constructions are presented. Emphasis is set on approximation and compression methods based on the matching pursuit algorithm and wavelet thresholding.

Part 4 explains the connection existing between wavelet-based spline modeling and the Takagi-Sugeno fuzzy model in so-called fuzzy-wavelet methods. It is shown how wavelet modeling can be used to develop a fuzzy system automatically from a set of data on a regular grid. One starts from a dictionary of pre-defined membership functions forming a multiresolution. The membership functions are dilated and translated versions of a scaling function. Appropriate fuzzy rules are determined by making a wavelet decomposition, typically with B-wavelets and keeping the most significant coefficients. Part 4 treats a number of issues central to the application of fuzzy-wavelet methods in applications (boundary processing, interpretability and transparency of the fuzzy rules).

Part 5 is on wavelet networks. After a presentation of wavelet networks and wavenets and their applications, the methods in part 4 are extended to on-line learning. The resulting multiresolutional neurofuzzy method permits to determine and validate fuzzy rules on-line with very efficient algorithms using elements of

wavelet theory. The data are modeled by an ensemble of feedforward neural networks using, each, wavelet and scaling functions of a given resolution as activation function. Rules are validated, on-line, by comparing the results at two consecutive resolutions with the fast wavelet decomposition algorithm. When only few datapoints are known, the fuzzy rules use low resolution membership functions. With an increasing number of points, a larger number of rules are validated and the fuzzy system is refined by taking higher resolution membership functions. Different approaches are explained, that have in common to be easily implementable.

Part 6 presents an alternative method to the neural network approach using nonparametric wavelet-based estimation and regression techniques. After an introduction on (orthogonal) wavelet estimation and regression techniques, we show how to extend these techniques to biorthogonal wavelets. The main motivation is that it makes possible to implement these regression techniques to determine appropriate fuzzy rules describing an incoming flow of datapoints. Advantageous in that technique is that the datapoints do not have to be stored and also that the wavelet-based validation methods described in part 5 are still applicable.

Part 7 discusses pragmatically our experience with wavelet-based learning techniques. Some reflections are made on how to develop optimally intelligent products using multiresolution-based fuzzy techniques. We explain how important it is to keep at all time the *man in the loop*. It is of vital importance to have a clearly defined interface between the computer assisted learning method and the development team that allows the verification of all automatically generated rules. We show further how template methods can be applied to compare and validate the knowledge of different experts. These methods facilitate the fusion and the comparison of information from different sources (human, databank,...). With the help of this computer tool, new rules can be proposed to reconcilate conflicting experts.

Part 8 explores the connections existing between genetic algorithms and wavelets. In the first sections, the classical discussion on deceptive functions, based on Walsh partition functions, is reformulated within the framework of multiresolution analysis. In the following sections, a very simple genetic algorithm is presented. The algorithm uses a single operator that tries to catch into a single operator some of the main features of the crossover and mutation operators in the standard genetic algorithm. Analytical results on the expectation of the population are expressed in terms of the wavelet coefficients of the fitness function. This simple model permits to discover some important relationships between sampling theory and multiresolution analysis. At the end of this part, the model is generalized to multiresolution search methods.

This page is intentionally left blank

Contents

This page is intentionally left blank

Acknowledgements

I would like to thank particularly C.J. Harris. I have been very fortunate to co-chair with him the technical commitee on control and monitoring of the European network of excellence ERUDIT. His work has been a constant inspiration over the years. This book has been initiated by a series of tutorials that I have held during EUFIT'98 and EUFIT'99. I would like to thank K. Liewen and H.-J. Zimmermann for their kind invitation. I must also thank very particularly E. Juuso for his very pertinent comments and his precious inputs. Finally I would like to thank my colleagues at Cerberus and in particular its technical director G. Pfister for his support.

PART I INTRODUCTION TO WAVELET THEORY

This page is intentionally left blank

1. Introduction to Wavelet Theory

Marc Thuillard

Siemens Building Technologies AG, Cerberus Division

A short overview on the development of wavelet theory

Wavelet analysis started in the 80's. Scientists processing recordings of seismic waves recognized the need for methods allowing the analysis of signals at different resolutions. In the 90's, multiresolution analysis had grown into a very active field, with the appearance of very efficient computing methods.

Multiresolution analysis has become a quite standard tool in signal processing. Wavelet theory has been applied to basically all scientific fields, including fields as different as quantum mechanics, econometrics or social sciences. Despite the large variety of wavelet applications, the main domain of applications is still in image processing. The image processing community has been using algorithms containing elements of multiresolution analysis for already quite some years. The new standard JPEG 2000 is for instance based on wavelet data compression schemes.

Historically, one generally finds the roots of wavelet theory in the work of Morlet, a scientist by Elf-Aquitaine, who worked in the domain of oil research. Morlet recognized the need for signal processing techniques going beyond Gabor analysis of short-time signals. Morlet modified the Gaussian window used by Gabor. In order to palliate to a drawback of Gabor's approach, namely the bad resolution obtained at high frequencies due to the constant window-size, Morlet used variable-sized windows. Morlet tagged the name wavelet, meaning little wave (Burke Hubbard, 1996). Due to lack of funding and interest by his company, no real-world applications appeared then. Grossmann grasped rapidly the potential of Morlet' s wavelet and contributed significantly to further developments.

After the pioneering work by Morlet and Grossmann (Grossmann, 1984), the next major development was the axiomatic formulation of wavelet theory. This work was mostly carried out within the mathematical community. The development of wavelet theory represents a good example of the importance of cross-fertilization between different fields in science. A first illustration is given

by the development of perfect reconstruction filters within the filter theory community. Perfect reconstruction allows a signal to be split into downsampled subband signals and then reconstructed perfectly. It was later shown that subband coding and wavelet theory are essentially equivalent. The equivalency between subband coding and wavelet theory has permitted the development of efficient algorithms for wavelet decomposition and reconstruction. Possibly as important, it offered a view on multiresolution analysis that was more familiar to the signal processing community, than the mathematicians' approach.

One can trace the take off of wavelet methods in signal processing to the creation of the fast wavelet decomposition algorithm. The discovery of a fast wavelet decomposition and reconstruction algorithm marks the beginning of a new era for wavelets. The fast wavelet decomposition algorithm allows for an efficient computation of the wavelet coefficients using a cascade of filters. This algorithm, originally proposed by Mallat (1989), reduces very considerably the computing burden of a wavelet transform. A fast wavelet decomposition consists of the iterative decomposition of a signal into a coarse and a detail approximation. The original signal can be reconstructed with a second algorithm. The possibility of reconstructing the signal after decomposition has resulted in several applications in the domain of noise reduction and data compression. A full wavelet decomposition is invertible in O(N) operations, making the wavelet transform well suited to lossless compression of a signal.

Recently, wavelets of the second generation have appeared. They are more flexible and permit to solve important problems, such as the representation of a signal on an irregular grid or on a sphere. Second generation wavelets are closely related to multiresolution methods used in computer graphics. An important asset of second generation wavelets is that they provide a geometric interpretation of wavelet theory. Also second generation wavelets have lead to elegant solutions to the problem of endpoints processing, that has plagued wavelet methods for years.

New developments have shown the utility of wavelet theory and multiresolution analysis in the domain of soft computing. Soft computing is a fast developing field in computer science. It deals with solving computationally intensive problems with a limited amount of computing power and memory by giving up some of the precision. Soft computing is also often defined by the techniques it uses, such as neural network, fuzzy logic, genetic algorithms and to some extent multiresolution analysis. One of the major developments in recent years has been the recognition of the complementarity and similarities existing between neural network, genetic algorithms, wavelets analysis and fuzzy logic. Soft computing and multiresolution are by many aspects very complementary. Everyday experience, teaches that the difference between two actions lies often in small details. Finding the important details is difficult, since experience also shows that focusing only on details leads to a tremendous waste of time and unproductive works. Finding the right balance between details and coarse features or actions is a highly human activity, that finds its mathematical expression in the combination of wavelet theory and soft computing.

In part 1, the wavelet transform is introduced as an extension of the Fourier transform, following here the historical development of wavelet theory. Wavelet theory has been originally developed to process non-stationary signals encountered in seismic waves. Before wavelet theory, non-stationary signals were essentially processed with the short-time Fourier transform, also called the Gabor transform. Wavelet theory offers a very flexible alternative to the short-time Fourier transform. A wavelet decomposition is often compared to a mathematical microscope (fig. 1.1). A signal is analyzed at different resolution levels and each level corresponds to a magnification factor.

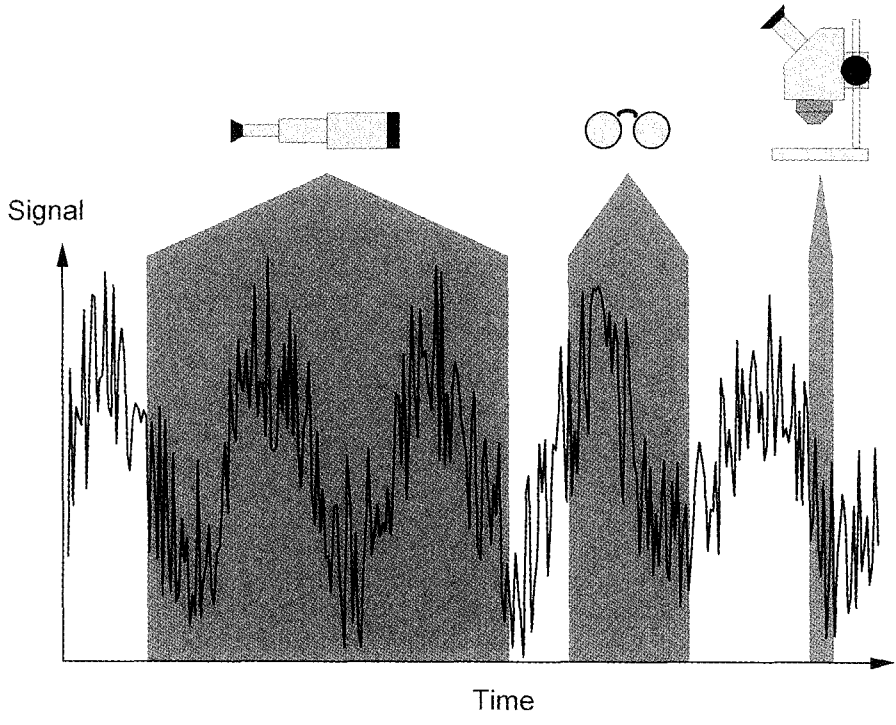


Figure 1.1: The wavelet decomposition of a signal corresponds to a multiresolution analysis of the signal. A wavelet decomposition can be considered as a kind of mathematical microscope.

The analogy between wavelet theory and Fourier theory offers a simple framework to grasp the main ideas behind multiresolution analysis and wavelet theory. Both a discrete wavelet transform and a continuous wavelet transform can be defined. The discrete wavelet decomposition of a signal corresponds to the projection of a signal on a series of translated and dilated versions of a wavelet. The analogy to the discrete Fourier transform is clear as one recalls that the Fourier coefficients in the discrete Fourier transform correspond to the projection of a signal onto a series of dilated sine and cosine (the cosine is a translated version of the sine function!).

Lately, major efforts have been undertaken to develop nonlinear multiresolution construction methods. Nonlinear constructions are being used mostly for lossless image processing. Nonlinear operators are introduced that round off the wavelet and approximation coefficients on integers, permitting an efficient storage of the coefficients. These nonlinear multiresolution projection methods have the particularity to be invertible.

In this first part, we will present the standard wavelet approach based on the analogy to the Fourier formalism. Wavelet theory is then explained more formally on the basis of the axiomatic formalism of multiresolution analysis and in the framework of subband coding. Second generation wavelets and nonlinear constructions are discussed in the annex of the book (Annex A-B). At the end of the chapter, a number of standard and historical applications of wavelet theory are discussed. One of the first applications of multiresolution analysis was in the domain of data compression. Multiresolution techniques were successfully implemented to compress the FBI fingerprint datafiles. An early application of multiresolution analysis in the domain of noise reduction has been the processing of the only recording of Brahms playing a sonate (Berger, 1994). The recording was of such bad quality that transposing the music was not possible. After processing with multiresolution techniques, it became possible to compare Brahms partition with its own interpretation.

Wavelet transform versus Fourier transform

Fourier series

Wavelet theory can be considered as an extension of Fourier theory. In a discrete Fourier decomposition, a periodic signal is represented by a weighted sum of sine and cosine. The coefficients of the sine and cosine correspond to the projection of the signal on sine and cosine. More precisely, the projection on the sine and cosine is an orthogonal projection. The projection is therefore unique and the functions $\{\sin(k \cdot i), \cos(k \cdot i)\}$ form an orthonormal basis.

A square-integrable periodic signal of period T can be decomposed into a sum of sine and cosine:

$$f(t) = \sum_k a_k \cdot \sin(k \cdot i) + b_k \cdot \cos(k \cdot i) \qquad \textbf{(1.1a)}$$

$$a_k = 2\int_0^T f(t) \cdot \sin(2 \cdot \pi \cdot k \cdot t) \cdot dt \; ; \; b_k = 2\int_0^T f(t) \cdot \cos(2 \cdot \pi \cdot k \cdot t) \cdot dt \qquad \textbf{(1.1b)}$$

The functions $\{\sin(k \cdot i), \cos(k \cdot i)\}$ form an orthonormal basis:

$$\int_0^1 \sin(m \cdot 2 \cdot \pi \cdot x) \cdot \sin(n \cdot 2 \cdot \pi \cdot x) \cdot dx = 0, m \neq n \quad \textbf{(1.2a)}$$

$$\int_0^1 \cos(m \cdot 2 \cdot \pi \cdot x) \cdot \cos(n \cdot 2 \cdot \pi \cdot x) \cdot dx = 0, m \neq n \quad \textbf{(1.2b)}$$

$$\int_0^1 \sin(m \cdot 2 \cdot \pi \cdot x) \cdot \cos(n \cdot 2 \cdot \pi \cdot x) \cdot dx = 0, m \neq n \quad \textbf{(1.2c)}$$

The Fourier decomposition possesses the important property to be invertible. In many cases, a limited number of coefficients is sufficient to compute a good approximation of the signal. Fourier analysis can be extended to wavelet theory. Instead of projecting the signal on sine and cosine, the signal can be projected on another set of orthogonal functions. This permits to analyze non-periodic signals. Figure 1.2 shows an example of such a function, a so-called Daubechies wavelet (Daubechies, 1992). The function is well localized and has a zero integral. Such a function is called a mother wavelet. We will show later that a signal can be projected on a series of dilated and translated versions of a mother wavelet. Similarly to the Fourier transform, the projection is unique and invertible.

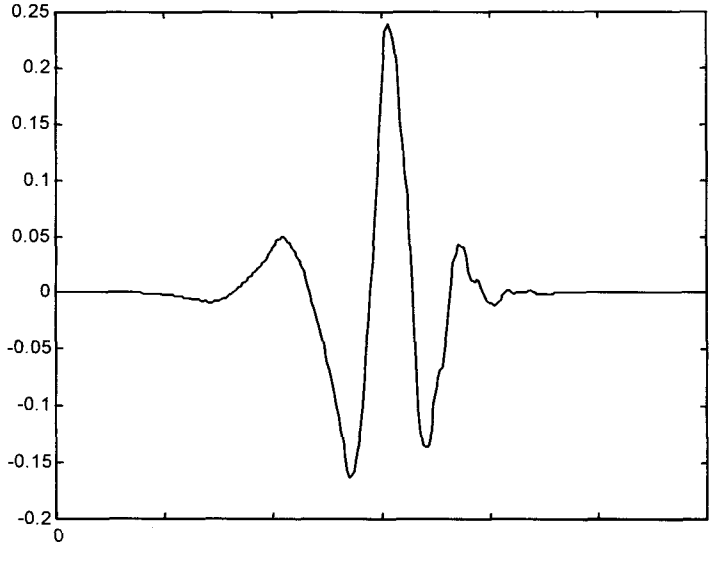


Figure 1.2: Example of a wavelet.

Continuous Fourier transform

A second important category of Fourier transform is the continuous Fourier transform. The Fourier transform of an absolutely integrable function f(t) is defined by

$$F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot \exp(-i \cdot \omega \cdot t) \cdot dt \qquad \textbf{(1.3)}$$

The inverse Fourier transform is given by

$$f(t) = \frac{1}{2\pi} \cdot \int_{-\infty}^{\infty} F(\omega) \cdot \exp(i \cdot \omega \cdot t) \cdot dt \qquad \textbf{(1.4)}$$

The power spectrum of the function f(t) is $|F(\omega)|^2$. The power spectrum is a measure of the energy content of the signal at the different frequencies ω. The continuous Fourier transform can also be generalized to the continuous wavelet transform.

Short-time Fourier transform versus wavelet transform

Practical applications of the Fourier transform request some adaptation of the method. The main problem encountered is that a signal is seldom stationary, so that the period of the signal tends to infinity. In order to satisfy the Fourier condition for convergence, the signal must be integrated from 0 to infinity. Several approaches permit to analyze the signal more locally in order to extract information on the local energy content of the signal or to decompose the signal in such a way that a good signal reconstruction is possible locally with a limited number of coefficients. The classic approach is the short-time Fourier transform, also called the Gabor transform (Gabor, 1946).

A short-time Fourier transform is obtained by first multiplying the signal by a window function $G(t-\omega)$ and then by performing the Fourier transform of the obtained signal.

$$SF(\omega, t) = \int_{-\infty}^{\infty} G^*(t-\tau) \cdot f(t) \cdot \exp(-i \cdot \omega \cdot t) \cdot d\tau \qquad \textbf{(1.5)}$$

The window used by Gabor is the Gaussian window (fig. 1.3):

$$G(t) = a \cdot \exp(-b \cdot t^2) \qquad \textbf{(1.6)}$$

The result of the transform depends on the time t, but also on the frequency ω. The parameter b controls the width or the spread in time. Its Fourier transform is given by

$$G(\omega) = a \cdot (\pi / b)^{0.5} \cdot \exp(-(\omega - \omega_0)^2 / 4b) \qquad \textbf{(1.7)}$$

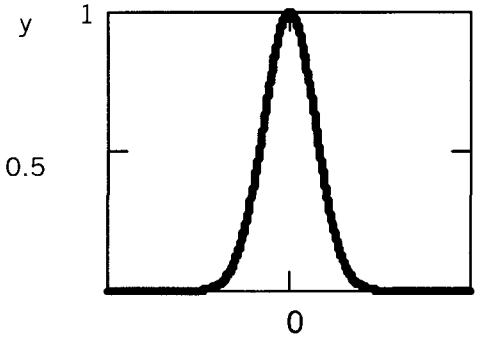


Figure 1.3: Example of a gaussian window (or Gabor window) used in windowed Fourier transform.

The spread of the function G(t) is the same as the one of its translate. Similarly the half-width of $G(\omega)$ is independent of the centered frequency ω_0.

In order to carry out a time-frequency analysis, the signal must be projected on a series of windowed functions. Perfect reconstruction from the projection is possible provided a large number of windows is taken. The projection is then very redundant. Perfect reconstruction requires a number of operations $O(N^2 \log_2 N)$ for a signal of period N (Mallat, 1998).

One of the main ideas of wavelet analysis is already contained in the short-time Fourier transform, namely the decomposition of a signal on dilated and translated versions of a basis function. Figure 1.4 compares the windows of the short-time Fourier transform to the wavelet transform. The main difference between the wavelet transform and the Gabor transform is that the time-frequency window of the Gabor transform is independent of the position and dilation of the window, while for the case of a wavelet transform, the time-frequency window depends on the dilation factor of the wavelet. At low frequency, the time-window is much larger than at higher frequencies. This property of wavelets is in many applications a useful feature. Indeed, it is often desirable to have a result on the high-frequency part of the signal with a good time resolution, while a less good resolution for the low frequencies is not so much of a problem in most applications.

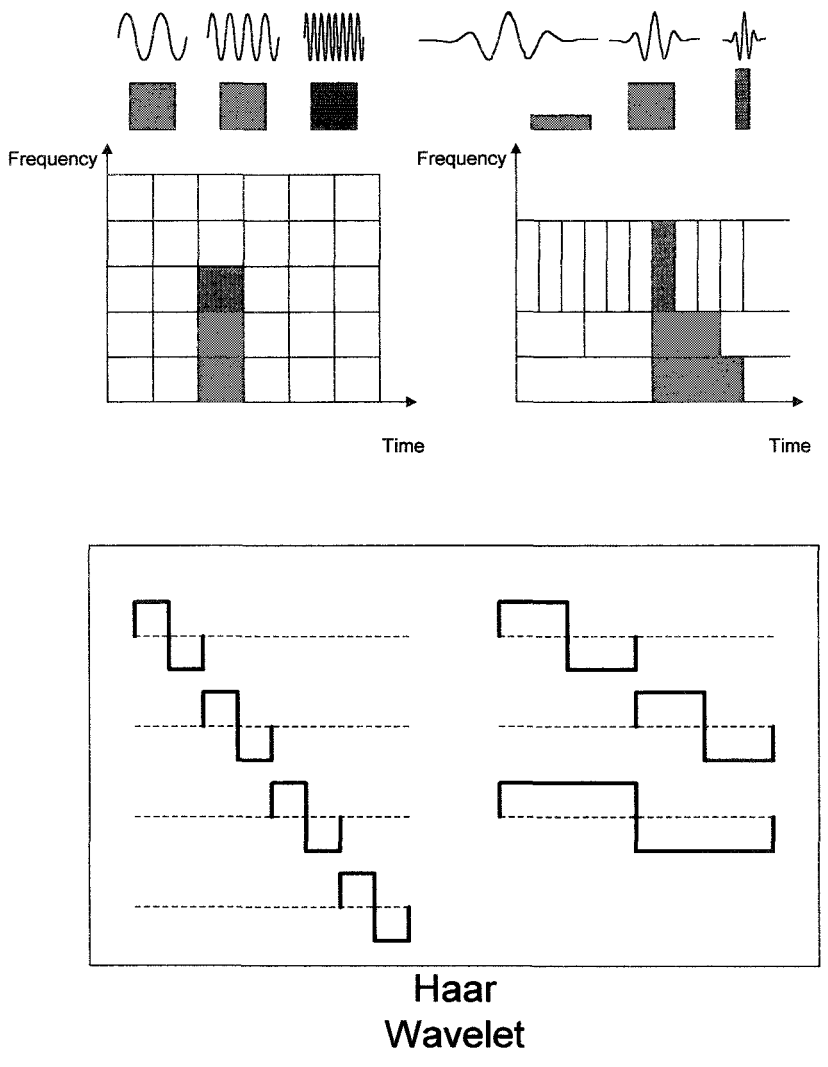


Figure 1.4: Time-frequency tiling of the time frequency domain. Left: Fourier transform, Right: Wavelet transform. Below example of dilated and translated wavelets.

Discrete wavelet decomposition

Let us start with a definition of wavelets englobing both orthogonal and non-orthogonal constructions.

Definition:

A function ψ is called a wavelet if there exists a dual function $\tilde{\psi}$ such that a function $f \in L^2(\Re)$ can be decomposed as

$$f(x) = \sum_{m,n} < f, \tilde{\psi}_{m,n} > \psi_{m,n}(x) \qquad \textbf{(1.8)}$$

The series representation of f is called a wavelet series. The wavelet coefficients $c_{m,n}$ are given by

$$c_{m,n} = < f, \tilde{\psi}_{m,n} > \qquad \textbf{(1.9)}$$

A function $\psi \in L^2(\Re)$ is called an **orthogonal wavelet**, if the family $\{\psi_{m,n}\}$ is an orthonormal basis of $L^2(\Re)$ that is $< \psi_{m_1,n_1}, \psi_{m_2,n_2} > = \int_{-\infty}^{\infty} \psi_{m_1,n_1}(x) \cdot \psi^*_{m_2,n_2}(x) \cdot dx = \delta_{m_1,m_1} \cdot \delta_{n_1,n_2}$ and every $f \in L^2(\Re)$ can be written as

$$f(x) = \sum_{m,n} c_{m,n} \cdot \psi_{m,n}(x) \qquad \textbf{(1.10)}$$

with

$$\psi_{m,n}(x) = 2^{m/2} \cdot \psi(2^m \cdot x - n) \qquad \textbf{(1.11)}$$

$m,n \in Z$.

The wavelet coefficients $c_{m,n}$ of an orthogonal wavelets are given by $c_{m,n} = < f, \psi_{m,n} >$. This follows from the fact that for an orthogonal wavelet the dual function $\tilde{\psi}$ is identical to the wavelet ψ.

The definition of an orthogonal wavelet is quite similar to the definition of a Fourier series. Actually the only difference lies in the definition of the candidates functions for the projection. Projecting the signal on wavelets permits to suppress the condition that the function must be periodic in order to guarantee perfect reconstruction. In a Fourier series, cosine and sine are used as basis functions together with integer dilated of the two basis functions $\cos(2\omega t)$ and $\sin(2\omega t)$. In orthogonal wavelets, dilated and translated of a function are taken: $\psi_{m,n}(x) = 2^{m/2} \cdot \psi(2^m \cdot x - n)$. The dilation factor is also different (m versus $2^{m/2}$).

The simplest example of an orthogonal wavelet is the Haar wavelet (fig. 1.5) defined as

$$\psi_H(x) = \begin{cases} 1 & 0 < y \le 1/2 \\ -1 & 1/2 < x \le 1 \\ 0 & \text{otherwise} \end{cases} \qquad \textbf{(1.12)}$$

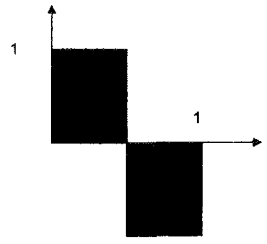


Figure 1.5: Example of an orthogonal wavelet, the Haar wavelet.

The orthogonality of the Haar wavelet can be easily verified, as schematically explained on fig. 1.6.

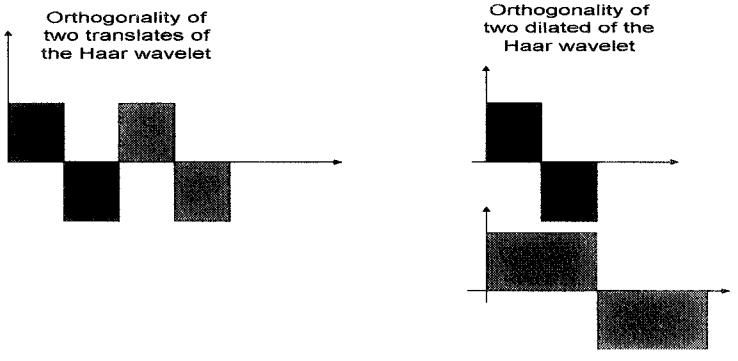


Figure 1.6: The Haar wavelets form an orthonormal basis.

Continuous wavelet transform

As for the Fourier analysis, a continuous wavelet transform can be defined. The integral wavelet transform is defined by

$$W_\psi f(a,b) = 1/\sqrt{a} \int_{-\infty}^{\infty} f(x) \cdot \psi^*\left(\frac{x-b}{a}\right) dx \qquad \textbf{(1.13)}$$

Contrarily to the wavelet series, the factor a is continuous. The interpretation of the wavelet transform is generally difficult and not too intuitive. Figure 1.7 shows an example of a continuous wavelet transform. The value of the wavelet transform is coded as a grey level.

Recent applications of the continuous wavelet analysis are found in a number of different domains. Continuous wavelet analysis has been implemented in vibration monitoring (Staszewski, 1997), heart rhythm perturbations monitoring (Thonet, 1998), in power system waveform analysis (Pham, 1999), ship detection

(Magli, 1999). Important applications of the continuous wavelet analysis in relation to atmospheric and oceanographic modeling have been made (Torrance, 1998). For instance, studies of the El Nino are of vital importance. El Nino corresponds to a phase in which, due to the periodic variation of the water currents, warm water poor in nutrients surges along the south American coast. The analysis of sea surface temperature with the continuous wavelet analysis has shown that the El Nino-southern oscillation index was significantly higher during some periods (1880-1920 and 1960-90). Considering the devastating effects of El Nino, such studies are of vital importance to the future of the many south American countries. An increase of El Nino due to pollutants might ruin the economy of these countries by causing enormous problems to fishery and agriculture.

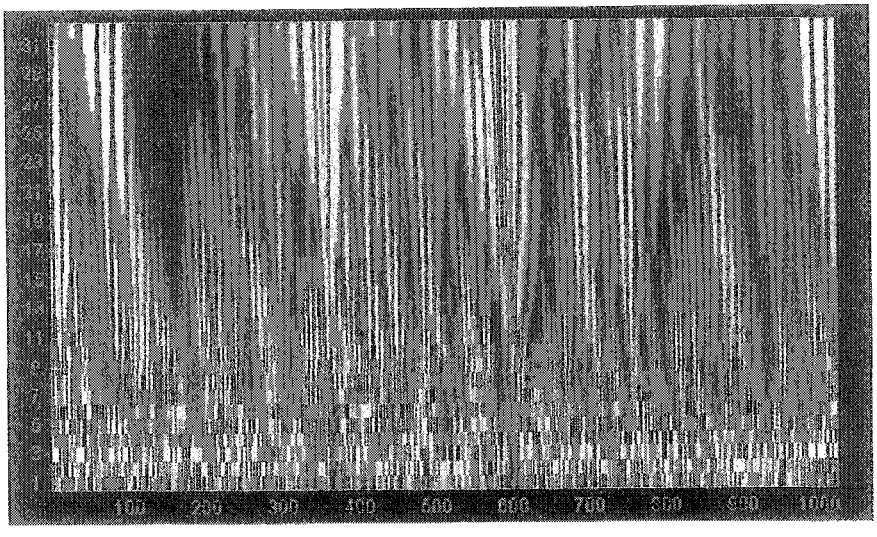


Figure 1.7: Continuous wavelet transform of the logistic map:
$x(n+1) = 4 \cdot x(n) \cdot (1 - x(n))$.

The fast wavelet transform

The Fast Fourier Transform is probably one of the algorithms that has had the most influence on science and engineering. The main idea of the FFT can already found in a paper by Gauss. The idea was rediscovered by James Cooley and John Tukey in 1965. The FFT reduces from N^2 to $N \log_2 N$ the number of necessary operations for a Fourier transform of a signal with N values. We will show that similarly to the Fourier transform, there exists a fast wavelet transform.

The fast wavelet transform is from the practical point of view the most important algorithm in multiresolution analysis. Contrarily to the fast Fourier transform that can be applied in most cases without a deep knowledge of the

algorithm, it is recommendable to understand the fast wavelet algorithm before using wavelets in an application.

The fast wavelet transform, permits the computation of the wavelet transform. At each level of the transform, the data are processed through a low-pass and a high-pass filter. The high-pass filtered data are known as the detail wavelet coefficients. The result of the low-pass transform is used as input data to compute the next level of detail wavelet coefficients.

In order to explain the fast wavelet transform algorithm, we will first introduce a few new concepts that represent the foundations of multiresolution analysis.

The dilation equations (or two-scales relations)

We will not discuss here in much details how to construct wavelets. We will restrict our discussion to giving the main ideas behind the construction scheme. From the practical point of view, it is generally sufficient to know the main properties of the different wavelet families, in order to choose appropriately the best wavelet family for an application. As we will see in the next chapter, a wavelet analysis reduces to a cascade of filters. It is therefore important to understand the properties of the filters. The form of the filter coefficients is essentially determined by the properties of the wavelet family associated to the filter.

One of the most important concept of multiresolution analysis lies in the definition of nested spaces. Nested spaces are like russian dolls, they fit nicely into eachother and the smaller doll is contained in the larger dolls. Figure 1.8 shows an example of a nested space, together with a representation of the complementary spaces W_0 and W_{-1}.

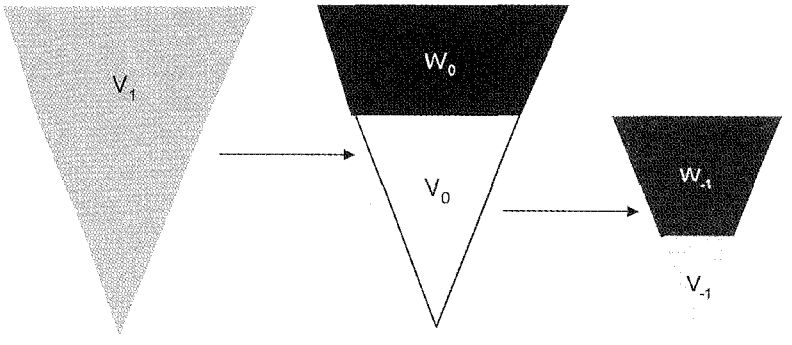


Figure 1.8: Example of nested spaces: $V_{-1} \subset V_0 \subset V_1$. The space W_{-1} is the complementary space of V_{-1} with $W_{-1} \oplus V_{-1}$. Similarly $V_0 = W_0 \oplus V_0$.

The concept of nested spaces (Daubechies, 1992) can be applied to spaces generated by linear combinations of a function, say ϕ. We define V_1 as the space generated by $\phi(2x)$ and its integer translates. The space V_1 corresponds to all possible combinations of ϕ and its integer translates: V_1: $\{\phi(2x-n)\}$. Let us consider now a second space V_0, generated by the 2x dilated function $\phi(x)$ and its translates: V_0: $\{\phi(x-n)\}$. The space V_0 is nested in V_1 if $V_0 \subset V_1$. Generally speaking, it follows from $V_0 \subset V_1$ that any function in V_0 can be written as a linear combination of the functions generating V_1.

$$\phi(x) = \sum_n g_n \cdot \phi(2x-n) \qquad \textbf{(1.14)}$$

Since $V_0 \subset V_1$ the space V_0 can be written as $V_1 = V_0 \oplus W_0$. The space W_0 is the complement of the space V_0. Following the same line of thought as previously, we have $W_0 \subset V_1$ which follows that any function ψ in W_0 can be written as a linear combination of the basis functions in V_1.

$$\psi(x) = \sum_n h_n \cdot \phi(2x-n) \qquad \textbf{(1.15)}$$

The two equations are the so-called dilation equations or two-scales relations. These two equations are central to multiresolution analysis. They permit the reconstruction of a signal starting from the wavelet coefficients (or detail coefficients) and the lowest level of approximation coefficients. Also most constructions of new types of wavelets start from the dilation equations (we come back to this point as we will sketch how to build wavelets).

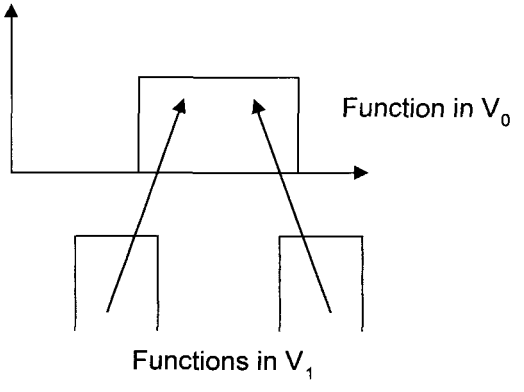


Figure 1.9: Example illustrating the dilation equation for the characteristic function.

As an example, let us take as function $\phi(2x-n)$, the characteristic function, then V_0 is the space of piecewise constant function consisting of zero order

polynomials defined on [n,n+1) with n an integer. In the example of the characteristic function (fig. 1.9), the 2x dilated characteristic function generates a space V_0 that is nested in V_1: $V_0 \subset V_1$. Any function in V_0 can be expressed as a linear combination of the generating functions in V_1. Figure 1.10 shows another example using a second order spline function.

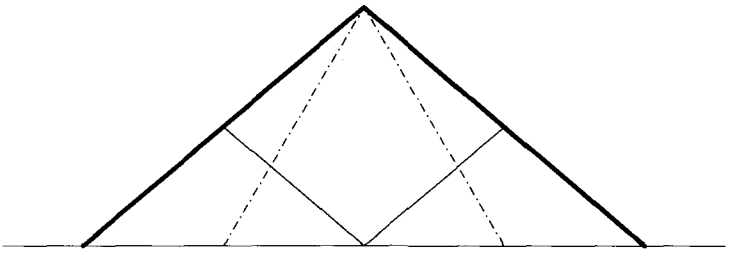


Figure 1.10: Illustration of the two-scales relation for the second order B-spline. The triangular spline function can be decomposed into the sum of translated triangular functions at the higher level of resolution.

Decomposition and reconstruction algorithms

Since $V_1 = V_0 \oplus W_0$, a function in V_1 can be written as the sum of two functions with the first function in V_0 and the second function in W_0. It follows that a basis function in V_1 can be expressed as the weighted sum of the basis functions of V_0 and W_0 (for an exact derivation of the relation below, see Chui (1992)).

$$\phi(2x-k) = \sum_k p_{k-2n} \cdot \phi(x-n) + q_{k-2n} \cdot \psi(x-n) \qquad \textbf{(1.16)}$$

$k \in Z$

This relation is called the decomposition relation. The function ϕ is called the scaling function, while the function ψ is the mother wavelet.

The decomposition algorithm of a function $f \in V_1$ can be computed from the decomposition relation (fig. 1.11). One obtains

$$c_{m-1,n} = \sum_k p_{k-2n} \cdot c_{m,k} \qquad \textbf{(1.17a)}$$

$$d_{m-1,n} = \sum_k q_{k-2n} \cdot c_{m,k} \qquad \textbf{(1.17b)}$$

The proof is according the following line. Take $f_m(x) = \sum_n c_{m,n} \cdot \phi_{m,n}$ with $f_m \in V_m$, then use the decomposition relation to obtain an expression for $f_{m-1}(x)$:

$$f_{m-1}(x) = \sum_k \sum_n c_{m,k} \cdot p_{k-2n} \cdot \phi_{m-1,k} + d_{m,k} \cdot q_{k-2n} \cdot \psi_{m-1,k} \qquad \textbf{(1.18)}$$

The decomposition algorithm can be used iteratively in a cascade of filters, so that a function f may be decomposed into the sum:

$$f = g_0 + g_{-1} + g_{-2} + ... + g_{-N} + f_{-N} \qquad \textbf{(1.19)}$$

with $g_{-j} \in W_{-j}$.

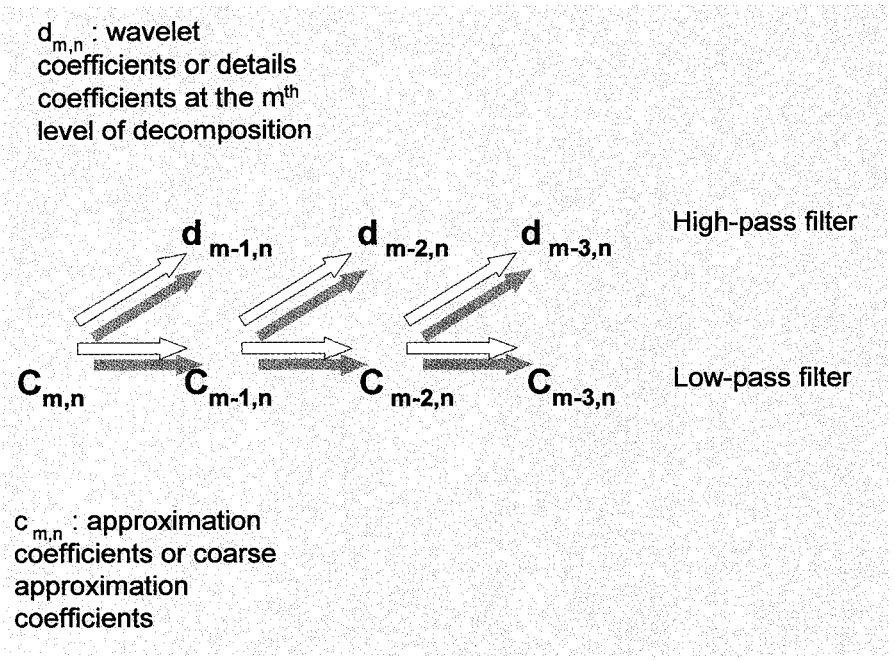


Figure 1.11: Decomposition algorithm.

The reconstruction algorithm is given by the following algorithm:

$$c_{m,n} = \sum_k g_{n-2k} \cdot c_{m-1,k} + h_{n-2k} \cdot d_{m-1,k} \qquad \textbf{(1.20)}$$

The coefficients g and h are defined by the two-scales relation. The proof is very similar to the decomposition algorithm and we will skip it.

The fast wavelet decomposition corresponds to a cascade of filters. The signal is iteratively filtered with a low-pass and a high pass filter. The detail coefficients correspond to the high-passed signal coefficients, while the approximation coefficients result from the low-pass filtering. The low-pass coefficients are then decimated by a factor two and used as input signal at the next level of resolution. After the decimation, the same two filters are applied to the data. The algorithm is invertible and the signal can be reconstructed iteratively from the detail coefficients together with the last level coefficients of the low-pass filter as shown in fig. 1.12.

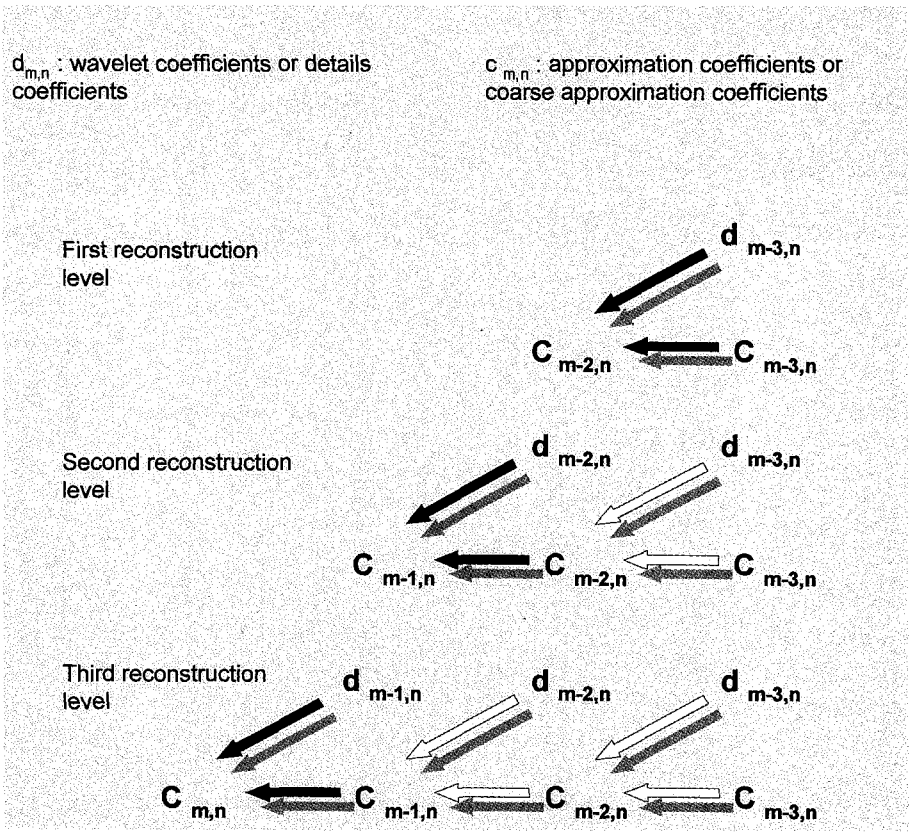


Figure 1.12: Reconstruction algorithm.

The filter coefficients corresponding to an orthogonal wavelet family can be generated from a single filter defined by its Fourier transform. Both the filter coefficients for the decomposition and the reconstruction algorithms are the same, what makes the filter very simple and efficient. An orthogonal decomposition is often optimal to compress information, since there is no redundancy in an orthogonal decomposition. Figure 1.13 shows an example of a wavelet decomposition using orthogonal wavelets.

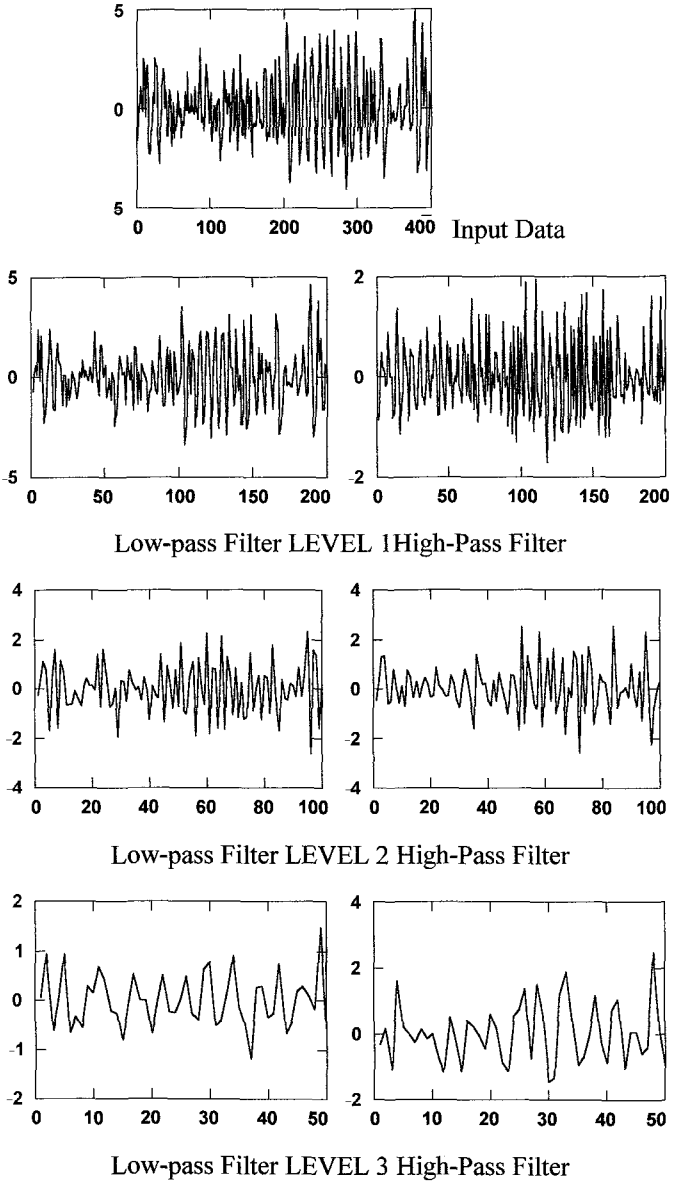
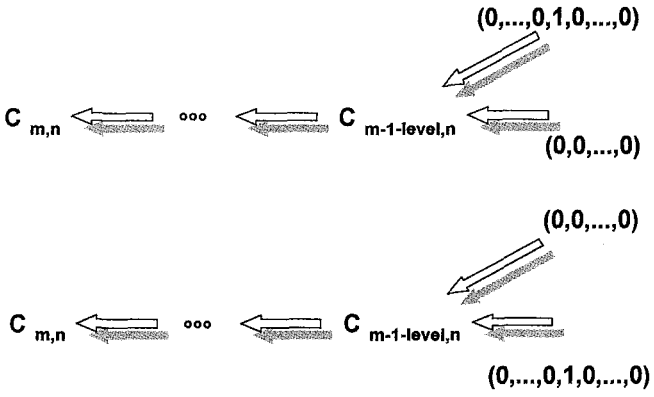
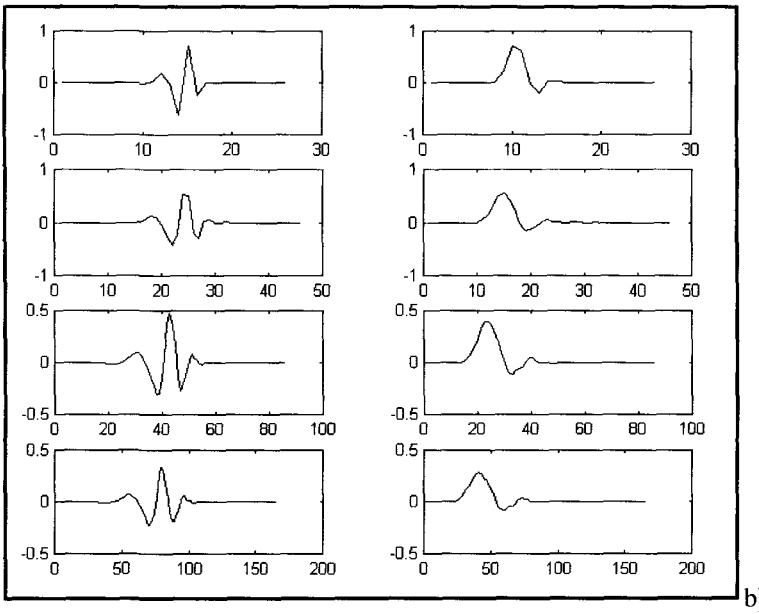


Figure 1.13: Example of a wavelet decomposition with a Haar wavelet.

A wavelet can be constructed from the filter coefficients by using the reconstruction algorithm. At each level, the wavelet approximation is refined as shown in fig. 1.14.



a)



b)

Figure 1.14: The scaling function and wavelet can be computed with the reconstruction algorithm setting all zeros but a one as input. a) Top: wavelet; Bottom: scaling function; b) Left: Approximation of the Daubechies-4 wavelet at the 4 first levels; right: Approximation of the Daubechies-4 scaling function at the 4 first levels.

Definition of a multiresolution

In the previous section, some of the main concepts behind multiresolution were presented somewhat informally. As a complement, we will give here an exact definition of a multiresolution following Daubechies' book, *Ten Lectures on Wavelets*. This definition is broadly accepted and forms the foundation of wavelet

theory. A multiresolution analysis consists of a sequence of embedded closed subspaces $...V_2 \subset V_1 \subset V_0 \subset V_{-1}...$ with the following properties:

-Upward completeness

$\bigcup_{m\in Z} V_m = L_2(R)$ (the ensemble of square-integrable functions on R in a Hilbert space)

Downward Completeness

$\bigcap_{m\in Z} V_m = \{0\}$

Scale Invariance

$f(x) \in V_m \Leftrightarrow f(2^m x) \in V_0$

Shift invariance

$f(x) \in V_0 \Leftrightarrow f(x-n,) \in V_0$ for all $n \in Z$

Existence of a Basis

There exists $\varphi \in V_0$ such that

$\{\varphi(x-n) \mid n \in Z\}$

is an orthonormal basis for V_0.

The above definition is axiomatically fundamental, as it permits to define and verify in practice if a basis forms a multiresolution.

Biorthogonal wavelets

A second central definition in wavelet theory is the definition of a Riesz basis (Mallat, 1998).

A family $\{\varphi_n\}$ of a Hilbert space H is a Riesz basis if for any $y \in H$, there exists A>0, B>0 such that

$$A\|y\|^2 \le \sum_n |< y, \varphi_n >|^2 \le B\|y\|^2 \qquad \textbf{(1.21)}$$

and $\{\varphi_n\}$ are linearly independent.

A Riesz basis can be regarded as a basis in which the orthogonality conditions are relaxed. The usefulness of the concept of Riesz basis will become clear soon. An important theorem (Mallat, 1998) states that if $\{\varphi_n\}$ is a Riesz basis, then there exists a dual basis $\{\tilde{\varphi}_n\}$ such that a function y in H can be decomposed as

$$y = \sum_{n\in Z} < y, \tilde{\varphi}_n > \cdot \varphi_n = \sum_{n\in Z} < y, \varphi_n > \cdot \tilde{\varphi}_n \qquad \textbf{(1.22)}$$

From this expression, it can be deduced easily that the bases $\{\tilde{\varphi}_n\}$ and $\{\varphi_n\}$ fulfill the following biorthogonality condition. Biorthogonality is obtained from (1.22). Setting $y = \varphi_p$, one gets

$$\varphi_p = \sum_{n \in Z} < \varphi_p, \tilde{\varphi}_n > \cdot \varphi_n \tag{1.23}$$

Since the basis is formed of linearly independent functions, the equation follows

$$< \varphi_p, \tilde{\varphi}_n >= \delta(p-n) \tag{1.24}$$

This relation is called the biorthogonality condition. For an orthogonal basis function, $\varphi_p = \tilde{\varphi}_p$, and the expression reduces to the orthogonality condition. A second important theorem states that if a sequence of subspaces satisfies the definition of a multiresolution then there exists an orthogonal basis $\{\psi_{m,n}\}$ for the orthogonal complement of V_m in V_{m-1} with

$$\psi_{m,n} = 2^{m/2} \cdot \psi(2^m x - n) \tag{1.25}$$

In other words, the space spanned by V_m and its orthogonal complement W_m is the space V_{m-1}: $V_m \oplus W_m = V_{m-1}$

In the orthogonal case, the function ψ can be constructed by writing

$$\phi(x) = 2^{-1/2} \cdot \sum_{n \in Z} g[n] \cdot \phi(2x - n) \tag{1.26}$$

$$\psi(x) = 2^{-1/2} \cdot \sum_{n \in Z} h[n] \cdot \phi(2x - n) \tag{1.27}$$

Taking the Fourier transform, one obtains after some manipulation an expression relating the Fourier transform of g to the Fourier transform of h:

$$H(e^{j\omega}) = -e^{-j\omega} \cdot G^*(e^{j(\omega+\pi)}) \tag{1.28}$$

or in the time domain:

$$h[n] = (-1)^n \cdot g[-n+1] \tag{1.29}$$

Inserting (1.29) in (1.25), one obtains

$$\psi(x) = 2^{-1/2} \cdot \sum_{n \in Z} (-1)^n \cdot g[-n+1] \cdot \phi(2x - n) \tag{1.30}$$

This shows that a wavelet $\psi(x)$ can be expressed as a weighted sum of scaling functions $\phi(2x-n)$.

Wavelets and subband coding

Wavelet theory has been created first by researchers with a background in physics and mathematics. Subband coding has been developed mostly by the electrical engineering community (Croisier, 1976). At one point, both communities realized the parenty between the two subjects (Vetterli, 1984, 1992).

Independently from wavelet theory, filters were found that allow a signal to be split into downsampled subband signals and then reconstructed perfectly. These filters are called perfect reconstruction filters and can be shown to be equivalent to the filters used in the fast wavelet algorithm. Figure 1.15 shows an example of perfect reconstruction filters associated to Daubechies wavelet. From the practical point of view, the understanding that subband coding was equivalent to wavelet theory has had great practical implications, as it offered the signal processing community a bridge to filter theory.

We will present succinctly subband coding, emphasizing the similarities to the approach presented in the previous sections. We will introduce a number of new aspects, that are better explained within the framework of subband coding. In particular, the conditions on the filters to ensure perfect reconstruction are given.

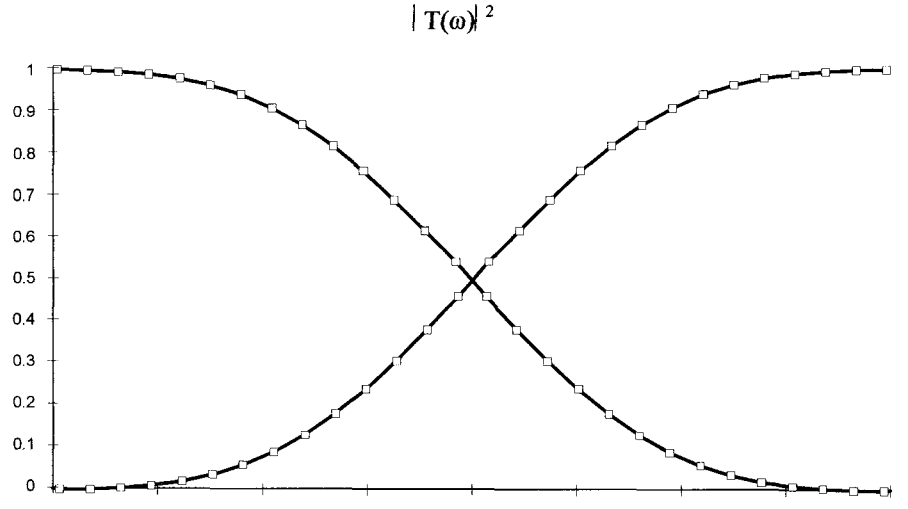


Figure 1.15: Filter characteristics of two filters satisfying the power complementarity condition (Daubechies-4.).

The close connection between wavelet theory and filter theory can be understood by looking first at series expansions of discrete-time series. A discrete signal x[n] can be expanded orthogonaly or biorthogonaly. Orthogonal expansion conserves the energy. This property is very useful for spectral analysis, as the energy in the different subbands sums up to the total signal energy. Biorthogonal

expansions extend series expansions to a large number of filters. In many problems, orthogonal expansions are not suited and biorthogonal expansions are necessary, as for instance in wavelet-based fuzzy methods. Consider a signal x[n]. An orthogonal expansion of x[n] on an orthogonal basis $\{\varphi_k\}$ is

$$x[n] = \sum_{k \in Z} X[k] \cdot \varphi_k[n] \tag{1.31}$$

with

$$X[n] = \sum_{k \in Z} \varphi_k^*[n] \cdot x[n] = < \varphi_k^*, x[n] > \tag{1.32}$$

Energy conservation can be expressed under the form:

$$\|x\|^2 = \|X\|^2 \tag{1.33}$$

For dual bases $\{\varphi_k\}, \{\tilde{\varphi}_k\}$ satisfying the biorthogonality condition

$$< \varphi_k[n], \tilde{\varphi}_l[n] >= \delta[k-l] \tag{1.34}$$

the biorthogonal expansion is given by one of the two expansions:

$$x[n] = \sum_{k \in Z} X[k] \cdot \varphi_k[n] \tag{1.35a}$$

$$x[n] = \sum_{k \in Z} \tilde{X}[k] \cdot \tilde{\varphi}_k[n] \tag{1.36a}$$

with

$$X[n] = \sum_{k \in Z} \varphi_k^*[n] \cdot x[n] = < \varphi_k^*, x[n] > \tag{1.35b}$$

$$\tilde{X}[n] = \sum_{k \in Z} \tilde{\varphi}_k^*[n] \cdot x[n] = < \tilde{\varphi}_k^*, x[n] > \tag{1.36b}$$

Similarly to the continuous series expansion, the expansion is stable only if a condition similar to the one in the definition of a Riesz basis is fulfilled: there exists A>0 and B>0 such that

$$A \cdot \sum_n |X[k]|^2 \leq \|x\|^2 \leq B \cdot \sum_n |X[k]|^2 \tag{1.37}$$

In the biorthogonal case, energy conservation (1.33) is replaced by a different energy conservation relation:

$$\|x\|^2 = < X[k], \tilde{X}[k] > \tag{1.38}$$

The recognition that the wavelet series formulation and the theory of perfect reconstruction filter banks are deeply related is a major achievement in signal processing. The link between filter banks and wavelet theory has lead to the development of fast algorithms to implement practically and efficiently the wavelet ideas in practical applications. Following the line of presentation by Vetterli (1995), let us discuss this.

Consider the four filters P,Q,G,H with the impulse responses p, q, g, h satisfying the relations

$$\varphi_{2k}[n] = g[2k-n] \tag{1.39a}$$

$$\varphi_{2k+1}[n] = h[2k-n] \tag{1.39b}$$

$$\tilde{\varphi}_{2k}[n] = p[n-2k] \tag{1.39c}$$

$$\tilde{\varphi}_{2k+1}[n] = q[n-2k] \tag{1.39d}$$

The filters P,Q correspond to the decomposition filter coefficients (1.17-18), while the filters G,H are the reconstruction filters. After some manipulations, it can be shown that perfect reconstruction is achieved if the following relation is fulfilled:

$$\sum_{k\in Z} g[k]\cdot p[2n-k] = \delta[n] \tag{1.40}$$

$$\sum_{k\in Z} h[k]\cdot q[2n-k] = \delta[n] \tag{1.41}$$

In words, perfect recognition is achieved if the biorthogonality relations (1.34) are fulfilled.

For an orthogonal basis, all the filter coefficients can be derived from just one filter. The impulse responses q, g, h are given by

$$h[n] = (-1)^n \cdot g[2K-1-n] \tag{1.42a}$$

$$p[n] = g[-n] \tag{1.42b}$$

$$q[n] = h[-n] \tag{1.42c}$$

with K the filter length.

There are several methods to construct wavelets, either using the Fourier approach, the lifting scheme or the z-transform. The z-transform is defined as $F(z) = \sum_{n=0}^{\infty} f[n]\cdot z^{-n}$. Using the z-transform the perfect reconstruction condition can be expressed as:

$$P(z)\cdot G(z) + Q(z)\cdot H(z) = 2 \tag{1.43a}$$

$$P(z)\cdot G(-z)+Q(z)\cdot H(-z)=0 \qquad \textbf{(1.43b)}$$

A number of wavelet constructions start from the above expression (Mallat, 1998).

The two-channels filter bank is given by the diagram in fig. 1.16, in which the symbol with the arrow pointing towards the bottom represents downsampling, while the other symbol with the reverse arrow is for upsampling.

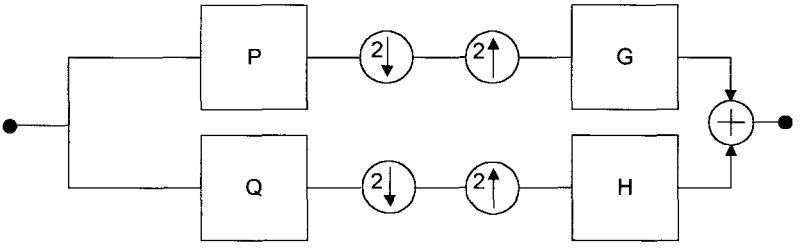


Figure 1.16: Two channels filter bank with analysis filters P,Q and synthesis filters G,H.

Applications

It is not possible anymore to offer a complete review of wavelet applications. Several reviews have been written to cover some particular fields of applications. Biomedical applications have been reviewed by Unser (1996), Kumar et al. (1997) survey geophysical applications. Image compression and pattern recognition are discussed by Vetterli (1999), Tang (2000) and Szu (1996a). Applications to chemical analysis are reviewed by Leung (1998).

In this first section dedicated to applications, only a few significant applications will be discussed to furnish a rapid overview of the main domains of applications. Numerous other examples will be discussed in other chapters.

Data Analysis

There is a large number of applications, such as in astronomy, medical imaging or satellite imaging, for which the simple analysis of images at a well-chosen resolution permits to discover features in an image that would have otherwise stayed not clearly visible (Li, 1995a, 1995b). The good edge detecting properties of wavelets have been also used in data fusion (Fonseca, 1996) for images obtained from different satellites. The images are fused by constructing a wavelet pyramid using the more dominant high frequency wavelet coefficients among the two images.

Data compression

Multiresolution and subband coding have been used to compress video signals, still images or sound. The JPEG format is for instance a compression method using discrete cosine transform. The discrete cosine transform (DCT) is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot \cos(2\pi(2N+1) \cdot K / 4n) \qquad \textbf{(1.44)}$$

In many applications, the blocky appearance of JPEG compressed images is not acceptable. Several wavelet-based compression codes have been developed (JPEG). The JPEG 2000 will most likely replace the JPEG standard. The JPEG 2000 standard uses wavelet for compression. The standard supports several decomposition schemes and most wavelets. Figure 1.17 shows an example of the superior quality of image compression, by comparing to images reconstructed at 0.125 bits per pixel.

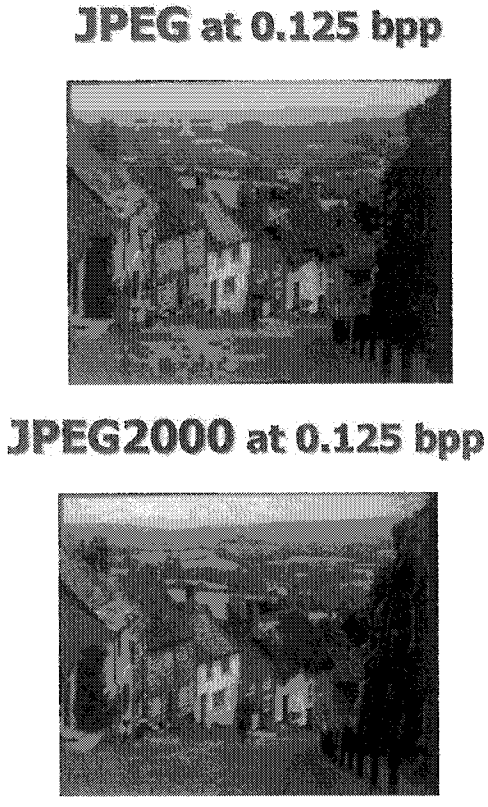


Figure 1.17: Example comparing the quality of an image with JPEG and JPEG 2000. By courtesy from C.Christopoulos (Christopoulos, 1999).

One of the first real world application of wavelets in data compression is the FBI fingerprint digitalization standard (Brislawn, 1995). Fingerprint images are digitized at a resolution of 500 pixels per inch with 256 levels of gray. A single fingerprint needs about 0.6 Mbytes to store. The FBI has collected about 30 millions fingerprints from the beginning of the century. Obviously data compression is necessary. Compared to JPEG, the quality of the wavelet-based compressed image at a 15:1 compression factor is better. Roughly speaking wavelet-based methods are superior to JPEG to compress images, for when most of the information is contained in the image contours. The FBI compression standard combines two methods: wavelet packets and best basis (see part 2). The idea consists of determining at each level of decomposition the best basis. The best basis is defined as the basis chosen in a wavelet dictionary that minimizes the entropy (Coifman, 1992). In other words, the basis is kept for which most of the signal is contained in a small number of coefficients. The entropy is defined as

$$S = \sum_{n,m} d_{m,n} \cdot \log_2 d_{m,n} \qquad \textbf{(1.45)}$$

with $d_{m,n}$ the wavelets coefficients at level m.

Denoising

Certainly one of the most interesting applications of multiresolution analysis is in the domain of denoising (Donoho, 1994). The main idea of the so-called thresholding methods is quite simple. Remove all the coefficients below a given threshold. This approach consists of approximating the signal with only the largest coefficients. The problem is then to determine a threshold that is not too high, in order to keep the essential signal features and also not too low to reach efficient denoising. For a signal corrupted with white noise, the wavelet denoising approach can be justified. The fast wavelet transform consists of a number of linear operations on the data. The wavelet transform of a white noise results into normally distributed values of the wavelet coefficients. The operation of removing all the small coefficients at all levels of resolution is a way of filtering the signal in the whole frequency range.

One distinguishes between soft and hard thresholding. With hard thresholding, the coefficients are estimated with the expression

$$\hat{d}_{m,n} = \begin{cases} d_{m,n} & \text{if} \quad d_{m,n} > \lambda \\ 0 & \text{otherwise} \end{cases} \qquad \textbf{(1.46)}$$

In the soft thresholding method, each coefficient is reduced by a small value. An example of soft thresholding is given below

$$\hat{d}_{m,n} = \begin{cases} \operatorname{sgn}(d_{m,n}) \cdot (|d_{m,n}| - \lambda) & \text{if} \quad d_{m,n} > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{1.47}$$

Opinions diverge strongly on which method is the best. Donoho and Johnstone (1994) have proposed several methods to choose the value of the threshold. An example of such a threshold is

$$\lambda = \sqrt{2 \cdot \sigma^2 \log n} \tag{1.48}$$

with σ^2 the variance from the original data set containing n values.

Another method to estimate the threshold λ is to minimize Stein risk function:

$$S = n + a(\lambda) \cdot (\lambda^2 - 2) + \sum_{k=a(\lambda)+1}^{n} d_k^2 \tag{1.49}$$

with $a(\lambda)$ the number of coefficients less than equal to the threshold λ and d_k the wavelets coefficients rearrange into an increasing series.

Denoising is a very active and fast developing field. It is difficult at time to make a simple statement on the merits of the different thresholding methods. Thresholding methods rely often on different assumptions on the signal and on the noise distribution. A certain type of thresholding may prove to be almost optimal for a certain class of problems. In real world applications, it is often difficult to compare the different results. The evaluation of the results of denoising, for instance in denoising of images or videos, may depend quite much on the subjective judgement of the observer.

This page is intentionally left blank

PART II PREPROCESSING: THE MULTIRESOLUTION APPROACH

This page is intentionally left blank

2. Preprocessing: The Multiresolution Approach

An impressive number of applications combine wavelet analysis to another standard signal processing method. In many applications, a wavelet decomposition is used for preprocessing. The goal of preprocessing is very often the reduction of a problem‘ s dimensionality or complexity. This chapter takes the stand to identify some of the major issues in signal preprocessing and to explain the contributions of wavelet theory to these issues. The methodological aspects are privileged at the expense of an exhaustive presentation of the multitude of combinations between standard signal processing methods and multiresolution analysis.

It is very difficult to define exactly what is preprocessing. The boundary between preprocessing and processing is often very fuzzy. Tentatively, preprocessing may be defined as the transformation of data into a form suitable for processing with a standard processing method. We focus primary on two related topics that are not only central to signal preprocessing, but are also at the very heart of soft computing:

-The curse of dimensionality:

This expression refers to the fact that the sample size needed to estimate a function grows very often exponentially with the number of variables.

-The complexity issue:

Some problems are intrinsically difficult to solve exactly. The necessary computing time to solve a difficult problem increases often very rapidly with the size of the problem (the size of a problem is often characterized by the number of inputs). Some of the science and art in signal processing consists of choosing the right method to find satisfactory solutions to hard problems with a limited amount of computing time.

Let us recall that soft computing deals with solving computationally intensive problems with a limited amount of computing power and memory by giving up some of the precision. Soft computing covers a range of methods that are somewhat tolerant of imprecision, uncertainty and partial truth. The necessary computing power to solve a problem depends on the difficulty of the problem and on the necessary accuracy of the solution. Also the number of necessary datapoints in learning depends on these factors. In this chapter, we present a number of methods to determine or to decrease the dimensionality of a problem, through projection techniques, pursuits and data transforms.

The double curse: dimensionality and complexity

Recently a financial software company claimed that 90 % of the development and computing time in financial problems is used for preprocessing. It is very difficult to argue against or to confirm such a provocative statement as the concept of preprocessing cannot be defined in a general manner. The boundary between preprocessing and processing is in many problems quite difficult to draw. Preprocessing can be defined as the preparation or the transformation of the data into a form suitable for processing with a standard method. This definition reports the problem onto defining what is a standard method.

Some of the most powerful tools in signal processing perform badly at high dimensions. Therefore a very important part of preprocessing deals with the problem of dimension reduction. The general reasons involved for the failure of many classical signal processing methods at high dimensions are

-The curse of dimensionality

-The increasing complexity of many problems as the dimension increases.

In the first part of this chapter, we will discuss these two problems. We should nevertheless not hide the reality: the true reasons for an unsatisfactory modeling are, more often than one wants to admit, not the signal processing part. In many cases, the failure to describe a problem correctly or to find a solution is related to the difficulty for the human brain to deal with more than 3 dimensions. For that reason, high-dimensional problems are often ill-posed and the following difficulties occur:

-Missing variables

-Inappropriate variables were chosen

Other common problems are

-Missing data

-Wrong data

-Noisy data

Publication on these problems are scarce. Understandably, one prefers to report on a success than on failures. Also it is difficult to learn from failures as the necessary know-how is very often specific to a given and well specified problem. This situation is nevertheless unsatisfactory and may lead to a broadening of the gap between applications and fundamental research.

Fortunately, one observes a new trend in many commercial signal processing tools. Many programs include diagnostics tools, often based on statistical methods, to diagnose automatically outliers, reject insignificant variables or even suggest that some results are most likely not significant due to a lack of data.

In the next chapter, we will discuss preprocessing with a biased mind. We have selected a number of preprocessing methods and show how multiresolution may improve them. Among the multiresolution methods, we focus here essentially on wavelet theory.

Wavelet preprocessing may be used in connection to many problems. Feature extraction, classification, modeling, data compression and denoising are the problems that have benefited most of multiresolution preprocessing. Different

goals may be set to the preprocessing stage. In image and speech processing, data reduction is very often the issue. Wavelet preprocessing permits to reduce the dimensionality of the problem. Wavelets can be also used to filter data, remove noise in data or to carry out a segmentation of the input space. Applications using wavelet preprocessing include fire detectors, filtering of satellite images, detection of emergency states in neurosurgical patients, quality control and inspection, denoising of magnetic resonance images, Chinese character recognition, face classification, image compression, classification of EEG signals, features extractions in seismic waves. Generally wavelet preprocesing results into the selection of a number of significant wavelet coefficients on which a standard processing method is applied (fig. 2.1).

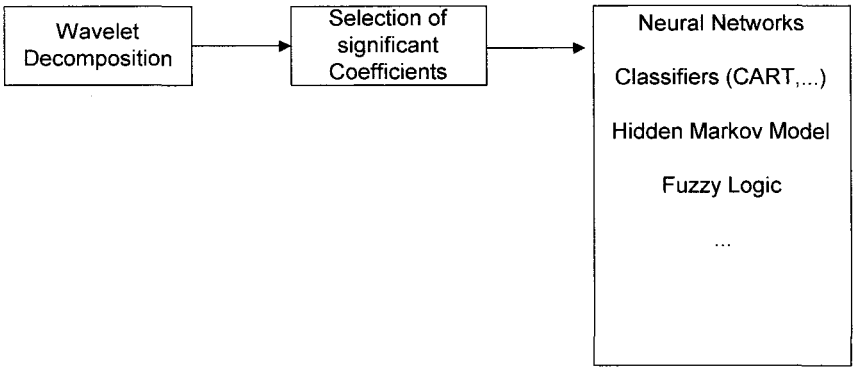


Figure 2.1: Wavelet preprocessing has been used in a number of applications (image processing, pattern recognition, spectral analysis, controllers,...) in connection with standard signal processing methods. The preprocessing stage corresponds generally to selecting a number of significant wavelet coefficients for further processing with a standard method.

Curse of dimensionality

The curse of dimensionality is a term coined by Bellman (1961). It refers to the fact that in many problems, the sample size needed to estimate a function to a given accuracy grows exponentially with the number of variables. It is only in recent years, that this expression has taken a widespread significance. Paradoxically, it is only as an impressive computing power became available in the last years, that the implications of the curse of dimensionality became fully appreciated. This may be explained by the fact that the exponential growth of computing power has encouraged many industries and universities to address problems of increasing complexities in which the curse of dimensionality did strike more often than expected.

The *curse of dimensionality* is about to become a new icon in signal processing, in the same class as the butterfly effect for chaos theory. One of the problems with the curse of dimensionality expression is that depending on the scientific community, the definition of the dimension is different. Take the example of an image. For the physicist, an image is essentially a two-dimensional object. For the mathematician, an image is often considered as a surface embedded into a 3-dimensional space, while in digital signal processing its dimension corresponds to the number of pixels. In Bellman's views, the curse of dimensionality refers to the exponential growth of hypervolume as a function of dimensionality. It corresponds the best to the signal processing vision. Beating the curse of dimensionality consists of finding through diverse preprocessing techniques an acceptable representation of the information with a reduced number of variables.

The expression *curse of dimensionality* covers only one part of the problem described in the introduction. The second part consists of the increasing complexity of a problem with the number of inputs. In order to describe this aspect, another approach based on the classification of problems according to their difficulty is necessary.

Classification of problems' difficulty

The level of difficulty of mathematically posed problems may be measured by the time, number of steps and memory space required to solve them (respectively time complexity, computational and space complexity). Much effort in mathematics have been made to characterize the computational complexity of problems. One often distinguishes between so-called *easy* and *hard* problems. Depending on how the number of steps necessary to solve a problem scales with the size of the number N of inputs, the problem is defined as easy or hard.

An *easy* problem is a problem that is verifiable and solvable in polynomial time. This means that a known algorithm is guaranteed to terminate within a number of steps, which is a polynomial function of the size of the problem.

A very hard problem is a problem, that is neither solvable, nor verifiable in polynomial time. Many problems may be classified into an intermediate class, the so-called NP problems. A problem is NP if no polynomial function of the number of inputs N describes correctly the increase with N of the number of necessary steps to solve the problems with a deterministic Turing machine/algorithm, but a solution is verifiable in polynomial time

Finally a problem is called NP- hard if solving it in polynomial time would make it possible to solve all problems in class NP in polynomial time (Garay, 1979).

An important example of an NP-hard problem.

From the point of view of soft computing, an important NP-hard problem is the determination of an optimal approximation of a signal using a redundant dictionary (see definition below). It follows that

-the search for an optimal decomposition of a signal using a redundant wavelet dictionary is NP-hard problem.

-the search for an optimal fuzzy controller using a redundant dictionary of membership functions is NP-hard

Definition: dictionary/optimal approximation

Let H be a N-dimensional Hilbert. A dictionary D for H is a family of functions g_i of norm 1 in H, such that linear combinations of functions g_i in the dictionary are dense in H. The smallest possible dictionary is called a base of H, while the dictionary is redundant otherwise. We define an optimal approximation $\hat{f}$ of a function f, to be a linear combinations of functions g_i in the dictionary such that $||\hat{f} - f||$ is minimum.

Summarizing the above discussion, we conclude that in many problems there are two curses to face. On the one hand, the curse of dimensionality that leads with increasing dimension to an exponentially growing dataset. On the other hand, many problems, such as decomposing optimally a signal with a redundant dictionary are NP-hard. The number of operations necessary to solve them increases also exponentially with the size of the dictionary.

A possible strategy to fight the *double curse* is to reduce the dimension of the problem and/or the accuracy of the requested solution to the problem. We discuss below these two approaches.

Dimension reduction

The Karhunen-Loève transform is the classical linear method to reduce the dimension of a dataset with a projection technique. The Karhunen-Loève method corresponds to a change of basis. The new basis is formed by a linear transform of the original orthogonal basis. The Karhunen-Loève method is the ideal method to reduce the dimension of a dataset of gaussian random vectors. The dataset approximated on the most significant Karhunen-Loève basis minimizes the error in comparison to any other linear basis transform.

If the number of different bases is large, then the Karhunen-Loève method becomes intractable. This is the case of many on-line problems. For such cases, a principal component analysis may be approximated by using neural networks, such as for instance Oja' s network. The Karhunen-Loève method is often more efficient on preprocessed data. We present an example (the problem of finding the best coordinates to represent data in a fuzzy system) for which wavelet

preprocessing is necessary in order to implement efficiently the Karhunen-Loève transform.

In nonlinear problems, the Karhunen-Loève transform does not generally furnish good results. Nonlinear projections techniques are possible alternatives. Projection pursuit regression and exploratory projection pursuit have been applied with success to a number of problems. In recent years, these methods have been completed by wavelet-based methods. The best basis and the matching pursuit algorithms are two good examples of wavelet-based methods. The *best basis* corresponds to searching within a redundant basis, an orthogonal basis that approximates best the Karhunen-Loève basis by minimizing an entropy function. The matching pursuit searches iteratively for the best matching between a basis contained in a dictionary and some portion of a signal. The algorithm is a greedy algorithm, in the sense that the contribution of the best matching basis is removed from the signal and the algorithm iterated on the residue.

Karhunen-Loève transform (principal components analysis

Principal component analysis is the classical linear method to search for a low dimension space to embed data. A principal component analysis consists of a Karhunen-Loève transform. The Karhunen-Loève transform corresponds to a change of basis. It furnishes an orthogonal basis of vectors that represent optimal direction of projections. The new basis corresponds to the eigenvectors of the covariance operator $R[n,m] = E\{Y[n]\, Y^*[m]\}$.

For random gaussian vectors Y of zero mean, the realizations of Y (sometimes called objects) define a cloud of points in $\Re^N$. The Karhunen-Loève transform furnishes an orthogonal basis of vectors g_n giving the directions of the principal axes of the cloud. The most useful properties as well as the limitations of the Karhunen-Loève may be understood from the theorem below (Vetterli, 1995). The theorem states that provided the data points are randomly distributed, the Karhunen-Loève transform is the ideal tool to determine the optimal low-dimensional representation of data. The error obtained by removing the dimensions corresponding to the lowest eigenvalues equals the sum of the removed eigenvalues. A central condition in the theorem is that Y is a random vector. If this condition is not fulfilled, the process may be highly non-uniform and the Karhunen-Loève may not provide good approximations of the process. In these cases, a nonlinear dimension compression method is generally necessary.

Theorem:

Let $\{g_m\}_{0 \le m < N}$ *be an orthogonal basis such as a gaussian random vector Y of zero mean can be decomposed as* $Y = \sum_{m=0}^{N-1} < Y, g_m > g_m$. *Define* Y_M, *as* $Y_M = \sum_{m=M}^{N-1} < Y, g_m > g_m$ *For all* $M \ge 1$, *the approximation error*

$\varepsilon(M) = E\{ \|Y - Y_M\|^2 \}$ is minimum if and only if $\{g_m\}_{0 \le m < N}$ is a Karhunen-Loève basis ordered by increasing eigenvalues □

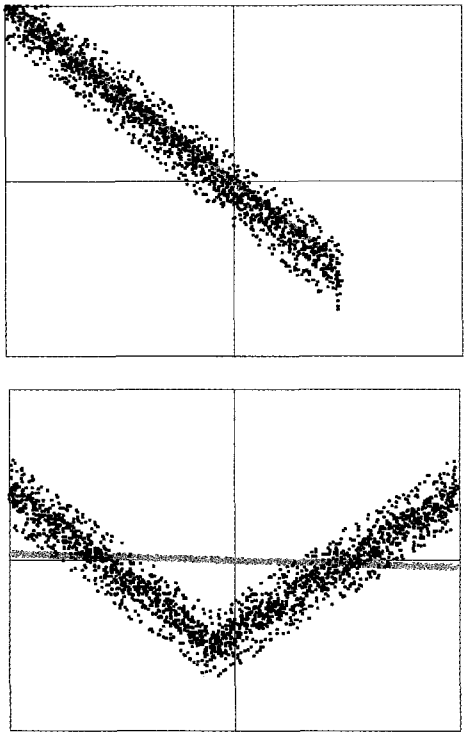
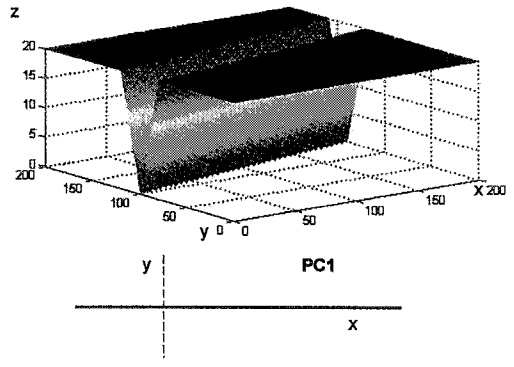


Figure 2.2: Principal component analysis with Oja's networks. a) the principal component gives the main direction of the data. b) example showing the failure of the principal component analysis approach for nonlinear data.

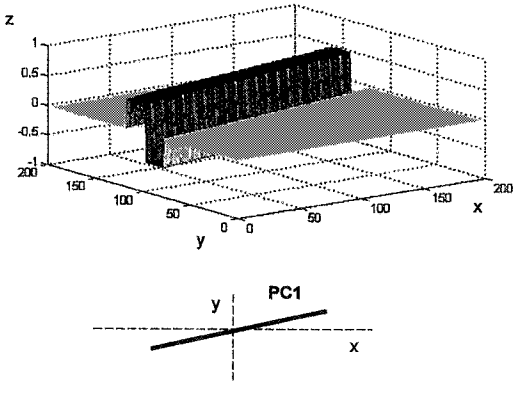
Principal components analysis can be implemented also with neural networks (fig. 2.2). In comparison to the normal computation method, these networks have the advantage that they can be computed on-line without having to store the covariance vector as in the matrix approach. Several types of neural networks have been used to extract the main component of the signal. Let us mention for instance the bottleneck networks (Baldi,1989) and Oja's networks (Oja, 1982). The network searches for projections Ax of the data that maximize the correlation to x.

Search for good data representation with multiresolution principal components analysis

Many problems in soft computing correspond to finding out a good description of a control surface. The description of some knowledge can often be very much simplified by choosing the right cartesian axis. For instance, the complexity of a fuzzy controller may be sometimes considerably reduced if the membership functions are chosen according to a preferred direction of the data. This is achieved by rotating the axis such as the control surface becomes aligned according to the preferred direction.



a)



b)

Figure 2.3: Principal components analysis of the points defined on a square grid: a) Input data; the first principal component is along the x-axis; b) Wavelet coefficients; The first principal component gives correctly the main direction of the triangular bump in fig. 2.3a.

Consider the surface in fig. 2.3a defined on a regular grid. The surface is defined by the value f(i,j) of the function on a regular two-dimensional grid, defined by its coordinates (i,j). The naive processing of the data points with a PCA of the input vectors (i,j,f(i,j)) gives a principal axis that is not well correlated to the direction of the triangular bump. This can be corrected by preprocessing the data first with a wavelet transform to obtain the wavelet coefficients d. In a second stage, the vectors (i',j',d(i',j')) with an absolute value above a given threshold are processed with PCA. Figure 2.3b shows a one level wavelet decomposition of the surface in fig. 2.3a with a one-dimensional Haar wavelet. The correct main direction is given by the PCA of the wavelet coefficients.

The method can be further improved by weighing the different data points with their corresponding wavelet coefficients.

$$Y' = \begin{pmatrix} d_{1,1} \cdot i_1 & d_{1,1} \cdot j_1 \\ \dots & \dots \\ d_{max,max} \cdot i_{max} & d_{max,max} \cdot j_{max} \end{pmatrix} \qquad (2.1)$$

Figure 2.4 shows that the weighing of the different points with wavelet coefficients has the effect to concentrate the points with low wavelet coefficients towards the origin and to give more weight to the points containing much energy (high values of d).

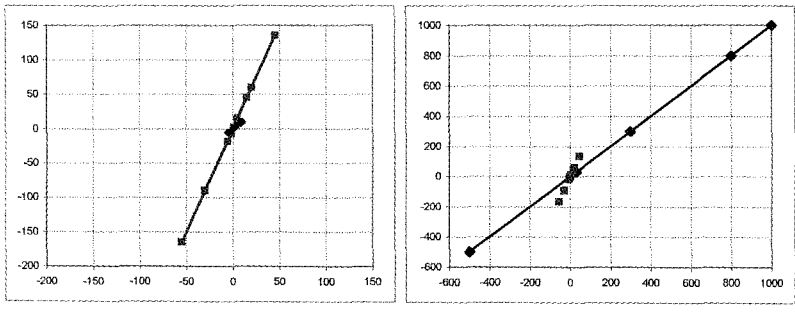


Figure 2.4: Illustration of the principal component analysis method using the transform defined in.(2.1). The projection on the plane of the original points are shown in fig. 2.4a. The main component is given by the line assuming d=1 for all points. b) The wavelets coefficients of the points marked as diamonds are assumed to have 100x larger coefficients (d=100) than the points marked as squares. The main line gives the direction of the principal component, which is almost aligned with the direction given by the diamonds.

The above method can be also applied to different problems. Recently, a number of applications have demonstrated the potential of multiresolution PCA methods. In those applications, features are extracted by combining the ability of

PCA to decorrelate the different variables with the tendency of orthogonal wavelets to decorrelate signals. After the PCA of the wavelet coefficients is carried out, only a number of relevant features at different resolution levels are kept for further analysis.

Feng et al. (2000) show that face recognition based on PCA is improved if the principal component analysis is carried out on a midrange frequency subband. Okimoto and Lemonds (1999) claim that principal component analysis in the wavelet domain provides powerful features for underwater object recognition. Bakshi (1999) reviews multiresolution principal analysis in process monitoring. In the same spirit, Szu et al. (1998a) have combined wavelet preprocessing to independent component analysis (Szu, 1998a).

A different approach was used for fusion of satellite images (Jun Li, 1999). In that approach, images are first analyzed with a PCA. The k-principal components are subsequently decomposed with wavelets. Finally, low-resolution images are enhanced by adding the wavelet coefficients of the k-principal components of high-resolution images.

Projection pursuit regression

Parameter estimation becomes often unpractical in a high-dimensional space due to the sparseness of the data. The dimension of the problem can be reduced by projecting the data in several low-dimensional spaces. The purpose of projection pursuit regression (Friedman, 1981) is to find simple approximations of a function f(**x**) from n observations. The first stage consists of estimating a vector **a** $\in \Re^d$ and a smooth function g, such as a spline or a polynomial, that minimize the residue r_1 with

$$r_1 = f(x) - g(\mathbf{a}^T \cdot \mathbf{x}) \qquad \textbf{(2.2)}$$

The process is iterated, starting from the residue r_i ($i \geq 1$), till the residue is small enough. Dimension reduction is achieved by keeping the largest values of **a** and setting the others to zero.

Exploratory projection pursuit

Exploratory projection pursuits are methods that search for interesting low-dimensions projections. The basic motivation to the method is furnished by the observation that the projection of mixtures of gaussian distribution (typically $d>10$) to a low-dimensional space (typically $d=2$) is normal (Diaconis, 1984). An interesting projection is therefore a projection in a low-dimensional space that differs strongly from a gaussian distribution. A projection index is used to characterize the projection. Exploratory projection pursuit searches for projections that maximize the projection index. Different indexes can be employed and we refer to Huber (1985) for a review paper.

Strong connections do exist between exploratory projection pursuit and neural networks. Back-propagation can be regarded as a projection pursuit., the mean-squares error taking the role of a projection index. The unsupervised BCM neuron can also be interpreted as a projection pursuit algorithm (Intrator,1992). The above methods have been used in various speech or face classification problems and in unsupervised feature extraction problems.

Dimension reduction through wavelets-based projection methods

Best basis

The best basis method is, together with the wavelet matching pursuit algorithm (Davis, 1994), the main wavelet-based algorithms for dimension reduction. The best basis (Coifman, 1992) consists of choosing a redundant basis of orthogonal functions to compress the signal. The orthogonal redundant basis is chosen such as the decomposition algorithm may be obtained from one set of filter coefficients.

The algorithm is best understood, if one starts from the wavelet decomposition algorithm. In the wavelet decomposition algorithm, the wavelet coefficients are computed with a decomposition tree made of a cascade of filters. At each decomposition level, the signal is decomposed into a low frequency and a high frequency component. At the next decomposition level, the low-frequency component of the signal is processed after decimation with the same two filters (fig. 2.5b). Using the same two filters, a large number of different signal decomposition are possible, by processing the high-frequency components of the signal further. A full decomposition tree at level J is given by the 2^J possible decompositions with 2 orthogonal filters. Each node corresponds to the projection on a different function. Each basis function is orthogonal to the other bases. The full decomposition tree for J=3 is represented in fig. 2.5b. The best basis method searches for a subtree (fig 2.5c) in which the signal is optimally projected. The best basis algorithm furnishes a method to choose a good basis for data compression among the set defined by the full tree.

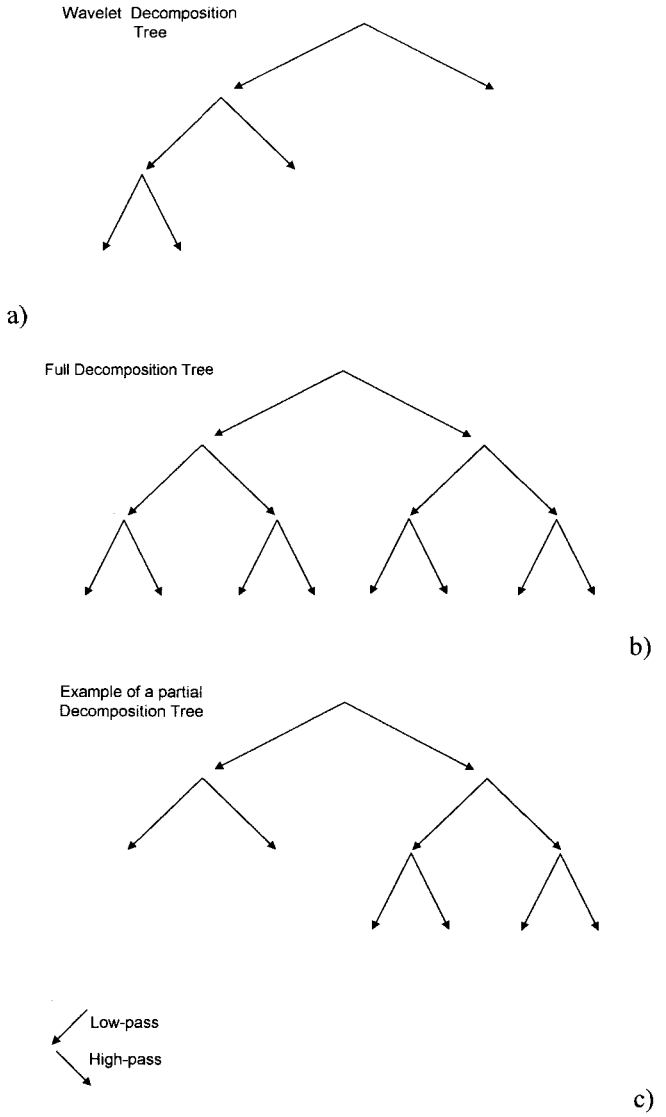


Figure 2.5: Different decomposition trees: wavelet decomposition tree, full decomposition tree, partial decomposition tree. The best basis method determines the partial decomposition tree that minimizes the entropy of the decomposition coefficients.

The search for the best basis among a dictionary of orthogonal bases consists of finding the basis that minimizes the approximation error:

$$\epsilon = \|f\|^2 - \sum_m |< f, g_m >|^2 \qquad \textbf{(2.3)}$$

This search requires very often too much computing time. The problem can be simplified, to the cost of optimality, by searching to minimize a cost function

$\gamma(x)$. The best basis is obtained by minimizing this cost function. The chosen basis is the partial tree with the minimum value of the cost function.

$$\sum_m \gamma(|< f, g_m >|^2 / \|f\|^2) \qquad \textbf{(2.4)}$$

For orthogonal wavelets, the entropy function S(x)= -x log x is taken and the best basis is obtained by minimizing the cost function.

$$\sum_m S(|< f, g_m >|^2) / \|f\|^2) \qquad \textbf{(2.5)}$$

This cost function has the advantage to be additive. The methods of dynamic programming can be used, what simplifies much the search. Generally speaking, it can be shown that the basis that minimizes the cost function in (2.5) corresponds to the Karhunen-Loève basis. Since the Karhunen-Loève basis does seldom belong to the dictionary, only a suboptimal solution is found by the best basis method. It follows that the best basis can be regarded as a method of finding a basis among a dictionary of redundant basis that approximates best the minimal cost function obtained by the Karhunen-Loève method. In comparison to the Karhunen-Loève algorithm, the best basis is much more efficient computationally and furnishes in most cases a signal approximation of a lower complexity than the Karhunen basis. Since, the best basis furnishes an approximation of the Karhunen-Loève basis, the best basis algorithm is sometimes described as a fast Karhunen-Loève transform (Wickerhauser (1992)).

Let us describe the best basis in more details. Given a function or an indexed datafile f, the entropy of the energy distribution on the basis $B=\{g_n\}$,with N basis function g_n, is according to (2.5)

$$S(f,B) = -\sum_{m=1}^{N} \frac{|< f, g_m >|^2}{\|f\|^2} \log_e \frac{|< f, g_m >|^2}{\|f\|^2}$$

The entropy function is additive in the sense that for any orthonormal basis B_0 and B_1 of two orthogonal spaces, one has

$$S(f, B) = S(f, B_0 \cup B_1) \qquad \textbf{(2.6a)}$$

$$\textbf{with } B = B_0 \cup B_1 \qquad \textbf{(2.6b)}$$

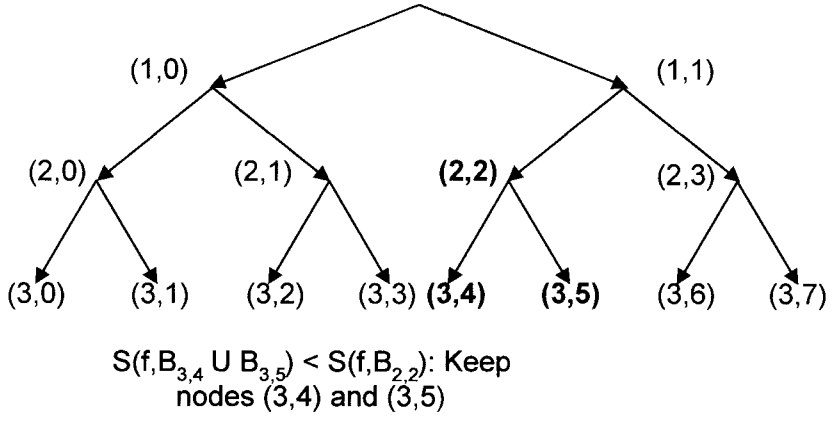
The best basis algorithm works as following

Step 1:

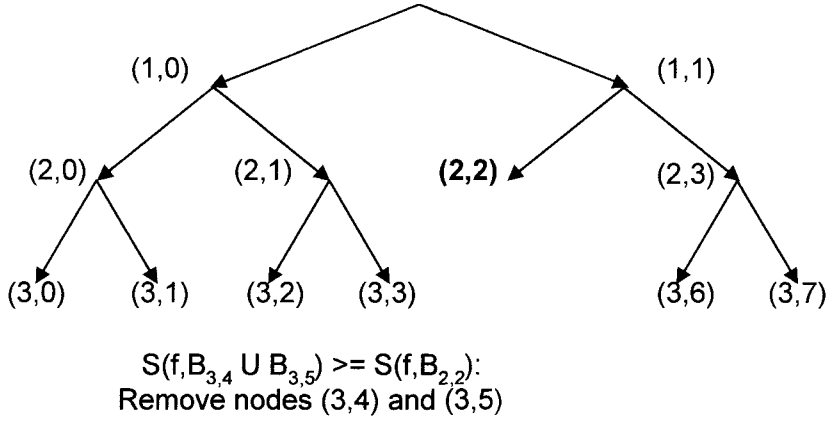
Choose a mother wavelet ψ and construct a full decomposition tree till level J. Compute the different S(f,B(j,k,)) with B(j,k) the basis corresponding to the node (j,k) in the full decomposition tree. Set j=J-1.

Step 2:
Prune each tree by iterating the following operations till j=0

If $S(j,k) \leq S(j+1,2k)+S(j+1,2k+1)$ then
remove node (j+1,2k) and (j+1,2k+1) else
set S(j,k) equal to $S(j+1,2k)+S(j+1,2k+1)$
j=j-1
The best tree is given by the pruned tree with the minimal entropy found by the above algorithm. Figure 2.6 illustrates the algorithm with an example.



a)



b)

Figure 2.6: Example showing the pruning algorithm of the best basis discriminant algorithm on the lowest level of decomposition for the two main cases:
a) $S(f,B) \geq S(f,B1 \cup B2)$; b) $S(f,B) < S(f,B1 \cup B2)$.

For non-orthogonal wavelets, the best basis is generally implemented using a different cost function:

$$\sum_m S(|< f, g_m >|)/\|f\|) \qquad (2.7)$$

Matching pursuit

The dimension of a signal can be decreased by determining a good decomposition of a signal as a weighted sum of a small number of wavelets. While the best basis algorithm uses a dictionary that can be all constructed from a single set of filter coefficients, the matching pursuit uses a generally much larger wavelet dictionary, that cannot be constructed from a single set of filter coefficients. For a general dictionary, the best basis algorithm does not work and a different algorithm, for instance the matching pursuit, must be implemented. The matching pursuit is a greedy algorithm. At each iteration a function that matches well some part of the signal is searched into the dictionary (Fig. 2.7). The contribution of the signal projection on this wavelet is removed and the process repeated with the residue. The algorithm is stopped when the norm of the residue is below a given threshold. More formally, consider a function f to be approximated as a sum of functions contained into the dictionary. The first step in the algorithm consists of finding, with *brute force*, a basis function g_i such as

$$|< f, g_i >| \geq \beta \cdot \sup_\gamma |< f, g_\gamma >| \qquad (2.8)$$

with $0<\beta \leq 1$ and γ indexing the different functions in the dictionary.
The function f can be rewritten as

$$f =< f, g_i > \cdot g_i + Rf \qquad (2.9)$$

With Rf the residue. Since the residue is by definition orthogonal to g_0, the following relation is fulfilled:

$$\|f\|^2 = |< f, g_0 >|^2 + \|Rf\|^2 \qquad (2.10)$$

It follows that the residue decreases at each iteration step. The convergence rate is related to the value β. Roughly, the smaller is β, the slower is the convergence rate (For an exact computation of the convergence rate, see Mallat (1993)).

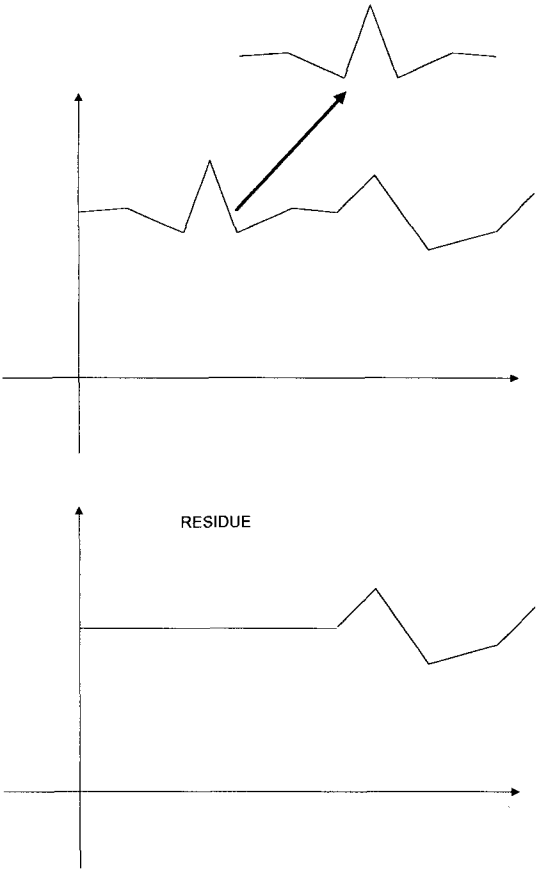


Figure 2.7: Illustration of the matching pursuit. The matching pursuit is a greedy algorithm.

Exploratory knowledge extraction

Finding the right model may represent a very large investment in effort and time towards a good description of a dataset. The search for good indicators and indexes in finance is a typical problem. Also, the modeling of sensors may need some extensive basic research work. During modeling of a complex unknown process from data, it is important to determine first what are the important variables and how much nonlinearity is necessary to obtain a satisfactory model. We will present two simple methods that we found to be quite useful for data exploration at a very early stage.

Detecting nonlinear variables interactions with Haar wavelet trees

The number of vanishing moments of a wavelet is related to the degree of the maximal order polynomial such as the projection of the polynomial on the wavelet is zero. A wavelet has n vanishing moments if

$$\int_{-\infty}^{\infty} t^k \cdot \Psi(x) \cdot dx = 0, (k < n) \qquad (2.11)$$

By definition, the projection of any n-1 order polynomial on a wavelet with n vanishing moments is zero. This property of wavelets can be used in exploratory data analysis to detect low-order nonlinear interactions. Let us illustrate the method with a simple example using the Haar wavelet.

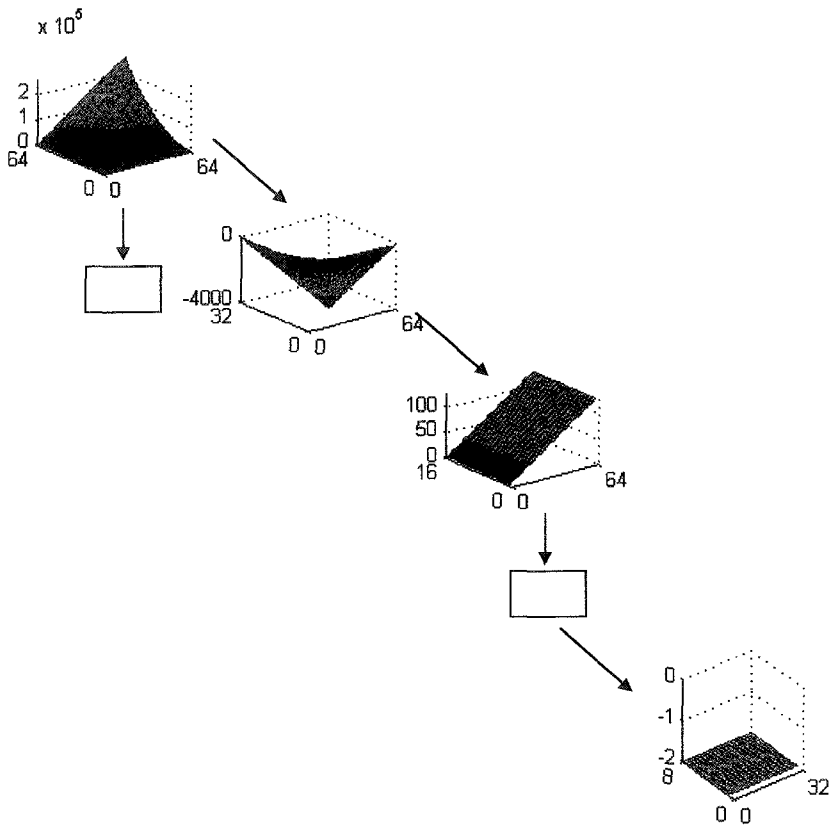


Figure 2.8: The Haar decomposition of a surface with a wavelet tree decomposition permits to discover low order interactions between data.

The Haar wavelet has one vanishing moment. Generally speaking, a wavelet transform with n vanishing moments can be put under the form of a differential

operator, equivalent to taking the n^{th} derivative. Therefore, the Haar transform is related to the first order differential operator. The wavelet coefficients obtained from a one level decomposition with Haar wavelets are proportional to the derivative of the surface along the considered direction. Similarly, the one-level wavelet coefficients of the derivative is linearly proportional to the second derivative. This property of the Haar wavelet can be used to discover nonlinear interactions between variables. Consider the function $y_3(x_i,x_j)=x_i^2\, x_j$ with ($0< x_i,x_j <65$, x_i,x_j integer). Figure 2.8 shows part of a decomposition tree. The first surface corresponds to the input data. After a wavelet decomposition along the x_i-axis, the surface defined by the wavelet coefficients is proportional to x_i. A further wavelet decomposition along the x_i-axis results into a plane. Finally, a decomposition along the x_j-axis results into equal and nonzero coefficients. From the above decomposition, one deduces that the equation of the original surface contains an interaction term in $x_i^2\, x_j$. Generally speaking, the identification of decomposition levels with large and equal coefficients is a good indicator for low-order nonlinear interactions between variables.

If the signal is a sum of terms with different interaction orders, the interaction terms at the highest order are identified and removed from the signal through an inverse decomposition in which all high order terms are set to zero. The same procedure is repeated till the low-order terms are identified. The method is quite efficient also when the signal is noisy. In this case, the Haar wavelet decomposition smoothes the signal by filtering out some of the high-frequency noise. A slightly different approach has been taken by Flehmig et al. (1998) to identify trends in process measurements. The method relies also on the fact that the number of vanishing moments is related to the degree of the maximal order polynomial such as its projection on the wavelet is zero (eq. (2.11)). The data are fitted to a m^{th} order polynomial through least-squares and the residue is taken to quantify the goodness of fit. The residue corresponds to the energy contained in the low resolution wavelet coefficients. A threshold may be set to the residue, below which a trend is validated. We would like to point out that Flehmig's approach can be greatly simplified by taking m^{th} order splines. As m^{th} order cardinal B-splines form a basis for m^{th} order polynomials, a wavelet decomposition of the data is sufficient to fit the data to a m^{th} order polynomial. Despite the fact that biorthogonal splines wavelets are non-orthogonal, the squared values of the low-order coefficients characterizes well enough the goodness of fit. This approach is computationally much less demanding than Flehmig' s approach as no least-squares computation is necessary. The strength of the method is its capability to detect trends locally at many different resolutions and to allow for nonlinear polynomial trends search.

Discovering non-significant variables with multiresolution techniques

Assume a databank containing the output data y_k as a function of a number of input variables. As an example, y_k may be the output of a sensor in volts and the input variables may be temperature, time, humidity, luminosity,.... In many cases,

the databank may contain irrelevant data or noisy data. The problem is now to identify which variables are relevant before using standard modeling tools for learning or knowledge discovery. The multiresolution approach uses the following procedure. After having approximated the different data on a fine grid, data are preprocessed with a standard multiresolution analysis using one variable at a time. For instance, if there are two input variables x_1 and x_2, then the output data y_k is given by a matrix. A one level wavelet decomposition of the rows is carried out, followed by a one level of decomposition of the columns. This process is then carried out as many times as necessary to reach a level with a very low resolution level.

Let us define the projection indices $E_1(x_i)$,$E_2(x_i)$,...,$E_L(x_i)$,...$E_J(x_i)$ for a given variable x_i. The projection index E_L corresponds to the normalized energy contained into the wavelet coefficients at level L.

$$E_L(x_i) = \sum_{n=1,\ldots,2^{L-1}} d^2_{L,n}(x_i) / \sum_{i=1,\ldots,J} E_I(x_i) \qquad \textbf{(2.12)}$$

Depending on the values of the projections indices, the variable may be assumed to be significant or not. If data are not noisy, then variables having only low projection indices should be discarded. The interpretation of the indices is more difficult for the case of very noisy data. As a guideline, very similar values of the projection at all levels is an indication that the signal may be simply a white noise signal along the variable x_i. Let show how to interpret the indices with two examples.

As a first example, consider the function $y_1(i, j)=0.04$ i

with (0<i,j<65, i,j integer).

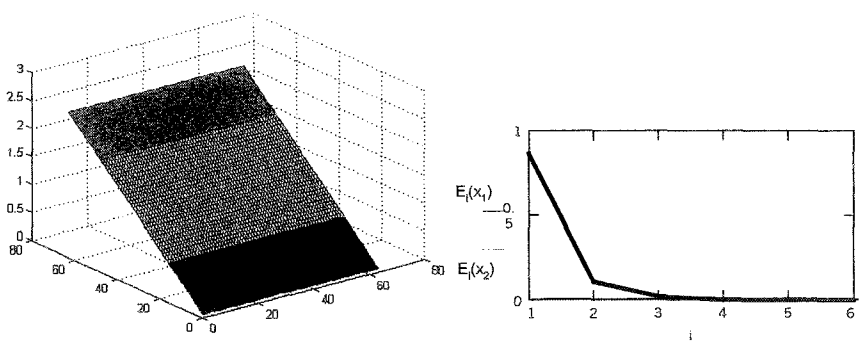


Figure 2.9: a) Surface given by the equation $y_1(i, j)=0.04$ i; Projection indices along the axis x_1 and x_2 for the curve in fig. 2.9a.

The output function $y_1(i, j)$ is independent of the variable x_2 and only depends on x_1. All the wavelet coefficients corresponding to the decomposition along the x_2 axis are zero, while the wavelet coefficients along the second axis have non-

zero values (Fig. 2.9). This means that the variable x_2 can be discarded and the variable x_1 can be most likely modeled at low resolution. If a control surface given by the above equation has to be modeled with a fuzzy controller, then a low granularity is sufficient.

A somewhat more difficult case is furnished by the second example. This example is more difficult in the sense that there is no unique interpretation of the projection indices. Consider the function $y_2(i,j) = 0.04\ i + \text{rand}(i,j)$ with $(0<i,j<65$, i,j integer and rand(i,j) a uniformly distributed random number between [-1,1]).

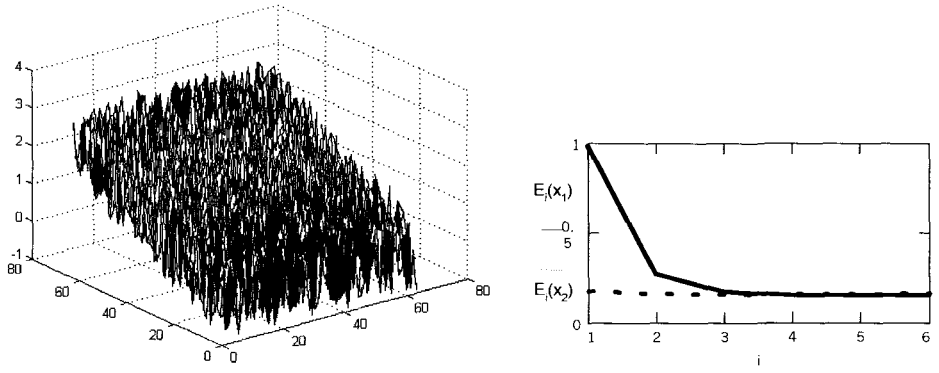


Figure 2.10: a) Same function as in fig. 2.9 except for some additional noise. b) Projection indices along the axis x_1 and x_2 .

The coefficient $E_1(x_1)$ corresponding to the lowest resolution level has the highest value. This suggests that the signal can be preprocessed so as to keep only the low-frequency component along the x_1-axis of the signal. The values of the coefficients $E_1(x_2)$ are very similar (Fig. 2.10). The signal along the x_2-axis is characteristic of a white noise signal. In this case, one may try to discard the variable x_2. Let us point out that the above situation may also occur under different circumstances. The signal may be deterministic but simply contains many components at various frequencies. In this case, a low-resolution modeling of the process will give quite bad results.

Wavelets in classification

Multiresolution analysis is obviously an important candidate method to discover important features in signals. On the one hand, it permits to compare features at different resolutions, a necessary ingredient in many images classification problems. On the other hand, multiresolution permits to identify features such as transients, edge, spikes that are often a fingerprint of the signal in many problems (speech processing, image processing, stock market fluctuations).

The contribution to classification problems of multiresolution analysis is two-folded. In many classification problems, multiresolution has been used to decrease the complexity of the data to the extent that a standard classification method becomes feasible. A second interesting development has been the appearance of classification techniques based on the best basis.

In a previous section, the *best basis* algorithm was introduced. The basic idea behind the *best basis* is to find a representation of a function by using an orthogonal basis that minimizes the entropy. The algorithm searches for a representation in which most of the signal energy is contained in a small number of coefficients, a low entropy representation. With slight modifications, the algorithm can be applied to classification. Instead of the entropy, the algorithm uses a different cost function: the relative entropy. The algorithm permits to extract a few important features (wavelet coefficients) that characterize well dataset belonging to different classes (Saito, 1994a, 1994b). In classification problems, the problem is not to find the best basis to compress a signal but to find a parsimonious signal representation which furnishes well-separated classes. In classification, the criteria for a good projection is a measure of the class separability.

Classification with local discriminant basis selection algorithms

The standard linear method in classification is the linear discriminant analysis. Linear discriminant analysis consists into bisecting the space with a number of hyperplanes. The method finds the bisection that minimizes the scatter of sample vectors within each class and maximizes the scatter of mean vectors between classes.

Let Mc be the mean vector of class c and M the total mean vector:

$$Mc = 1/N \sum_{1}^{N} Xc_i \tag{2.13}$$

with x_c the N points belonging to class c
The sample covariance matrix of class c is given by

$$\Sigma_c = 1/N \sum_{1}^{N} (Xc_i - M_c) \cdot (Xc_i - M_c)^T \tag{2.14}$$

The within-class covariance is

$$\Sigma_w = \sum_{c} \Sigma_c \tag{2.15}$$

The between-class covariance is

$$\Sigma_b = \sum_{c} (M_c - M) \cdot (M_c - M)^T \tag{2.16}$$

The linear discriminant analysis maximizes the class separability index J(s) which measures how much the classes are separated.

$$J(S) = tr[(S^T\Sigma_b S)^{-1}(S^T\Sigma_w S)] \quad \textbf{(2.17)}$$

The matrix S after solving is a diagonal matrix containing the eigenvalues. The matrix S describes a map $S^T x_i$ transforming the input space such as the class separability is maximized.

The linear discriminant basis method does have the disadvantage to be a global method. In some cases, a local method is necessary to separate correctly the data in different classes. This may be carried out by using different approaches involving either neural networks, self-organized maps or alternative methods. One of these alternative methods is the local discriminant basis selection algorithm (Saito, 1994a, 1994b). The best basis approach can be adapted to the problem of finding a good basis to discriminate signals belonging to a number of different classes.

Consider first a set of N_c one-dimensional training signals of same length belonging to a given class. The energy map of the class can be defined as

$$\Gamma_c(j,k,l) = \sum_c (\sum_l^{N_c} Wc^2_{j,k,l}) / \sum_c (\sum_l^{N_c} Xc_i^2) \quad \textbf{(2.18)}$$

with j giving the level of decomposition, k the branch in the tree at this level and l the position. Wc represent the wavelet coefficients in the decomposition and Xc the original data.

The Kullback-Leibler distance $D_{c1,c2}$ between two classes (Kullback, 1951) can be computed from the energy map: The Kullback-Leibler distance is a measure of the relative entropy:

$$D_{c1,c2}(j,k) = \sum_l \Gamma_{c1} \log \frac{\Gamma_{c1}(j,k,l)}{\Gamma_{c2}(j,k,l)} \quad \textbf{(2.19)}$$

The relative entropy is asymmetric. If a symmetric quantity is preferred, one can use the expression:

$$D \equiv D_{c1,c2} + D_{c2,c1} =$$
$$1/2(\sum_l \Gamma_{c1} \log \frac{\Gamma_{c1}(j,k,l)}{\Gamma_{c2}(j,k,l)} + \Gamma_{c2} \log \frac{\Gamma_{c2}(j,k,l)}{\Gamma_{c1}(j,k,l)}) \quad \textbf{(2.20)}$$

For several classes, one may take as measure of the relative entropy, the sum of all individual two-classes relative entropy. The algorithm to find the best basis, also called the local discriminant basis selection algorithm is very similar to the best basis algorithm. The relative entropy function D is additive in the sense that for any orthonormal bases B_0 and B_1 of two orthogonal spaces, one has

$D(f, B) = D(f, B_0 \cup B_1)$

with $B = B_0 \cup B_1$

It follows that the method of dynamic programming can be applied to search for the best basis. The best basis in of the discriminating power is the one that minimizes the relative entropy. The algorithm works as follows:

Step 1:
Choose a mother wavelet and construct a full decomposition tree till level J for the training data in each class. Compute the different relative entropies D(j,k) with j the decomposition level and k the position of the node in the decomposition tree. Set j=J-1.

Step 2:
Prune the tree by iterating the following operations till j=0

If $D(j,k) \leq D(j+1,2k)+D(j+1,2k+1)$ then
remove node (j+1,2k) and (j+1,2k+1) else
set D(j,k) equal to $D(j+1,2k)+D(j+1,2k+1)$
j=j-1

Step 3:
Keep the k most discriminant basis functions to construct the classifier.
Figure 2.11 shows a single step of the algorithm on the lowest level of decomposition for an given tree.

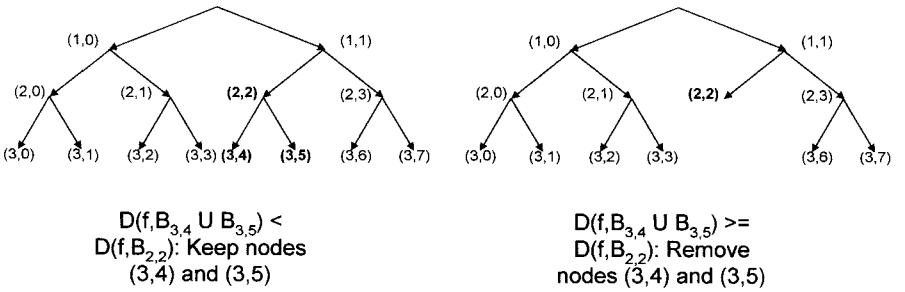


Figure 2.11: Example showing the pruning algorithm of the best basis discriminant algorithm on the lowest level of decomposition for the two main cases: a) $D(f,B) \geq D(f,B1 \cup B2)$; b) $D(f,B) \leq D(f,B1 \cup B2)$.

Classification and regression trees (CART) with local discriminant basis selection algorithm preprocessing

Local discriminant analysis can be combined with CART (Classification And Regression Tree). CART is a statistical model conceived at Stanford (Breiman (1984). CART belongs to the class of binary recursive partitioning. Parent nodes

are always split into exactly two children nodes. The process is recursive because the process can be repeated by treating each child node as a parent. A so-called maximal tree is grown using a variable or a linear combination of variables to determine splits. The results are presented under the form of a tree structure. After the tree is considered as complete, the tree is pruned. Figure 2.12 shows an example of a binary regression tree.

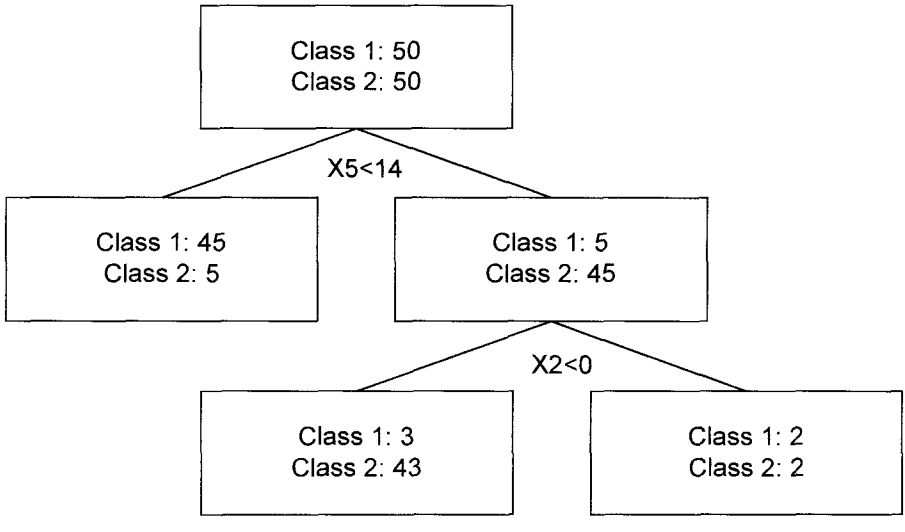


Figure 2.12: Example of a classification with CART.

Saito (1994b) has shown that the classification of seismic signals with a tree gives significantly better results if the local discriminant basis selection algorithm is used to preprocess the data. In this approach the tree is grown on the signals represented in the local discriminant basis selection algorithm. The classification tree is carried out with a classification and regression tree (CART).

As mentioned earlier, wavelets are not shift-invariant, which is sometimes a major drawback in classification problems. In order to solve this problem, shift-invariant multiresolution representations have been tested (Saito, 1992). This shift-invariant multiresolution is based on so-called auto-correlation shells, formed by dilations and translations of the auto-correlation functions of compactly supported wavelets. The shift-invariance is introduced by using a redundant wavelet decomposition. Auto-correlation shells have two properties that makes them attractive. First, if an orthogonal wavelet is taken to build the auto-correlation, the auto-correlation is an interpolating functions. Second, a fast algorithm permits to compute very efficiently the projection coefficients on the different functions.

Applications of multiresolution techniques for preprocessing in soft computing

This chapter gives a short review of the major achievements of multiresolution preprocessing techniques in connection to soft computing. Wavelets have been used in a significant number of applications. A review on wavelet transforms for pattern recognition can be found in Tang (2000) and in Szu (1999). Wavelet methods are implemented in a number of intelligent systems, for instance in power engineering (Ashenayi, 1997), for quality inspection of surface mounted devices (Brito, 1994), condition monitoring, inspection (Serrano, 1999) process monitoring in nuclear power plants (Schoonewelle, 1996) image enhancement (Qian, 1994) or registration in medical applications (Unser, 1996). Wavelet preprocessing has found a large range of applications in chemistry. In the following sections, we have selected a number of significant applications connecting wavelet preprocessing to soft computing.

Neural networks

A large number of applications combining wavelet techniques and neural networks use multiresolution analysis for preprocessing. Wavelet preprocessing serves typically the purpose of reducing the complexity of the problem, by selecting a number of characteristic features in the signal, features that are then fed to the neural network. Applications range from automatic target recognition (Park, 1997; Zhang Xun, 1996; Baras, 1994), face recognition (Foltyniewicz, 1996), Thai character recognition (de Vel, 1995), analysis of underwater acoustic signals (Dawn, 1993). The success of these techniques is somewhat paradoxical, in the sense that wavelet decomposition is a priori not the best method for pattern recognition due to the lack of translation invariance. It may be explained by the fact that if the number of examples is large enough, the neural network will be trained to recognize the statistical correlations patterns between wavelet coefficients at different levels of resolution. A large range of neural networks have been implemented: perceptrons, Kohonen (Deschenes, 1995) and self-organizing networks (Przylucky, 1997), cellular neural networks (Moreira-Tamayo, 1996).

Many applications of wavelet preprocessing are in the medical domain. Medical images are well suited to wavelet processing as often the information is contained in sharp edges or localized contrasts. Also the detection of life-threatening situations in neurosurgical operations may benefit from multiresolution techniques. The patient state may be evaluated from wavelet processed data of the intracranial pressure (Swiercz, 1998). Several papers address the problem of automatic classification of EEG signals (Hazarika, 1997; Halgamuge, 1996) with multiresolution techniques. Denoising of medical images, for instance magnetic resonance images (Sarty, 1997), is also a classical

application of wavelet preprocessing. A very interesting application of wavelets must be particularly mentioned. Linkens (1997, 1998) and Abbod (1998) have developed a closed-loop controller for monitoring the depth of anesthesia for patients undergoing surgical operation. The system uses neurofuzzy and wavelet analysis to monitor and evaluate the depth of anesthesia based on the auditory evoked response signals, heart rate, and blood pressure. The system has been developed in close collaboration with anesthetists. Depending on the evaluation of the depth of anesthesia, a target concentration is decided by a rule based fuzzy controller which is fed to a target controller infusion algorithm. This application is a good illustration of the power of hybrid approaches combining several signal processing methods.

Texture analysis has also benefited significantly from wavelet preprocessing. A number of articles consider the combination of wavelet preprocessing techniques together with a neurofuzzy classifier. Wang et al. (1996, 1997a) have followed such an approach to classify textures with a fuzzy ART model. Westra (2000) describes the application of wavelet and neurofuzzy classification techniques to identify defects in printed decorations. Wavelet preprocessing has been implemented also in a system classifying automatically clouds. Features from clouds' texture are extracted and fed into a neural network in order to determine the type of a cloud (Shaikh, 1996).

A further application domain is signal enhancement, for instance to enhance characteristic features in face recognition problems or in quality controls (Ko, 1995).

We would like to mention a number of interesting applications in sensors that combine wavelet preprocessing to neural networks. Pratt et al. (1995) use wavelet methods to preprocess signal from electromagnetic sensors in order to diagnose the depth and nature of buried waste. The method is non-invasive and the success rate is quite high. Using force, strain or vibration sensors, the condition of tools can be estimated (Kamarthi, 1997; Zhou, 1995). Another application is automatic sensor recalibration (Padgett, 1998; Kunt, 1998). Gas sensors or electronic noses are known to drift over time and to require from time to time a recalibration. As field recalibration is generally critical, recalibration should be done only when it is absolutely necessary to ensure the correct functioning of the sensors. The typical signature of sensors is used as criteria for deciding on whether a recalibration is required.

Other examples are in the field of quality inspection and condition monitoring. In condition monitoring, the vibration signature measured from a number of strain or force sensors is compared to prototype vibrations. As an example, Samual et al. (1997) use piezoelectric strain sensors to identify the vibration signature from a planetary geartrain under faults conditions. In quality inspection, an image of an object is taken, for instance with a CCD camera and wavelet transformed. Defective pieces are found by comparing some wavelet extracted features to a template image.

Fuzzy logic

Image processing is an important domain of application for multiresolution hybrid techniques. Besides the already mentioned applications in neurofuzzy techniques, a number of applications combine fuzzy logic to multiresolution analysis. Let us mention here image queries software using a fuzzy similarity matching of the wavelet transformed color image (Tolias, 1999). Hybrid methods combining fuzzy logic and multiresolution analysis have been also developed for contour extraction and segmentation (Wavelet-based contour extraction rely on the high-pass filter characteristics of the wavelet decomposition.) In those approaches, the wavelet preprocessed images are analyzed subsequently by a fuzzy system. Examples of applications in computer tomography of the brain and digital mammography can be found in Cheng (1998).

Multiresolution analysis has been used in conjunction to fuzzy logic in automatic target recognition, tracking and image registration. The multiresolutional character of a wavelet decomposition is especially useful when the distance to an object is not known. Wang et al. (1997b) describe a computer vision system for automatic target recognition and tracking. Target recognition is carried out using a morphological neural network, while wavelet analysis is used for tracking. A fuzzy module integrates the results from different frames.

Segmentation of an image can be enhanced by including information on the image texture. Betti et al. (1997) improve the texture discrimination on synthetic aperture radar data by using a fractal representation of the texture derived from the wavelet coefficients.

A recent development has taken place in the domain of denoising. One of the most critical issue in wavelet denoising is the choice of the thresholding method and parameters. The choice of the thresholding parameters is generally based on relatively simple statistical criteria. A bad choice of the parameters leads either to overfitting or underfitting the data. Fuzzy systems have been proposed to determine adaptively the thresholding method or parameters (Shark, 2000).

Genetic algorithms

At time, the combination of multiresolution methods with genetic algorithm is still in its infancy. An important reason for that situation is that in many problems, the best basis or the matching pursuit algorithms are often superior to genetic algorithms (Lankhorst, 1995). Matching pursuit or the best basis approach are for instance the favored methods to determine a good wavelet basis in a dictionaryy. Nevertheless, there are situations for which genetic algorithms may be preferred. For instance, genetic algorithms can be implemented to select wavelets among a very large dictionary (Lee, 1999; Tagliarini, 1996). Such an approach has been used in texture classification (Naghdy, 1997) and radio transients identification systems (Toonstra, 1996). Also, some automatic target recognition (Wilson, 1996) and image registration systems have been optimized with genetic algorithms. The image is first preprocessed and well discriminating

wavelets coefficients are selected with the genetic search. Related approach were used by Chalermwat (1999) for image registration and Liao (1998) for texture classification.

Genetic algorithms have also found applications in a number of other situations. Genetic algorithms are quite commonly used to optimize neural networks or systems of several neural networks in problems where a gradient descent is not optimal. It is therefore not surprising that optimization of wavelet networks with genetic algorithms have had some success (Yang, 1997).

Hybrid methods combining genetic algorithms and multiresolution analysis have been tested in finance for trend analysis. Let us give here two examples. Recently, a fuzzy inference system for predicting stock trends has been designed by optimizing membership functions with a genetic algorithm (Kishikawa, 2000). The system predicts trends based on the multiresolution analysis of past data. The second application is on exchange rate forecasting. The method uses a neural network to predict future exchange rates. The efficiency of the network depends on the quality of the input data. Data are preprocessed by denoising them with a wavelet thresholding method (see 1.7.3). We have mentioned in the previous section, that finding a good threshold method for wavelet denoising is a very important and difficult problem. In that application, the value of the threshold is selected through a genetic algorithm (Taeksoo Shin, 2000).

Genetic algorithms will find many more applications in connection to multiresolution analysis, especially in multi-objective optimization and search problems, or situations in which strong constraints are set on the space of possible solutions. As a final remark, we believe that part of the essence of genetic and evolutionary computing can often be expressed in the language of multiresolution. We show in part 8, that multiresolution analysis and genetic algorithm can be combined in *wavelet-based genetic algorithms* and *multiresolution search methods*. We refer to part 8 for more information.

Application of multiresolution and fuzzy logic to fire detection

Smoke detectors have suffered over many years of the so-called *false alarms* problem (Thuillard, 1994). A spectacular improvement was reached in the last years, due to the combined influence of more reliable components, better alarm organization and the appearance of high quality microprocessors in the low price segment. Signal processing is playing an increasing role in smoke detectors as more sophisticated algorithms are being developed. Fuzzy logic has had a central role in fire detection. The acceptance of fuzzy logic has been quite large after the first successful implementation of fuzzy logic in fire detectors. At the beginning, fuzzy logic has been used mostly in classifiers, while with time fuzzy logic did get used to describe the alarm surface in multisensors detectors.

Linear beam detector

We would like to present two commercial products that have significantly benefited from fuzzy logic. The first example is a linear beam detector. The basic operation principle is documented in fig. 2.13. An energetic pulse is emitted by a LED and travel a distance between typically 5m to 150 m. The light is reflected back to the detector by a high quality retro-reflector. The detector will go into alarm, if a certain smoke level is in the beam path.

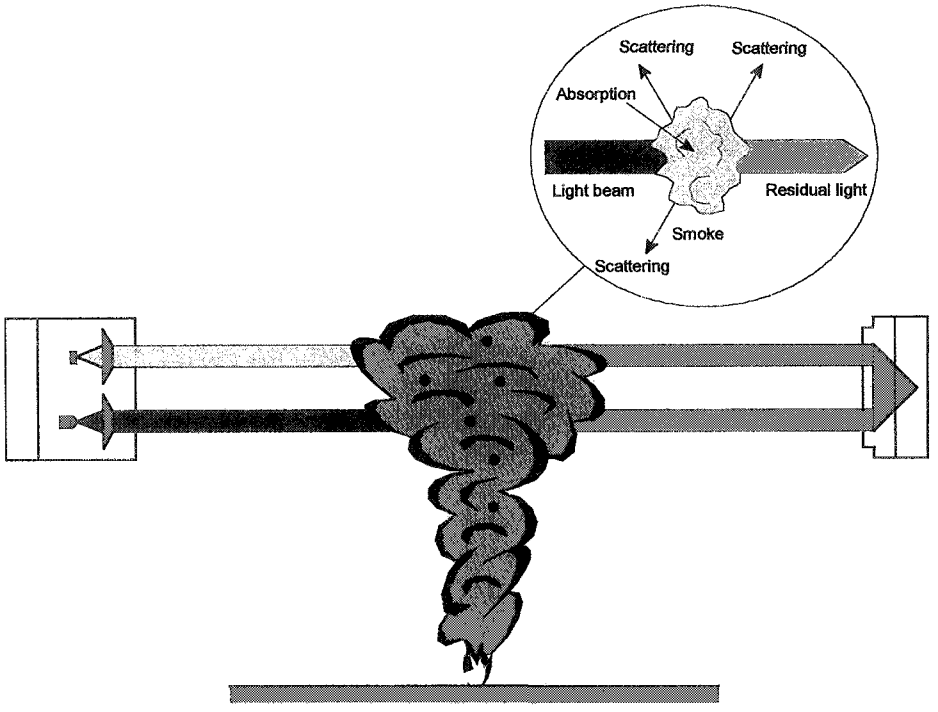


Figure 2.13: A linear beam detector is a smoke detector in which light attenuation is used as alarm criteria.

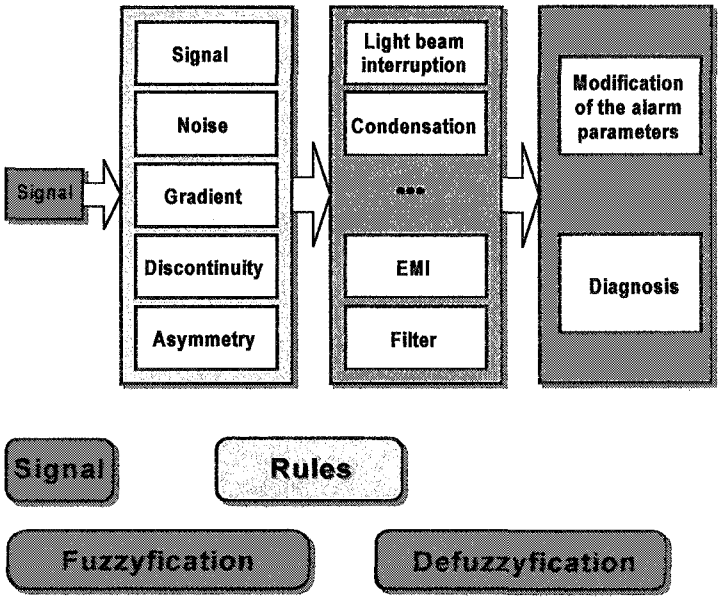


Figure 2.14: The signal of the linear beam detector is analyzed on-line by a system containing a number of fuzzy rules.

In order to prevent false alarms, fuzzy algorithms were developed that are capable of distinguishing the signature of non-fire (signal intermission due to a moving object: a bird or a truck, signal attenuation caused by mixing of cold and hot air,...) and fire events. In a first stage, features are extracted from the signal. These features are combined with a number of fuzzy rules (fig. 2.14). These rules serve two purposes, first to furnish a differentiated diagnosis of potential problems to the operator, and second to modify the alarm parameters depending on the diagnosis (Thuillard, 1996).

The algorithms were developed with a neurofuzzy method, using a multiresolution Kohonen type of network for signal classification. Data were collected in extensive field testing and fire testing. Data were classified using a constructive multiresolutional approach. At first, the network was optimized with only two membership functions per variable using a Kohonen network to determine the best partition and a simulating annealing method to optimize the shapes of the membership functions (fig. 2.15). The rules were then validated and the data corresponding to the validated rules were removed. The procedure was repeated by splitting the membership functions in two new membership functions to refine the fuzzy rules.

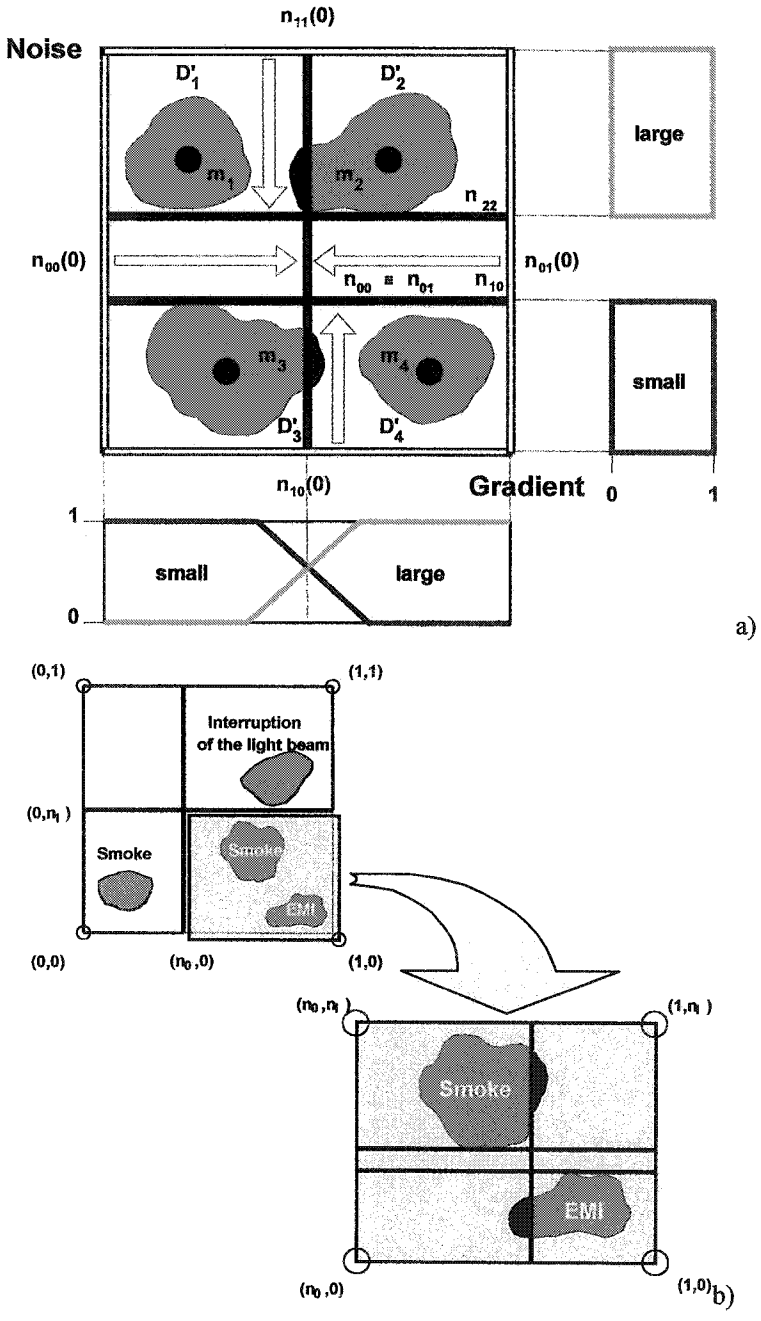


Figure 2.15: a) A Kohonen-based neurofuzzy system was developed and implemented during the development of the rule-based fuzzy system. b) A multiresolutional approach permits to add new membership functions iteratively.

Flame detector

Wavelet theory can be combined into an efficient spectral analysis method. We would like to discuss an example in some details to illustrate the power of the method. Flame detectors use pyroelectric sensors to record in the infrared domain the radiation emitted by flames. Flame radiation is measured at three different wavelengths.

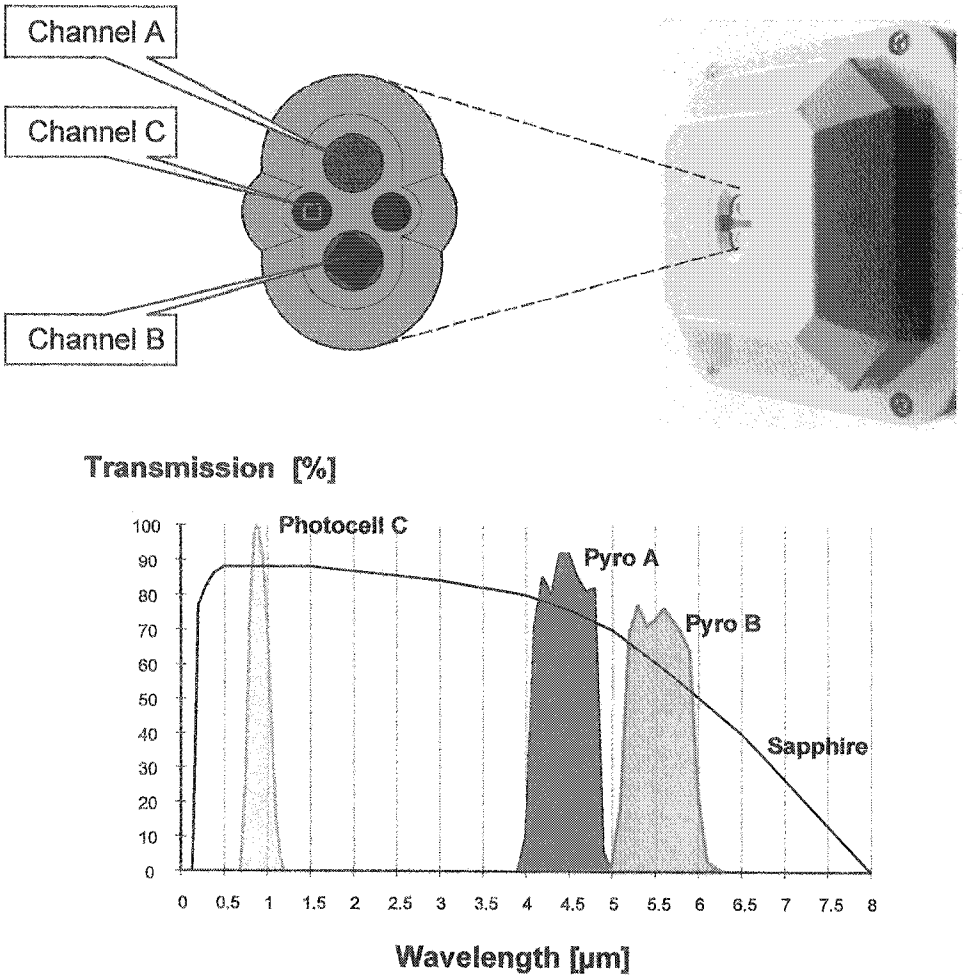


Figure 2.16: A flame detector records the radiation of a flame at a number of wavelengths. The signal is analyzed with a number of rules in the spectral domain and in the frequency domain.

Flame detectors must be sensitive enough to detect small fires, but must not be fooled by deceptive phenomena. Sun radiation and strong lamps are the main dangers for flame detectors. The ratio between the sun and a flame radiation is much lower in the infrared than in the visible range. For that reason, the radiation

is measured in the infrared domain and not in the visible. Even in the infrared, sun radiation is typically much larger than the radiation of the smallest fires, one wants to detect. By chance an hydrocarbon fire is characterized by two features. First the flame pulsates and second the flame emits strongly around 4.3 μm , the emission line of CO_2 (fig. 2.16). The ratio between the signal at the different wavelengths and the spectral analysis of the flame fluctuations can be used to characterize a true fire (Thuillard, 1999c). The spectral analysis is carried out, on-line, by a method combining fuzzy logic to wavelet analysis (fig. 2.17).

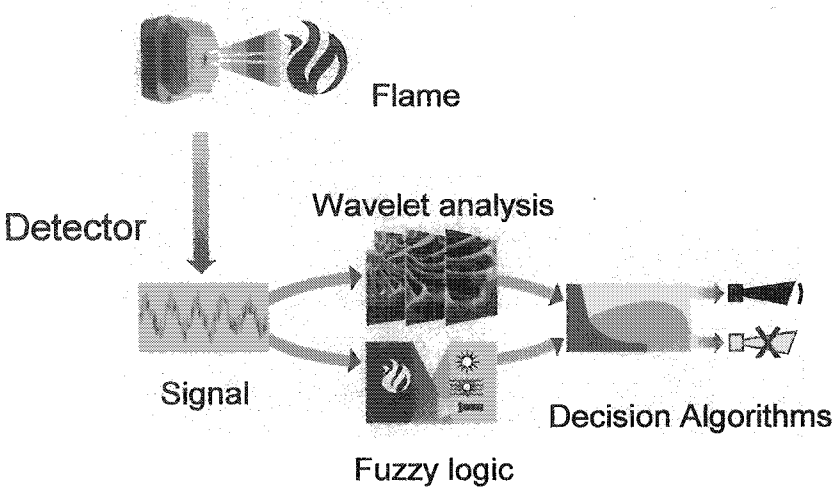


Figure 2.17: Spectral analysis, feature extraction and classification are made by combining fuzzy logic and wavelet analysis.

Under laboratory conditions, an hydrocarbon fire pulsates at a very regular frequency in the range between 0.5 Hz to 13 Hz. The larger the fire, the smaller is the pulsation frequency. A very simple law describes the pulsation frequency. The pulsation frequency is inversely proportional to the square root of the fire diameter. Amazingly, the pulsation frequency is in first approximation almost independent of the fuel. This can be explained by the fact, that a purely hydrodynamics instability causes the flame to oscillate. This instability is due to the density difference between the very hot air and the surrounding air flow, giving raise to an unstable gravity wave. We have suggested that the regular flame pulsation results from a resonant effect, that takes place, when the wavelength of the gravity wave is in a simple ratio to the fire diameter.

In real world applications, the regular flame pulsation may be easily perturbed. For instance, if a window is open, then an air drought may destroy the regular flame pulsation. We found out that even under those circumstances, flame pulsation still has some typical features. In order to understand why, we did carry

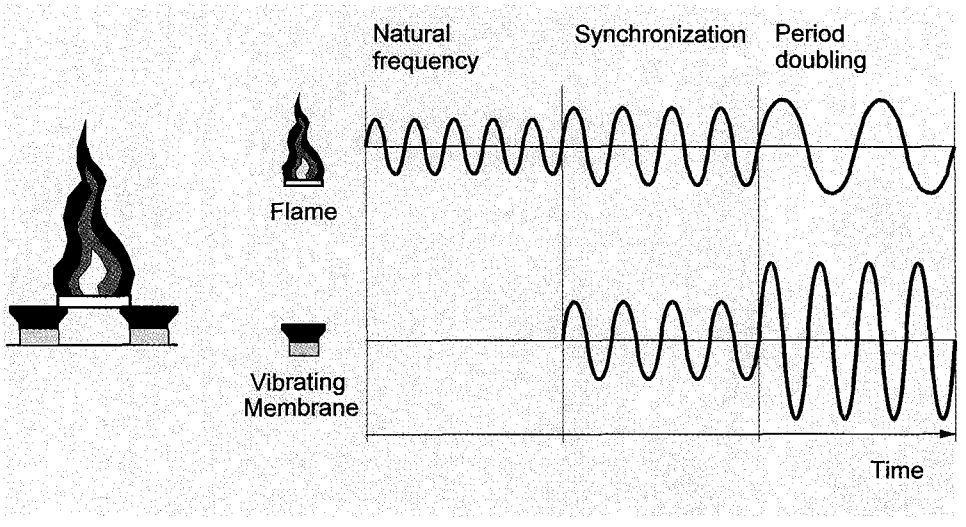
out a number of controlled experiments with oscillating membranes. With such experiments, we have been capable to show that external perturbations couple parametrically to a flame.

Flame pulsation can be quite well modeled by a self-excited van der Pol oscillator with parametric coupling (Thuillard, 1999b):

$$X'' + \omega_0{}^2 \cdot X + a \cdot (X^2 - K) \cdot X' = F(\omega, t) \cdot X \qquad \textbf{(2.21)}$$

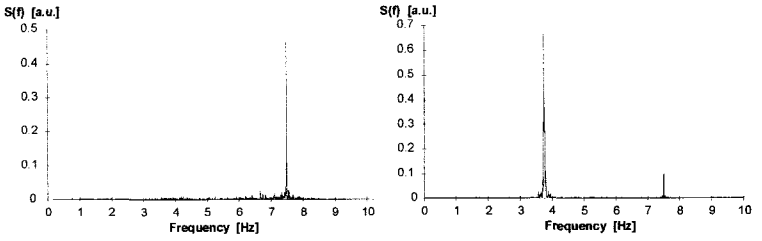
with $F(\omega, t)$ describing the perturbation, a and K constants, ω_0 the natural pulsation and X the average flame radiation.

Figure 2.18 compares the pulsation of a flame excited with an oscillating membrane to the van der Pol model. At low excitation, the flame does couple to the membrane oscillating at a frequency of the order of the natural flame pulsation frequency. As the amplitude of the excitation is increased, a bifurcation to the subharmonic takes place. The flame begins to pulsate at exactly half the excitation frequency. The comparison of many experiments with very different excitations did furnish a qualitative understanding of real world situations. The final outcome of this research was a catalog of the possible flame fingerprints.

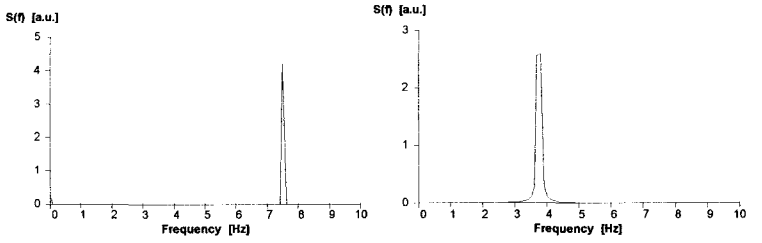


a)

EXPERIMENTS



MODEL



b)

Figure 2.18: Fundamental research on the physics and dynamics of flames did lead to a new model of flame pulsation. a) In an experiment, a flame was excited with an oscillating membrane. Depending on the amplitude of the oscillation, the flame does synchronize on the excitation frequency or period doubling is measured. b) The experiment can be modeled qualitatively very well with a forced van der Pol model.

The exploitation of these fundamental research results did require an efficient spectral analysis method to recognize the different fingerprints. Short time Fourier transforms in combination to a classifier was considered. The necessary power was too high for the microcontroller at our disposal (The main limitation is given by the electric power required for computation, not the computing power of the microcontroller. Very low electric power is allowed in fire detection to permit battery operation in emergency situations, where the normal power supply may be down!) .

An alternative to the Fourier transform was furnished by wavelet theory. Recall first, that a wavelet decomposition can be carried out by using a cascade of filters. A filter is associated to each level of resolution of the wavelet decomposition. For orthogonal wavelets, the energy conservation relation holds. It follows that the low-pass and the high-pass filters corresponding to the two decomposition filters for the first level of decomposition fulfill the power complementarity condition, as illustrated in fig. 2.19.

$$\sum_{m=1..p} |T_m(\omega)|^2 + |T_{low}(\omega)|^2 = 1 \qquad \textbf{(2.22)}$$

In a wavelet decomposition, the signal after low-pass filtering and decimation is filtered with the very same two filters. This corresponds to splitting the low-frequency filter band into two new bands.

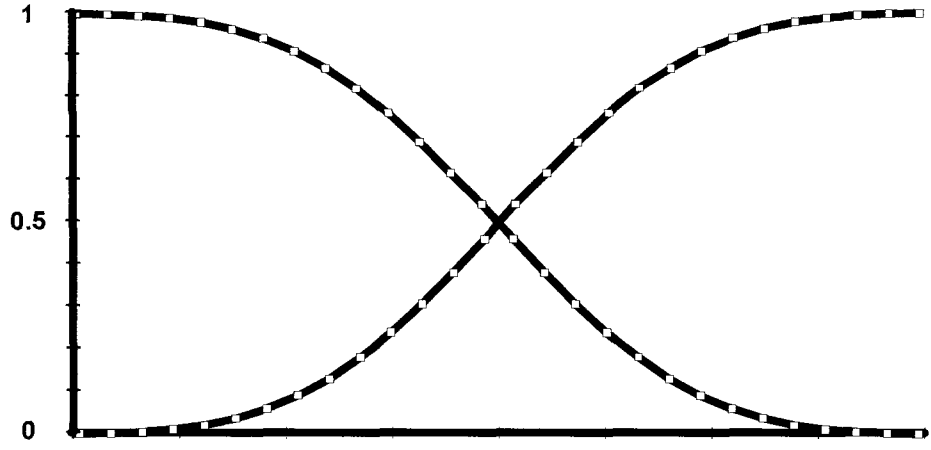


Figure 2.19: The filters associated to a one level wavelet decomposition fulfill the power complementarity condition.

At any decomposition level, the corresponding filters do fulfill the power complementary condition. The filter transmission functions in fig. 2.20 can be interpreted as fuzzy variables, for instance *low frequency* or *very high frequency*. The degree of membership μ_{T_m} are estimated from the wavelet coefficients by the expression (Thuillard, 1997, 2000a):

$$\mu(T_m) = \sum_n (d_{m,n})^2 / (\sum_{m=1}^{p} \sum_n (d_{m,n})^2 + \sum_n (c_{low,n})^2) \qquad \textbf{(2.23)}$$

The method possesses a number of advantages. The spectral analysis and the classification stage are blended into a set of fuzzy rules of the form:

if (frequency in spectral band 1 is A AND....) **(2.24)**
then...

With this approach, fuzzy rules in the frequency domain are simple and computer efficient. Rules in the frequency domain can be combined to rules describing other important criteria in flame detection, such as for instance the

degree of correlation between the signals in the different spectral bands. The analyzing wavelet function must be taken with great care. It determines the filter transmission as well as the number of coefficients. At a fixed sampling rate, the filter length determines essentially the time resolution of the method.

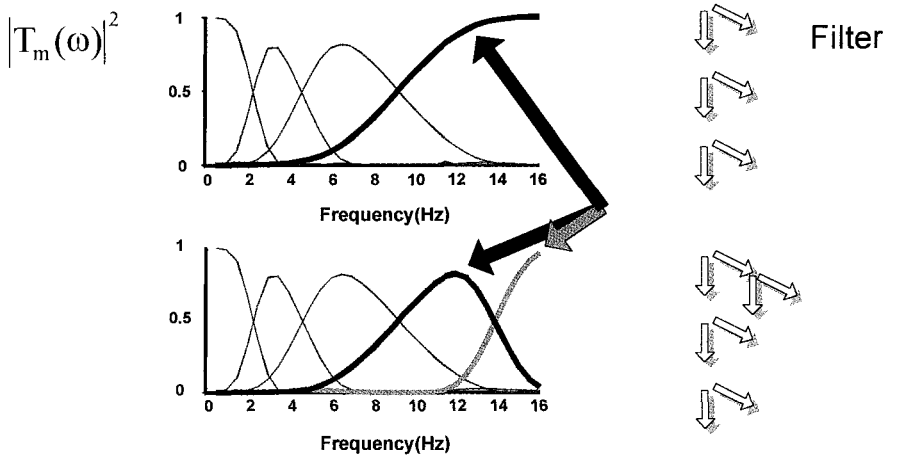


Figure 2.20: By cascading filters, a series of filters fulfilling the power complementarity condition is obtained. The transmission functions can be interpreted linguistically as membership functions.

The spectral analysis can be made very flexible by extending the method to wavelet packets. The wavelet coefficients may be further decomposed with the two wavelet decomposition filters. The power complementary condition is still preserved, but it does introduce two new variables. An example is given in fig. 2.20 with the lower filter tree.

Let us mention finally that a related approach was used in a medical application by Linkens (1997) to assess the depth of anesthesia.

This page is intentionally left blank

PART III SPLINE-BASED WAVELETS APPROXIMATION AND COMPRESSION ALGORITHMS

This page is intentionally left blank

3. Spline-Based Wavelets Approximation and Compression Algorithms

In the first section, a short introduction on cardinal B-spline is given. Cardinal B-splines are polynomial spline functions with equally spaced knots that have the nice property to be defined recursively by integral functions. In the following sections, three types of wavelet constructions based on B-splines are presented (Biorthogonal, semi-orthogonal and orthogonal). These represent a selection among the many spline-based wavelets. Their choice is motivated by the fact that these wavelets can be implemented in combination to fuzzy logic. Each of these three spline-wavelets permits to cover well an important aspect of so-called fuzzy-wavelet methods:

-Biorthogonal spline-wavelets are typically implemented in fuzzy wavelet networks. These wavelets have the great advantage to have compact supports, a useful property for on-line learning.

-Semi-orthogonal spline-wavelets are very useful in off-line learning, since they are the spline-wavelets the closest to being orthogonal.

-Orthogonal spline-based wavelets are good candidates to develop fuzzy rules in the frequency domain.

Spline-based wavelets can be implemented in multidimensional problems. Multidimensional wavelets can be constructed by using cartesian products of univariate spline wavelets.

Spline-based wavelets

Introduction to B-splines

The theoretical foundations of spline decomposition lies into the work by Schoenberg (1946). The first applications of splines methods came quite later. Besides the domain of surface fitting, splines have been implemented in computer graphics (Bartels, 1987; Diercks, 1995) and quite broadly in sophisticated medical applications (Carr, 1998). Splines are used in 3D animations and it would lead us beyond our topic to discuss advanced splines methods, such as for instance the Non-Uniform Rational B-Spline (NURBS). Splines are found in several commercial tools. Despite the fact that the field has reached a quite mature state, new important works do still appear in the literature.

In particular, the integration of multiresolution into the scope of spline research (Sweldens, 1995; Cohen, 1992) has opened up the field quite broadly. In this introduction, we limit the discussion to B-splines. A recent review on splines can be found in Unser (1999).

A B-spline function is a piecewise polynomial function defined on a lattice. The order of the spline function determines the properties of the spline. The simplest spline function is the characteristic function of the unit interval. The characteristic function $N^1(x)$ or Haar function (Haar, 1910) is defined as

$$N^1(x) = \begin{cases} 1 & 0 \le x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

The characteristic function is a piecewise zero order polynomial function (fig. 3.1).

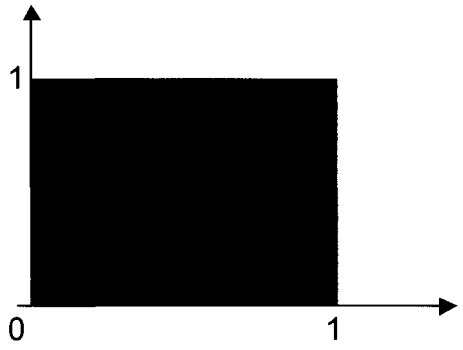


Figure 3.1: The characteristic function.

Cardinal B-spline functions of higher orders can be defined iteratively by the integral equation:

$$N^k(x) = \int_0^1 N^{k-1}(x-t)\cdot dt \tag{3.2}$$

As an example, the second order cardinal B-spline is the triangular function, a continuous function summing piecewise polynomials of order 1. More generally, a k^{th} order cardinal B-spline is C^{k-2} continuous, and are made of piecewise polynomials of degree k-1.

B-splines have a number of important and useful properties:

-The B-spline functions are the polynomial splines with the shortest support.

-All values are positive.
-Splines have a closed-form formula as their are piecewise polynomials. Spline-based wavelets are the only wavelets with such a property (Unser, 1999).
-Spline functions can be used to form a partition of unity, by using a superposition of translated B-splines. $\sum_{j=-\infty}^{\infty} N^k(x+j) \equiv 1$.

This is illustrated in fig. 3.2 for the case k=2. This property is very useful in order to give a fuzzy interpretation to a spline decomposition.

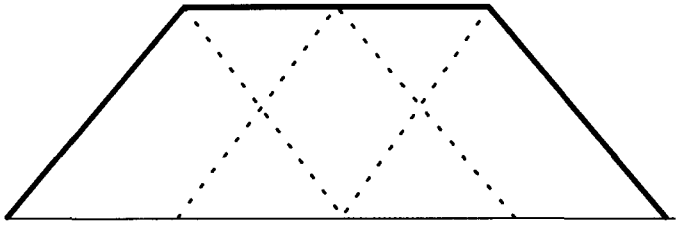


Figure 3.2: Translated cardinal B-splines partition the unity.

-Suppose a function can be put under the form $y = \sum_j a_j \cdot N_j^k$ with $N_j^k = N^k(x-j)$ and k>1. The derivative of y can then be written as: $dy(x)/dx = \sum_j (k-1) \cdot (a_{j+1} - a_j) \cdot N_j^{k-1}$.

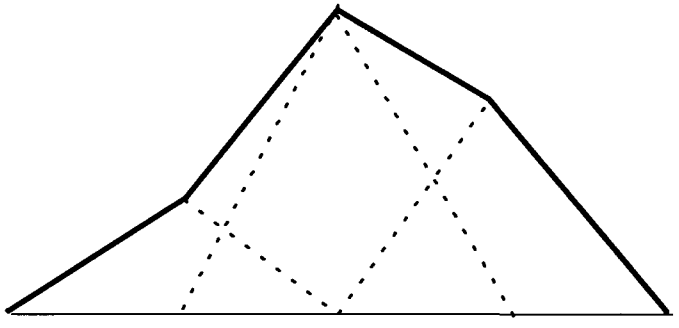


Figure 3.3: Example of a spline decomposition. The black curve can be decomposed into a sum of second order splines.

Splines are commonly used in function approximation (fig. 3.3). Multivariate B-spline basis functions can be formed by multiplying univariate basis functions (fig. 3.4). A basis function for a n-variables system is

$$N_j^k(\mathbf{x}) = \prod_{i=1}^{n} N_{j,i}^k(x_i) \qquad \textbf{(3.3)}$$

The good properties of univariate B-splines, such as bounded support and piecewise polynomial description, still hold in the multidimensional case.

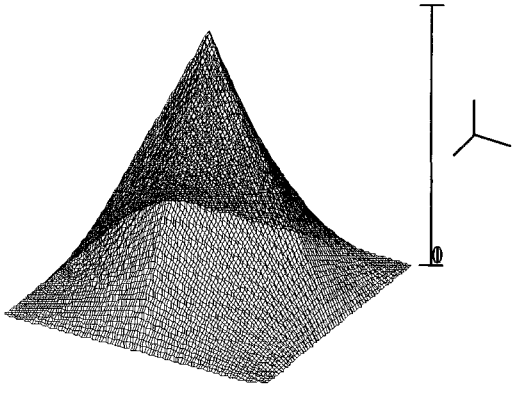


Figure 3.4: Example of a two-variate B-splines based on the triangular function.

Biorthogonal spline-wavelet

Biorthogonal spline-wavelets have a number of properties that are quite useful in real applications. On one hand, both the wavelet $\psi(x)$ and its dual $\tilde{\psi}(x)$ have a compact support, on the other hand, the scaling function $\phi(x)$ is always positive. This permits to interpret the scaling functions as membership functions in a fuzzy framework. Biorthogonal spline-wavelets are typically used in wavelet networks and also in on-line problems in which a simple method to process the boundaries is necessary.

Cohen et al. (1992) have shown how to construct biorthogonal spline-wavelets based on compactly supported splines. Their method is quite general and offers a great flexibility in the design of wavelets. For instance, the

construction permits to choose to a large extent the number of vanishing moments. Let us recall that a wavelet has n vanishing moments if:

$$\int_{-\infty}^{\infty} t^k \cdot \psi(x) \cdot dx = 0, (k < n) \qquad \textbf{(3.4)}$$

The number of vanishing moments, the degree of the scaling function and the support length are parameters that are taken into consideration as a wavelet family is chosen for an application. Figure 3.5 shows spline biorthogonal wavelets indexed as (4,2). The scaling function is the second order cardinal B-spline and the function ψ has 4 vanishing moments.

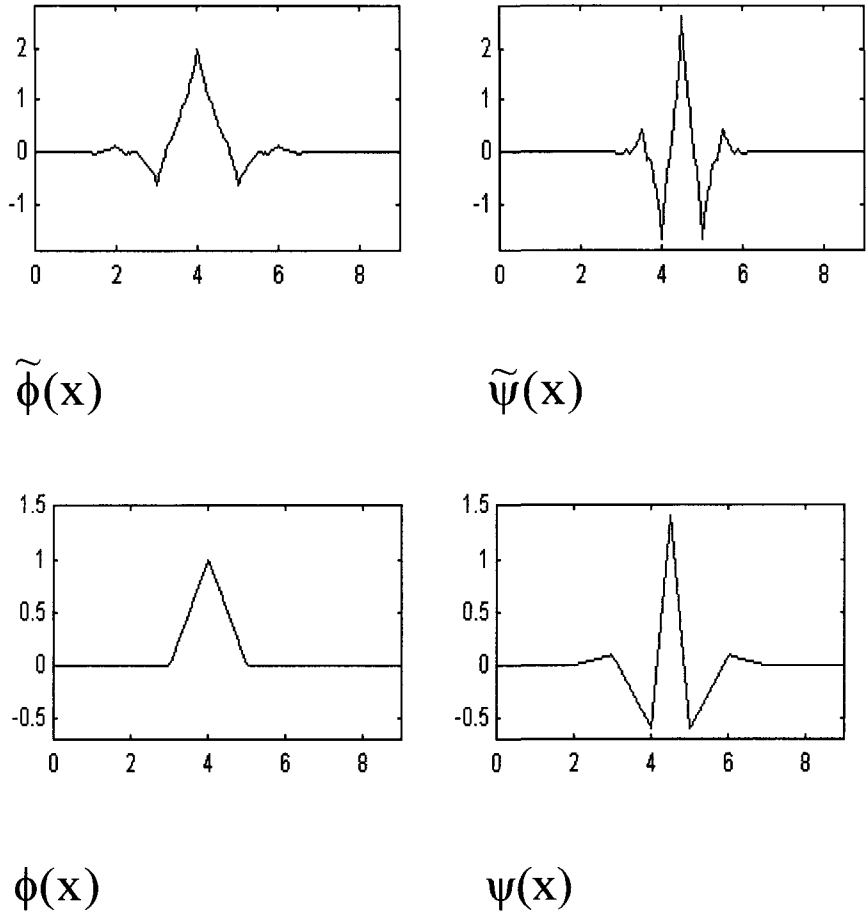


Figure 3.5: Biorthogonal spline scaling and wavelet functions, together with their duals.

Biorthogonal wavelets fulfill the biorthogonality condition $< \tilde{\psi}_{m,n}, \psi_{m',n'} >= \delta(m-m') \cdot \delta(n-n')$ (fig. 3.6).

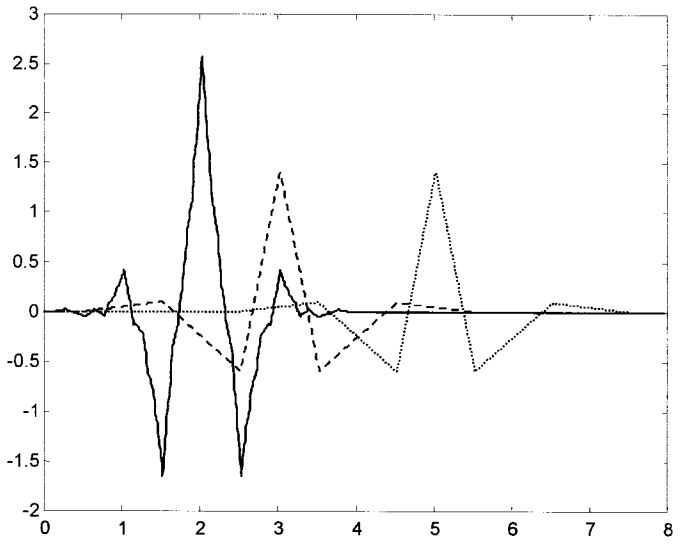


Figure 3.6: The dual wavelet $\tilde{\psi}$ (solid line) is orthogonal to any wavelet $\psi_{m,n}$. The two wavelet functions in that example are orthogonal to the dual wavelet. The filter coefficients for the wavelet decomposition and reconstruction algorithms are given below:

	k = 2	
j	p_j	q_j
1	0.994368911	0.707106781
2	0.419844651	-0.353553905
3	-0.176776695	
4	-0.066291260	
5	0.033145630	

	k = 2	
j	g_j	h_j
1	0.707106781	0.994368911
2	0.353553905	-0.419844651
3		-0.176776695
4		0.066291260
5		0.033145630

An example showing how spline-wavelets can be constructed is given in the annex.

Semi-orthogonal B-wavelets

Some of the most efficient algorithms in multiresolution analysis work best with orthogonal wavelets. The reason is that orthogonal wavelets fulfill the power complementarity condition. The squared value of the detail coefficients can be used to estimate the energy contained in the different projections. The total energy is the sum of the energy contained in the detail and the lower level approximation coefficients. This property is used to determine which coefficients should be kept in approximation, compression or denoising problems. There are no orthogonal wavelet constructions with a spline as a scaling function. Only semi-orthogonal wavelet constructions are feasible. For semi-orthogonal wavelets, the power complementarity condition does not hold. The squared-value of the wavelet coefficients together with the approximation coefficients is not equal to the total energy contained in the signal. Nevertheless, the energy contained in the signal at the different levels of resolution sums up to the total energy. This property is in many cases sufficient for many algorithms to work very well.

Orthogonal wavelets with B-splines as scaling functions do not exist, except for the trivial case of the Haar wavelet. Semi-orthogonal B-wavelets are the wavelets that are the closest to an orthogonal wavelet. Semi-orthogonal B-wavelets are wavelets with B-splines as scaling functions. Semi-orthogonality means that wavelets of different resolutions are orthogonal to eachother.

Definition: Semi-orthogonal wavelet

A wavelet ψ is called a semi-orthogonal wavelet if the basis $\{\psi_{m,n}\}$ satisfies

$$< \psi_{m,n}, \psi_{m',n'} >= 0, m \neq m' \qquad \textbf{(3.5)}$$

Contrarily to orthogonal wavelets, translated versions of wavelets of a given resolution are not always orthogonal. Orthogonality is only given between wavelets of different resolutions (fig. 3.7). Practically, the non-orthogonality has as a consequence that the decomposition and reconstruction filters are different.

The filters associated to the semi-orthogonal B-spline constructions are not finite. Great care must be therefore taken with boundaries when a good description of the end points is desired. For instance, the data may be folded about the end points. Semi-orthogonal wavelets are therefore not well suited to on-line learning. The next table gives the first filter coefficients corresponding to the semi-orthogonal wavelets of order 2 and 4, associated to the linear B-spline and the cubic B-spline (Chui, 1992).

	Order 2		Order 4	
j	p_j	q_{j+1}	p_{j+1}	q_{j+4}
1	0.683012701	0.866025403	0.893162856	-1.475394519
2	0.316987298	-0.316987298	0.400680825	0.468422596
3	-0.116025403	-0.232050807	-0.282211870	0.742097698
4	-0.084936490	0.084936490	-0.232924626	-0.345770890
5	0.031088913	0.062177826	0.129083571	-0.389745580
6	0.022758664	-0.022758664	0.126457446	0.196794277
7	-0.008330249	-0.016660498	-0.066420837	0.207690838
8	-0.006098165	0.006098165	-0.067903608	-0.106775803
9	0.002232083	0.004464167	0.035226101	-0.111058440
10	0.001633998	-0.001633998	0.036373586	0.057330952
11	-0.000598084	-0.001196169	-0.018815686	0.059433388
12	-0.000437828	0.000437828	-0.019473269	-0.030709700
13	0.000160256	0.000320512	0.010066747	-0.031811811
14	0.000117315	-0.000117315	0.010424052	0.016440944

	Order 2		Order 4	
j	g_j	h_{j+1}	g_{j+1}	h_{j+4}
1	1	5/6	0.75	-24264/8!
2	0.5	-0.5	0.5	18482/8!
3		1/12	1/8	-7904/8!
4				1677/8!
5				-124/8!
6				1/8!

Depending on the order of the B-wavelets, the filters associated to the wavelet decomposition and reconstruction algorithms have different properties. At order zero, the B-wavelet corresponds to the Haar wavelet, while at high order, the wavelet almost matches gaussian function. For odd order and for $m \geq 3$, the wavelet ψ^m_b can be approximated by $\sin(\omega t)\, g(t-b)$ with $g(t-b)$ the gaussian function. For even order ψ^m_b is almost of the form $\cos(\omega t)\, g(t-b)$. This means that B-wavelet of increasing order approach the limit set by Heisenberg to the product of the time and frequency resolution of a function.

Heisenberg uncertainty principle states that ($\Delta\omega\, \Delta t \geq ½$). The equality holds for a function f(t) of the form: $f(t) = a \cdot e^{i\cdot\alpha\cdot t} \cdot e^{-b(t-u)^2}$. It follows that the time-frequency resolution of B-wavelets tends to the inferior limit of ½ with increasing order.

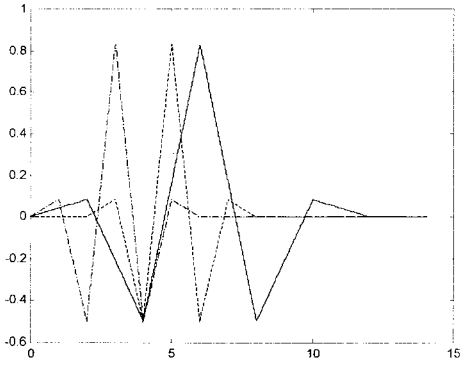
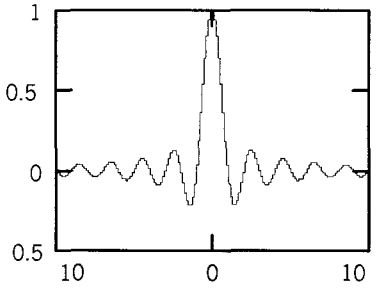


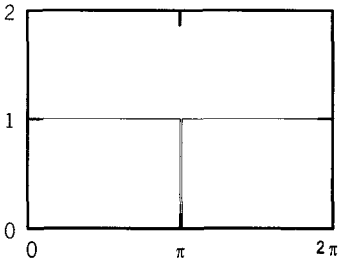
Figure 3.7: Semi-orthogonal wavelets fulfill the condition $< \psi_{m,n}, \psi_{m',n'} >= 0$ for $m \neq m'$. Illustration with semi-orthogonal second order B-spline wavelets. The two-times dilated wavelets (solid line) are orthogonal to the two other wavelets.

Battle-Lemarié wavelets

We have mentioned the impossibility of constructing orthogonal B-wavelets with a B-spline as scaling function. Nevertheless, orthogonal wavelets based on B-splines may be designed with an orthogonalization procedure. The resulting orthogonal wavelets are called Battle-Lemarié wavelets. The Battle-Lemarié wavelets (Battle, 1987; Lemarié, 1988) are piecewise polynomials, but their scaling function is not positive everywhere, a main problem to a linguistic interpretation of the results of a decomposition. The properties of the Battle-Lemarié wavelets depend on the chosen order. The two extremes N=0 and N→∞ correspond to two limit cases for filters. The scaling function of the first order case corresponds to the characteristic function: a perfect spatial filter. For large N, the scaling function tends to the sinc function, the perfect low-pass filter (fig. 3.8). By choosing the order of the Battle-Lemarié wavelet, one chooses simultaneously the type of filter. For this reason, Battle-Lemarié wavelets are good candidate functions to design a fuzzy controller in the frequency domain (see part 2).



Sinc function



Fourier Transform of the sinc

Fourier Transform of the sinc wavelet

Figure 3.8: The sinc wavelet is the equivalent of the Haar wavelet for the frequency domain.

A selection of wavelet-based algorithms for spline approximation

A function f(x) can be approximated as a weighted sum of wavelets

$$f(x) = \sum_{n,m} d'_{m,n} \cdot \psi_{m,n}(x) + \sum_{n,m_0} c'_{n,m_0} \cdot \phi_{m_0,n}(x) \qquad (3.6)$$

or equivalently as a weighted sum of scaling functions

$$\hat{f}(x) = \sum_{n,m} \hat{c}_{m,.n} \cdot \phi_{m,n} \qquad (3.7)$$

Depending on the available computing power and memory, different methods can be chosen to determine the values of the coefficients. The two next sections present thresholding techniques, while the last section describes an adaptation of the matching pursuit algorithm to splines.

In regard to the computing power, the least demanding wavelet-based method is thresholding. The approach is essentially identical to thresholding in data compression (part 1). It uses a central property of orthogonal wavelets, namely that the energy contained in the wavelet coefficients and the last level of approximation coefficients sum up to the total signal energy. In the thresholding method, the coefficients with a squared value above a given threshold are kept. This is equivalent to setting the coefficients below the threshold to zero. The energy contained in the reconstructed signal compared to the total signal energy is a measure of the quality of the approximation. A slightly different approach consists in keeping the K largest coefficients. For semi-orthogonal wavelets, in which orthogonality holds only between wavelets of different resolutions, the thresholding is still applicable. The thresholding algorithm gives also good results with some biorthogonal wavelets, such as biorthogonal splines. Biorthogonal spline-wavelets are not orthogonal, nevertheless the orthogonality relation holds to a sufficient degree and the thresholding method can also be applied.

The thresholding method can also be implemented to decompose the signal as a sum of scaling coefficients. If the wavelet and scaling functions do have a compact support, each wavelet can be decomposed with the two-scales relation as a finite sum of scaling functions at one higher level of resolution.

If some more computing power is available, the functions on which to decompose the signal can be chosen from a dictionary of functions. The best basis and the matching pursuit algorithm are the two standard methods in those cases.

Thresholding

Thresholding is a simple wavelet-based method to compress information. It can be used to find an approximate description of a function f(x) with a limited

number of terms. The filters corresponding to a decomposition on an orthogonal basis do fulfill the power complementarity condition. The power complementarity condition implies energy conservation. For a given level of decomposition the energy conservation is expressed by the relation:

$$\sum_{n} c^2_{m,n} = \sum_{n'} c^2_{m-1,n'} + d^2_{m-1,n'} \qquad (3.8)$$

For a complete decomposition, the energy conservation becomes:

$$\sum_{n} f^2(x_n) = \sum_{n} d^2_{m,n} = \sum_{n} d^2_{m-1,n} + d^2_{m-2,n} + \ldots + d^2_{0,n} + c^2_{0,n} \qquad (3.9)$$

The thresholding method consists of setting to zero all coefficients below a given threshold (fig. 3.9). A variant of the thresholding method can be used if the number of coefficients K is predefined. In this case, the function f(x) is reconstructed from the K coefficients among $d_{m,n}$, $c_{0,n}$ containing the most energy. The compression factor is given by the difference between the number of bits necessary to store the original signal and the memory capacity to store and address the K coefficients. The error $Er(\hat{f})$ on the reconstruction can be quantified by the relative difference between the energy contained in the function f(x) and its estimate $\hat{f}(x)$ given by an expression of the form:

$$\hat{f}(x) = \sum_{n,m} d'_{m,n} \cdot \psi_{m,n}(x) + \sum_{n,m_0} c'_{m_0,n} \cdot \phi_{m_0,n} \qquad (3.10)$$

$$Er(\hat{f}) = \sum_{n} (f^2(x_n) - \hat{f}^2(x_n)) / \sum_{n} (f^2(x_n)) \qquad (3.11)$$

$$d'_{m,n} = 0 \quad \text{if} \quad d^2_{m,n} \le T$$

$$d'_{m,n} = 0 \quad \text{if} \quad d^2_{m,n} > T$$

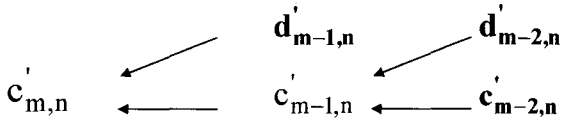


Figure 3.9: The thresholding method consists of keeping the largest coefficients among the wavelet coefficients and the approximation coefficients at the lowest level of resolution (i.e. bold coefficients) to compress the information. Illustration of the algorithm for a two levels decomposition tree. The coefficients c' are computed from the coefficient d'. The coefficients $c'_{m,n}$ approximate the function f(x).

This algorithm is optimal for orthogonal wavelets. For semi-orthogonal constructions, the algorithm can be also generally applied. The energy is only partially conserved in a semi-orthogonal decomposition. To see this, let us express the norm of the function f as a function of the detail coefficients. For a simplification of the formalism, let us assume a full decomposition of a function f of zero average.

$$< f, f > \; = \; < \sum_{m,n} d_{m,n} \cdot \psi_{m,n} , \sum_{m',n'} d_{m,n} \cdot \psi_{m',n'} > \qquad \textbf{(3.12)}$$

For semi-orthogonal wavelets, (3.12) can be put under the form:

$$< f, f >= \sum_{m} (\sum_{n} d_{m,n}^2 + \sum_{n \neq n'} d_{m,n} \cdot d_{m,n'} \cdot < \psi_{m,n} , \psi_{m,n'} >) \qquad \textbf{(3.13)}$$

The energy conservation is in general not fulfilled. There are two special cases in which the energy conservation is fulfilled to a good degree. For the energy conservation to hold to a good degree, the last term in (3.13) must be very small. This is the case if

a) the function f(x) can be reasonably described as a realization of a white noise signal. In this case, the coefficients $d_{m,n}$ are uncorrelated.

b) $< \psi_{m,n} , \psi_{m,n'} >$ is small for $n \neq n'$.

In summary, for semi-orthogonal splines, the sum of the squared detail coefficients is generally not equal to the total energy contained in the signal. Nevertheless, the energy contained in the signal at the different levels of resolution sum up to the total energy. Keeping the largest coefficients is therefore a good strategy, that furnishes good results.

For some biorthogonal wavelets, the thresholding algorithm can also be implemented. This is in particular the case of several biorthogonal spline-wavelets. In biorthogonal wavelets, the values of the frame bounds gives a good indication whether it is reasonable to use the thresholding algorithm. In order to see why, let us introduce the notion of a frame.

Definition:

An ensemble of functions (or vectors) $\{\theta_n\}$ with n an index is a frame of an Hilbert space **H** if there exists two constants A>0, B>0 such as for any $f \in \mathbf{H}$:

$$A\|f\|^2 \; \leq \; \sum_{n} |< f, \theta >|^2 \leq \; B\|f\|^2$$

It can be shown that biorthogonal wavelets form a frame. This follows directly from the fact that biorthogonal wavelets form a Riesz basis (see part 1). The expression (B/A)-1 can be used as an measure on how far is a basis from being orthogonal. Orthogonal wavelets are tight frames meaning that A = B = 1 .

Biorthogonal splines are to a reasonable approximation tight frames. In biorthogonal spline wavelets, the energy conservation holds in very first approximation.

Thresholding adapted to the decomposition with scaling functions

We will address the problem of finding a good representation of a function f(x) as a sum of scaling functions of different resolutions. More precisely, one searches for an approximation of f(x) in terms of the scaling functions associated to a dyadic wavelet decomposition: $f(x) = \sum_{m=0..J} \sum_{n} c'_{m,n} \cdot \phi_{m,n}(x)$ with $\phi_{m,n}(x) = 2^{m/2} \cdot \phi(2^m \cdot x - n)$ (m,n are integer).

The motivation behind this problem will become clear in the next chapters, as we will use splines as scaling functions and these scaling functions will be interpreted as membership functions in a fuzzy framework. The problem of decomposing a function f(x) as a sum of scaling functions of same resolution may be solved by a least mean-squares method. The least mean-squares method is quite computer-intensive and in many problems a neural network approach is implemented. The complexity of the problem increases if scaling functions of different resolutions are used, so that least mean-square methods become rapidly practically intractable. Also the linear dependence existing between scaling functions at different resolutions is computationally often problematic. In those cases, the problem can be tackled with a variant of the wavelet thresholding technique or with a matching pursuit algorithm.

By definition of a multiresolution, any wavelet can be expressed as a linear sum of the scaling function:

$$\psi(x) = \sum_{k} h_n \cdot \phi(2 \cdot x - n) \qquad \textbf{(3.14)}$$

This equation together with a similar relation for the scaling function are called the two-scales relations. As an example, the second order B-wavelet in fig. 3.10 is decomposed as a sum of scaling functions at one higher level of resolution. It follows that both the wavelet thresholding method and the matching pursuit may be used to search for a good decomposition in terms of scaling functions. In a first step, the function f(x) is decomposed as a sum of wavelets:

$$f(x) = \sum_{m=0..M} \sum_{n} d'_{m,n} \cdot \psi_{m,n}(x) + c_{0,n} \cdot \phi_{0,n} \qquad \textbf{(3.15)}$$

In a second step, each wavelet is expressed as a sum of scaling functions using the two-scales relation:

$$f(x) = \sum_{m=0..M} \sum_{n} c'_{m,n} \cdot \phi_{m,n}(x) \qquad \textbf{(3.16)}$$

The necessary computing power to determine the coefficients $c'_{m,n}$ is generally much smaller than a least mean-squares approach that requires dealing with large matrices.

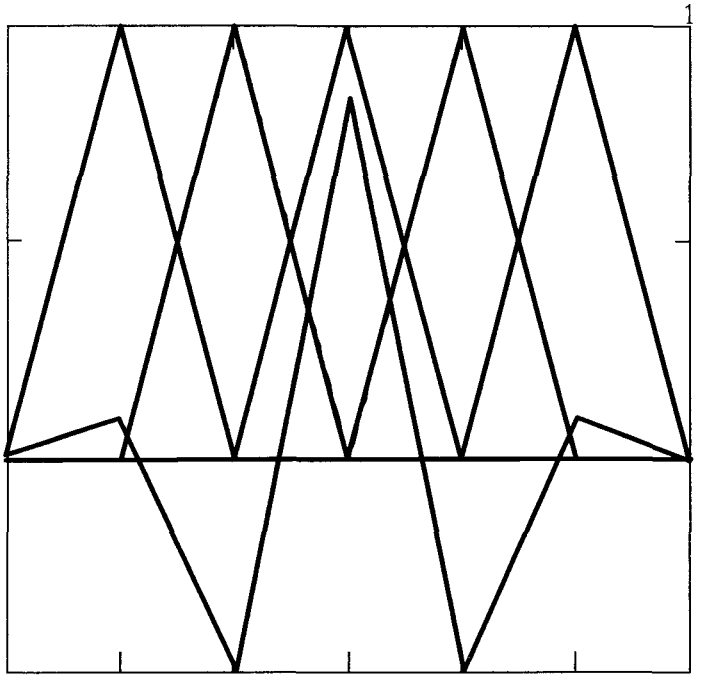


Figure 3.10: A wavelet can be decomposed into a sum of translated scaling functions. For the second order spline, the coefficients are (1/12,-0.5,5/6,-0.5,1/12).

An alternative and more efficient method is to transform first the detail coefficients with the reconstruction algorithm and to keep the largest coefficients expressed in terms of the scaling functions. First the detail coefficients are expressed in terms of the scaling function using the reconstruction algorithm: $c_{m,n} = \sum_k g_{n-2k} \cdot c_{m-1,k} + h_{n-2k} \cdot d_{m-1,k}$. Rewriting the equation as the sum of a low-frequency $c_{ml,n}$ and a high frequency contribution $c_{mh,n}$

$$c_{m,n} = c_{ml,h} + c_{mh,n} \tag{3.17}$$

one obtains

$$c_{mh,n} = \sum_k h_{n-2k} \cdot d_{m-1,k} \tag{3.18}$$

The problem of finding a good description of a function f(x) in terms of the scaling functions can be solved by using the largest coefficients among the reconstructed coefficients $c_{mh,n}$ and the lowest level approximation coefficients. The coefficients $c_{mh,n}$ correspond to coefficients of the scaling function. This procedure is illustrated in fig. 3.11.

$$c'_{mh,n}=0 \quad \text{if} \quad c^2_{mh,n} < T$$

$$c'_{mh,n}=c_{mh,n} \quad \text{if} \quad c^2_{mh,n} \geq T$$

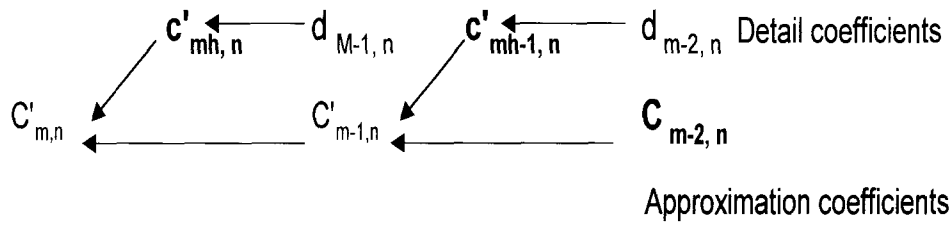


Figure 3.11: The algorithm to determine the best fuzzy rules consists of keeping the largest coefficients among the wavelet coefficients expressed in terms of the scaling functions and the approximation coefficients at the lowest level of resolution (bold coefficients).

In the spirit of regularization theory, a level-dependant multiplicative factor can be also used.

Matching pursuit with scaling functions

Mallat and Zhang (1993) have designed a very powerful matching pursuit algorithm that is well adapted to finding a good wavelet decomposition in terms of a small number of coefficients. The algorithm does also work with scaling functions (Shmilovici, 1996,1997; Thuillard, 1997), though for splines a modified matching pursuit algorithm is preferable (Thuillard, 1998a,c, 2000a). Figure 3.12 shows the basic idea of the modified matching pursuit algorithm, that is described below.

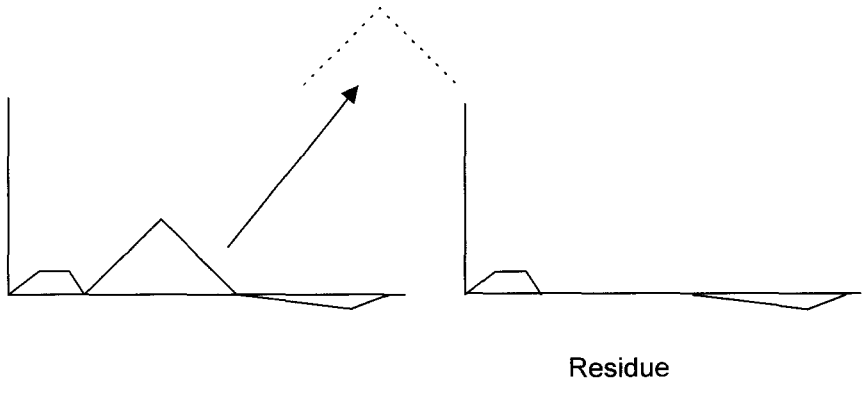


Figure 3.12: The best membership functions and rules to describe a set of data are determined with a matching pursuit algorithm.

Description of the algorithm

Define a dictionary $\boldsymbol{D}=\{\phi^k_{m,n}\}$ of scaling functions with $\phi^k_{m,n}=2^m\cdot\phi^k(2^m\cdot x-n)$. The index k indexing the order of the scaling function, m the dilation and n the translation (The normalization factor is contrarely to previous sections 2^m).

For each scaling function in the dictionary, decompose the datafile with the fast wavelet decomposition

algorithm.

- Keep for each k, the approximation coefficient $c^k_{m,n}$ with the largest m such as $\left|c^k_{m,n}\right| > \beta\ \sup_{m',n'}\left|c^k_{m',n'}\right|$ with $0<\beta\leq 1$.
- Choose the coefficient that minimizes the residue (i.e. write $f(x) = c^k_{m,n}\cdot\phi^k_{m,n}(x)+R(x)$ and choose the coefficient that minimizes $<R(x),R(x)>$).
- Take the residue as new input file.

Repeat the procedure till the residue is below a given value.

The algorithm is essentially the same as the wavelets' matching pursuit except for the supplementary condition that the coefficient with the smaller resolution is kept. The main idea behind this modification is the following. First, the condition $\left|c^k_{m,n}\right| > \beta\ \sup_{m',n'}\left|c^k_{m',n'}\right|$ with $0<\beta\leq 1$ ensures the convergence of the matching pursuit. As shown by Mallat (1993), the convergence rate is related to the value β. Roughly, the smaller the β, the slower is the convergence rate. On

the other hand, the supplementary step in the algorithm, requiring to keep the coefficient with the smaller resolution fulfilling the above condition, permits very often to discover the most appropriate resolution to compress the signal. Figure 3.13 illustrates the algorithm with a simple example: the decomposition of a second order spline function with a semi-orthogonal spline construction. Using a value of β=0.7 in $\left|c^k_{m,n}\right| > \beta \sup_{m',n'}\left|c^k_{m',n'}\right|$ restricts the best matching coefficients to the bald coefficients. In Mallat' s algorithm, one could have chosen any of them. The second condition prescribes to chose among them the coefficient corresponding to the scaling function with the lowest resolution. The chosen coefficient is underlined.

As long as the value of β is taken larger than 0.68, the algorithm furnishes the best matching function after a single iteration of the algorithm. It permits to catch the last level of resolution before the large decrease of the values of the approximation coefficient corresponding to the decomposition of the unit impulse. A smaller β value would not have permitted to discover the right resolution. On the other hand, a large value of β close to one is also not desirable as the slightest noise or some small deviation to the spline function may lead to choosing a suboptimal solution. For an a priori unknown function, a value about 0.9 is recommended for the modified matching pursuit with splines of order up to 3.

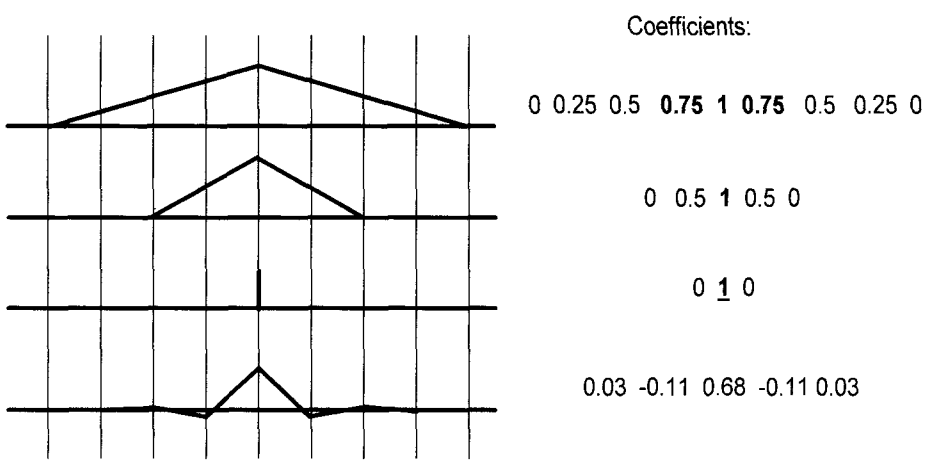


Figure 3.13: Illustration of the search algorithm with a modified matching pursuit algorithm. The algorithm is so modified to discover in most cases the best resolution to describe locally the dataset.

PART IV AUTOMATIC GENERATION OF A FUZZY SYSTEM WITH WAVELET-BASED METHODS

This page is intentionally left blank

4. Automatic Generation of a Fuzzy System with Wavelet-Based Methods

Fuzzy rule-based systems have found numerous applications in many different fields. The two main fuzzy methods are Mamdani's min-max inference mechanism and the Takagi-Sugeno approach. Many variations of these two models have been proposed and applied with success. For instance the product can be used as AND operator. Also the defuzzification process can be carried out with many different methods, the center of gravity defuzzification and the fuzzy mean are the most popular.

The modeling of a surface with the singleton Takagi-Sugeno model using splines as membership functions is equivalent to a functional decomposition of the surface with splines. This permits to relate multiresolution analysis to the problem of learning from data in fuzzy logic. Using the algorithms presented in the previous chapter, a fuzzy description of the data can be obtained. In the fuzzy framework, spline scaling functions are interpreted as membership functions. Wavelet-based fuzzy approaches are characterized by a number of features:

-The support of the membership functions is chosen a priori. In the fuzzy-wavelet approach, the most appropriate membership functions are selected in a dictionary of scaling functions, comprising translated and dilated versions of so-called mother scaling functions.

-The multiresolution properties of the scaling functions permit to express any rule as a sum of rules using membership functions of higher resolutions. For this reason, it is always possible to express the resulting fuzzy system under a linguistically interpretable form.

-The singleton Takagi-Sugeno model can be put under the form of a model in which both input and output are fuzzified. It will be shown below that the two models are equivalent provided spline functions are taken to fuzzify the output space and a center of gravity defuzzification method is applied.

Fuzzy rule-based systems

Fuzzy logic has found applications in basically all domains in science, from biology to particle physics. The majority of applications are clearly in the domain of control. What are the reasons for the success of fuzzy logic? The linguistic interpretation of fuzzy rules is certainly one of the main reasons. The possibility of translating human expert knowledge formulated by an experienced practitioner

without a strong mathematical background into a fuzzy system has often been given as the main motivation behind fuzzy logic. Very often, the way around is at least as important. Fuzzy logic allows the development of transparent algorithms that can be explained to specialists, practitioners or even sometimes to customers. Another strong point for fuzzy logic is that it represents a simple method to describe nonlinearities. Finally fuzzy logic furnishes a theoretic framework to fuse information under different form and quality. The fusion of qualitative or even imprecise knowledge together with knowledge under the form of experimental data is quite feasible, though in real world often more difficult than one wants to admit. If different experts (human or machine) are contradicting, the process of reconciling the different experts is very often ad-hoc. New methods based on adaptive templates try to introduce some clear methodology into the process (see part 7).

The majority of applications uses fuzzy rule-based systems expressed under the form of if-then rules:

$$R_i: \quad \text{if} \quad \mathbf{x} \quad \text{is} \quad A_i \quad \text{then} \quad y \quad \text{is} \quad B \qquad \textbf{(4.1)}$$

Here A, B are linguistic terms, x is the input linguistic variable, while y is the output linguistic variable. The value of the input linguistic variable may be crisp or fuzzy. If the value of the input variable is a crisp number then the variable x is called a singleton. As an example, suppose that x is a linguistic variable for the temperature. The value of the input linguistic variable may be given by a crisp number such as *30 (°C)* or by *about 25* in which *about 25* is itself a fuzzy set.

The Takagi-Sugeno and the Mamdani models are probably the most popular approaches to rule-based fuzzy systems. Alternatives to these models include, among others, the linguistic equation approach, a method that has proven to be successful in a broad range of real world applications (Juuso, 1996, 1998; Leiviska, 1996).

Max-min method (Mamdani)

An important definition in fuzzy logic is that of a membership function. The membership function $\mu(\hat{z})$ to a fuzzy set z is defined by the mapping $\mu(\hat{z}): Z \longrightarrow [0,1]$, in which Z represents the domain of definition of the fuzzy set z.

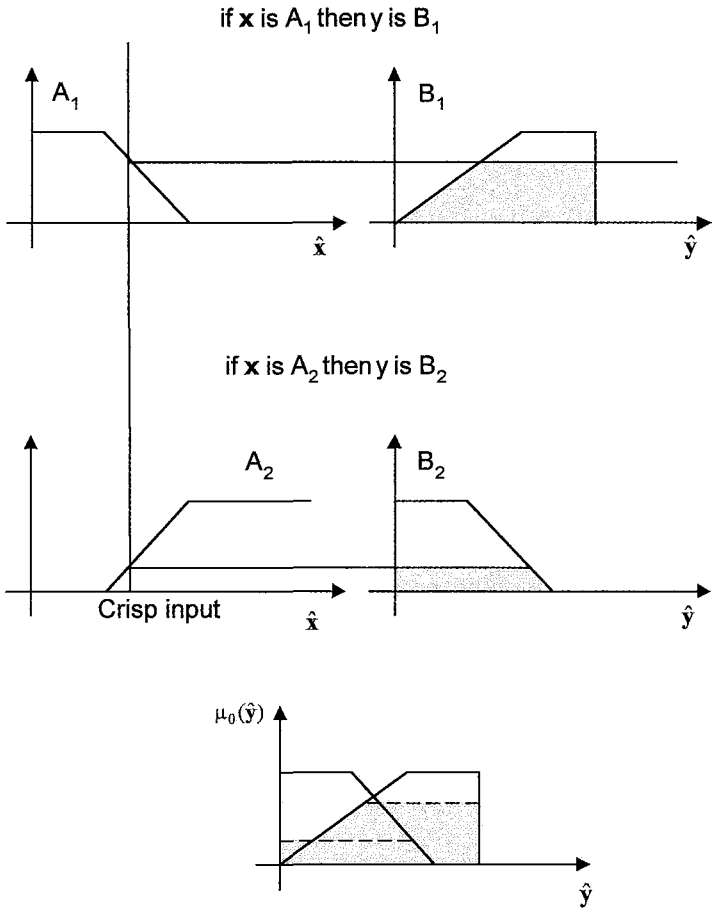


Figure 4.1: Illustration of Mamdani inference mechanism.

In Mamdani approach, the inference is computed with a 3 steps algorithm:

Step 1: Determine a set of fuzzy rules and membership functions.

R_i : if x is A_i then y is B_j

$\mu_{A_i}(\hat{x}) : X \longrightarrow [0,1]$

$\mu_{Bj}(\hat{y}) : Y \longrightarrow [0,1]$

Step 2: Compute the degree of fulfillment β_i of the inputs to the rule antecedents.

The membership function corresponding to the fuzzy input I is defined as

$\mu_I(\hat{x}) : X \longrightarrow [0,1]$

The degree of fulfillment is given by the expression:

$\beta_i = \max_X[\mu_I(\hat{x}) \wedge \mu_{A_i}(\hat{x})]$

with $\wedge$ the minimum operator (or the product).

Step 3: Derive the output fuzzy set $\mu_O(\hat{y})$.

The output fuzzy set $\mu_O(\hat{y})$ is obtained by aggregating the different output fuzzy sets:

$\mu_O(\hat{y}) = \max_{i,j}(\beta_i \wedge \mu_{B_j}(\hat{y}))$

The Mamdani type of fuzzy system is illustrated with two examples.

Figure 4.1 shows an example with a crisp input, while fig. 4.2 shows the algorithm for a fuzzy input.

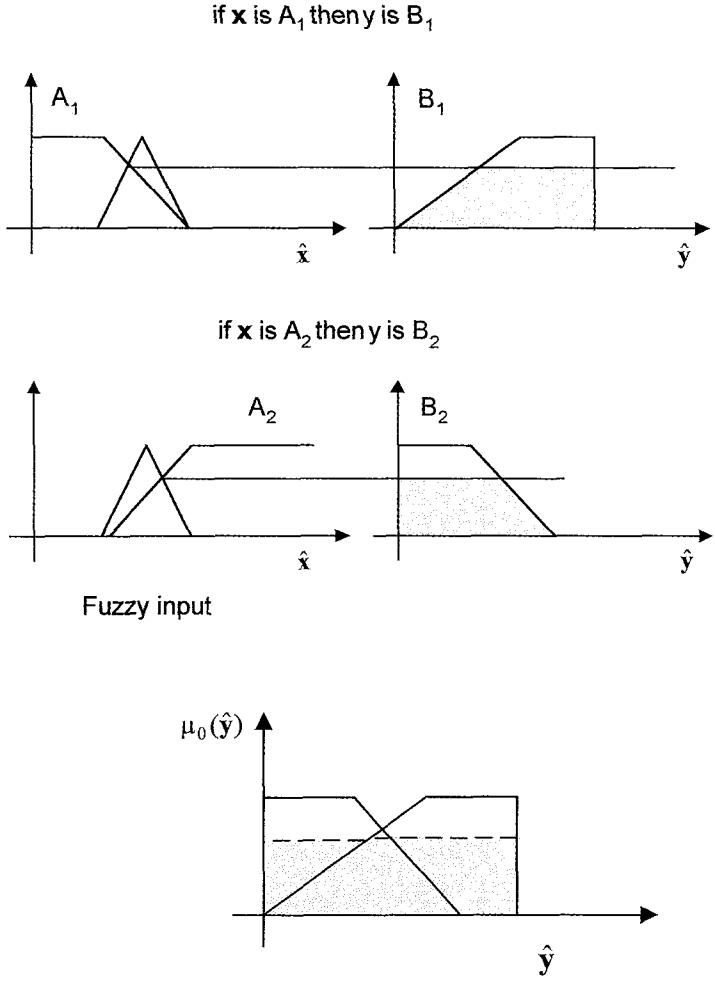


Figure 4.2: Illustration of Mamdani inference mechanism when the input is a fuzzy set.

In multivariate systems, the min operator is used for the conjunction AND.

Takagi-Sugeno model

In the Takagi-Sugeno method (Takagi, 1985) the fuzzy rules are expressed differently:

$$R_i: \quad \text{if} \quad \mathbf{x} \quad \text{is} \quad A_i \quad \text{then} \quad y = f_i(\mathbf{x}) \qquad \textbf{(4.2)}$$

Contrarily to Mamdani' s method, the output is a crisp number. The algorithm is slightly different (see fig. 4.3 for an example):

Step 1: Determine a set of fuzzy rules and membership functions.

$R_i: \quad \text{if} \quad \mathbf{x} \quad \text{is} \quad A_i \quad \text{then} \quad y = f_i(\mathbf{x})$

$\mu_{A_i}(\hat{\mathbf{x}}): X \longrightarrow [0,1]$

Step 2: Compute the degree of fulfillment β_i of the inputs to the rule antecedents.

The membership function corresponding to the fuzzy input I is defined as
$\mu_I(\hat{\mathbf{x}}): X \longrightarrow [0,1]$

The degree of fulfillment is given by the expression:
$\beta_i = \max_X [\mu_I(\hat{\mathbf{x}}) \wedge \mu_{A_i}(\hat{\mathbf{x}})]$
with $\wedge$ the minimum or the product operator.

Step 3: Derive the output.

$\hat{y} = \sum_i \beta_i \cdot f(\hat{\mathbf{x}}) / \sum_i \beta_i$

In many applications, a linear function is taken as a function $f(\hat{\mathbf{x}})$:

$$f(\hat{\mathbf{x}}) = \mathbf{a}^T{}_i \cdot \hat{\mathbf{x}} + b_i \qquad \textbf{(4.3)}$$

This model is quite attractive as the coefficients $\mathbf{a}_i, b_i$ can be computed by a least-squares method.

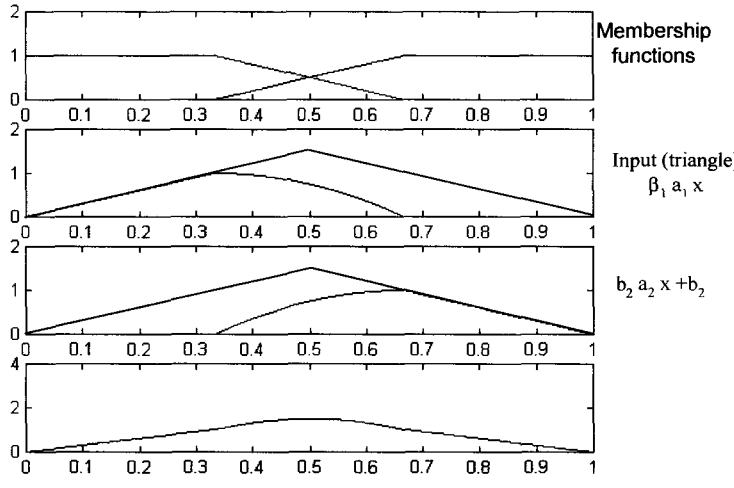


Figure 4.3: Illustration of Takagi-Sugeno inference mechanism. From above: membership functions, contribution of the first and second membership function with $a_1=-a_2=1$, output.

The singleton model

A constant b_i can be chosen to describe the crisp output y :

$$R_i: \quad \text{if} \quad x \quad \text{is} \quad A_i \quad \text{then} \quad y = b_i \qquad \textbf{(4.4)}$$

Figure 4.4 illustrates with an example the singleton model.

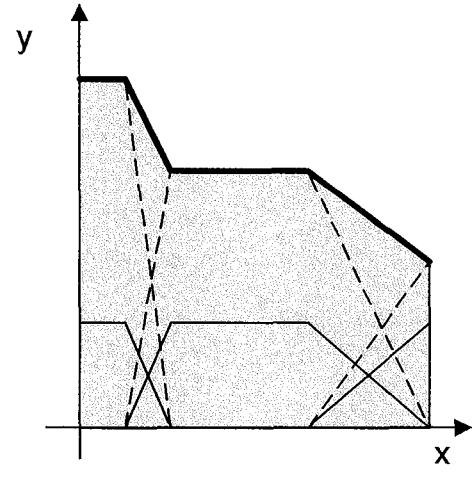


Figure 4.4: Illustration of Takagi-Sugeno inference mechanism for a singleton model.

Fuzzification of the output in a Takagi-Sugeno model

It can be shown that, in the setting of the singleton model, the Takagi-Sugeno model is equivalent to an ensemble of rules of the form:

$$R: \text{ if } x \text{ is } A \text{ then } y \text{ is } B \quad (C) \qquad \textbf{(4.5)}$$

provided a center of gravity defuzzification is applied and

$$\mu_{Bj}(x) = N^k(x-n) \qquad \textbf{(4.6)}$$

Let us consider a fuzzy system described by a set of rules of the form: $R_{i,j}$: if **x** is A_i then y is B_j ($C_{i,j}$) where A_i and B_j are linguistic variables and $C_{i,j}$ represent the confidence level. Further let take the product as AND operator and the addition to implement the fuzzy union.

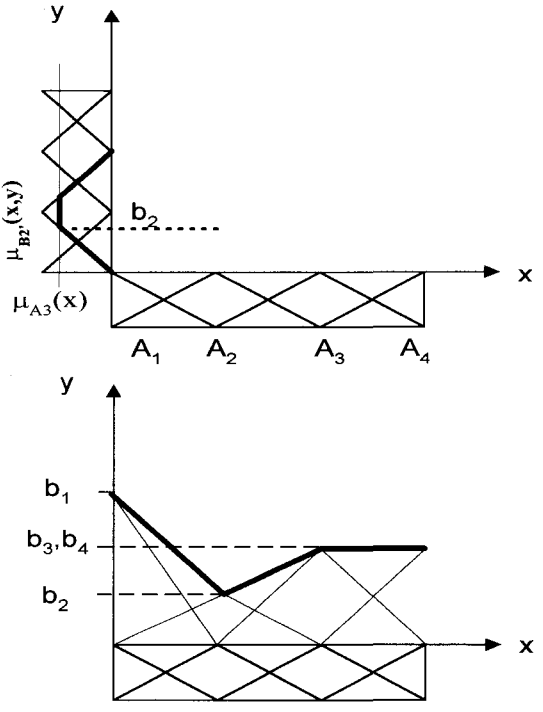


Figure 4.5: The Takagi-Sugeno fuzzy system can be transformed into a fuzzy system in which both input and output are fuzzified .

A remarkable property of spline functions can be used to compute the confidence levels starting from the weight b_i in a Takagi-Sugeno model. There

exists an invertible relationship between the confidence levels $C_{i,j}$ and the spline coefficients b_i. Assume a confidence level $C_{i,j}$ for the rule $R_{i,j}$ given by the following expression:

$$C_{i,j} = N^k(b_i - j) \tag{4.7}$$

in which $N^k(b_i - j)$ is a k^{th} order cardinal spline centered at j. In the fuzzy framework, N^k () can be interpreted as a membership function. In the fuzzy framework $N^k(b_i - j)$ corresponds to the degree of membership $\mu_{Bj}(b_i)$ to B_j.

Inversely the weight b_i can be computed from the different confidence levels $C_{i,j}$. It can be shown that the crisp output after defuzzification with a center of gravity defuzzification method gives exactly the value b_i.

$$b_i = \sum_j C_{i,j} \cdot y_j^c \tag{4.8}$$

with y_j^c the center of gravity of B_j or equivalently the position of the center knot of the B-spline function N^k. This relationship holds for all B-splines at any order. Figure 4.6 shows a graphical proof for second order B-spline functions. A complete proof can be found in Brown and Harris (1996).

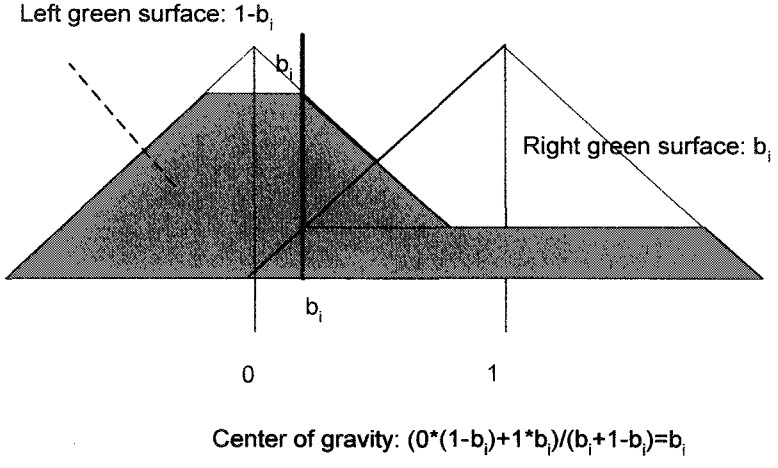


Figure 4.6: Graphical proof of the invertible relationship between the weight space and the confidence level space for the case of a center of gravity defuzzification and second order splines.

Neurofuzzy spline modeling

In the framework of the zero-order Takagi-Sugeno model, B-splines and fuzzy networks are equivalent. Quoting Brown and Harris (1996), *..the main difference between these techniques is the level of abstraction at which there are interpreted. B-splines are viewed as numerical processing or computational systems, whereas fuzzy networks can be given a linguistic interpretation as a fuzzy algorithm using terms such as small or large to label the basis functions.* A B-spline network estimates a function $f(x)$ as a weighted sum of B-splines forming a partition of unity:

$$\hat{f}(x) = \sum_j c_j \cdot \phi(x - x_j) \qquad \textbf{(4.9)}$$

The weights c_j may be computed by either an instantaneous gradient descent rule, iterative conjugate gradient or a least mean-squares method. Kalman filtering can be applied for state estimation and control (Gan, 1999). In a batch operation, the coefficients can be also directly computed from a singular-valued decomposition. B-splines are particularly well suited to constrained problems. B-splines of order k are piecewise continuous polynomials with (k-2) continuous derivatives. A main constraint in moving systems (robots, ship docking, automatic guidance) is the requirement that both the velocity and the acceleration are continuous. These continuity conditions are fulfilled by B-splines of order k ≥ 4. Cubic B-splines are therefore the lowest order spline fulfilling the continuity condition on the acceleration.

Extension of the model to first order Takagi-Sugeno types of models have been also designed (Harris, 1999a,b). Spline-based neurofuzzy methods have been implemented in a large number of research and development projects (Harris, 1999a), for instance in ship collision avoidance guidance (Harris, 1999b), helicopter guidance (Doyle, 1996), autonomous underwater vehicle (Bossley, 1997) or an intelligent driver warning system (An, 1996).

Fuzzy-wavelet

In this section, the equivalence between B-spline modeling and fuzzy modeling is extended to multiresolution fuzzy modeling. Wavelet-based fuzzy modeling is generally designed under the name *fuzzy-wavelet* (Thuillard, 1997). The equivalency between fuzzy modeling and wavelet-spline modeling has been recognized independently by different authors (Shmilovici, 1995,1996; Yu, 1996a,b,1999; Thuillard, 1997).

If x is a singleton, $\mu_{A_i}(x) = N^k(2^{-m} \cdot x - n)$ with $N^k(2^{-m} \cdot x - n)$ a k^{th} order cardinal B-spline function and the product operator is used for inference, then the system of eq.(4.9) is equivalent to

$$y(x) = \sum_{m,n} b_{m,n} \cdot N^k (2^{-m} \cdot x - n) \qquad \textbf{(4.10)}$$

In this particular case, the output y is a linear sum of translated cardinal B-splines. This means that under this last form the Takagi-Sugeno is equivalent to a multiresolution spline model. It follows that the wavelet-based techniques used to decompose a function as a weighted sum can be applied here.

The need for adapting the support of membership functions in learning has lead to the development of different neurofuzzy methods. The support of membership functions can be adaptively chosen, for instance by adding knots in spline networks. An important line of research is based on clustering methods (Babuska, 1998; Bedzek, 1981, Kosko, 1992), using neural networks or different variants of the fuzzy c-mean algorithm.

A main concern with neurofuzzy methods is to find the right balance between transparency, complexity and accuracy of the obtained fuzzy systems. Despite the fact that complexity has a number of definitions, it is generally possible to agree on a clear setting to discuss the complexity-accuracy issue. As soon as the notion of linguistic transparency is added, then opinions strongly diverge. A purely mathematical solution to that question is certainly not at hand, as linguistic transparency is a very human notion. *Keeping the man in the loop* is a central motivation for fuzzy logic and therefore a purely mathematical definition of linguistic transparency is not desirable. Linguistic transparency depends centrally on the level of education of the experts, as well as on their range of competence. A large number of neurofuzzy methods have been described as transparent or linguistically interpretable without much justification. It has been now understood that the transparency-complexity-accuracy issue is one of the most challenging question in fuzzy logic.

The fuzzy-wavelet approach implements the following strategy. A dictionary of membership functions forming a multiresolution is first defined. Each membership function defines a term, such as small or very small, that does not get modified during learning. The multiresolutional character of the dictionary makes rules fusion and splitting quite simple (fig. 4.7).

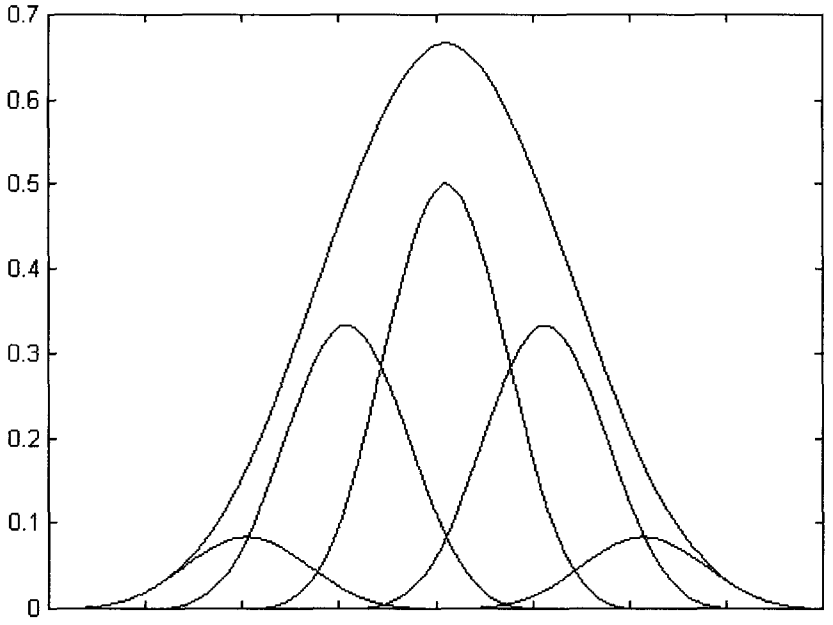


Figure 4.7: Illustration of the relation between scaling functions at different resolutions in a dyadic multiresolution analysis. The cubic spline function can be decomposed into the sum of translated cubic splines at the next higher level of resolution (coefficients are (1/8; ½; ¾; ½; 1/8)). The same holds for the corresponding wavelet.

General approach

Wavelet theory can be adapted to generate automatically fuzzy rules and confidence levels from a set of examples. This method, fuzzy-wavelet, takes advantage of the strong connection existing between a spline wavelet decomposition and a fuzzy system. Let us recall first, how to make a fast wavelet decomposition. A very efficient recursive algorithm, called the fast wavelet transform, carry out the computation of the wavelet transform. At each level of the transform, the data are processed through a low-pass and a high-pass filter. The high-pass filtered data $d_{m,n}$ are known as the detail wavelet coefficients. The result of the low-pass transform, the coefficients $c_{m,n}$ is used as input data to compute the next level of detail wavelet coefficients. Figure 4.8 describes symbolically the algorithm.

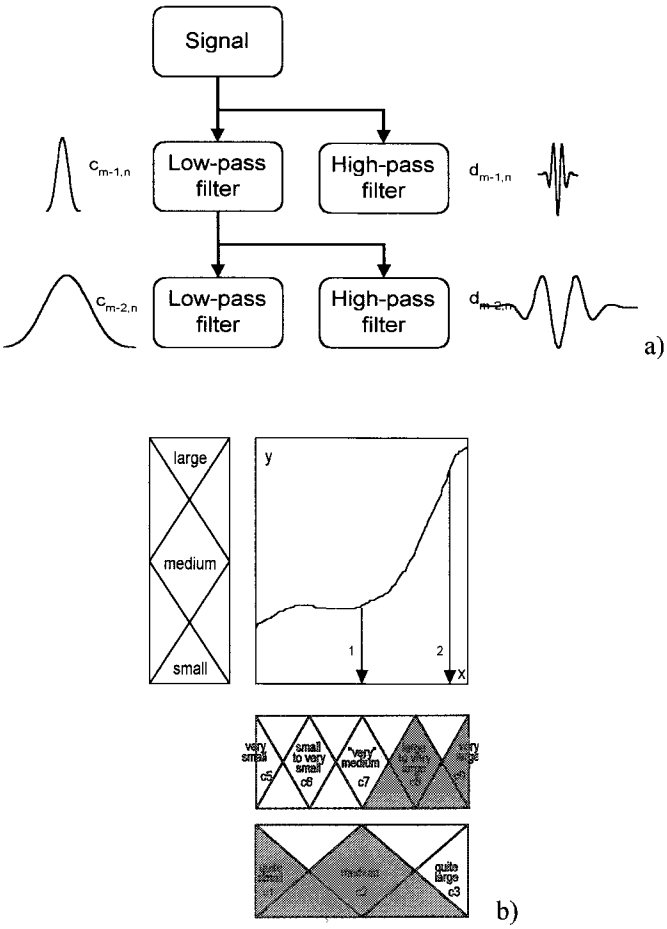


Figure 4.8: a) Example of a fast wavelet decomposition using B-wavelets. The low-pass filter corresponds to the projection on dilated and translated spline functions; b) Example of fuzzy rules using spline membership functions at several resolutions.

The connection between fuzzy logic and the fast wavelet algorithm is established by using B-wavelets. The main feature of B-wavelets is that the approximation coefficients $c_{m,n}$ represent the projections of the signal on spline functions. Spline functions are typical membership functions in fuzzy systems. The methods presented in part 3 can be therefore used to determine appropriate membership functions and fuzzy rules. Let us recall what these methods are:

-Thresholding

-Matching pursuit for splines.

A modified matching pursuit (explained below) permits to determine appropriate membership functions and rules to approximate a function in terms of a small number of fuzzy rules. The algorithm is a modified version of the matching pursuit algorithm that works specifically well with splines. The

resulting decomposition can be linguistically expressed with the zero order Takagi-Sugeno model with rules of the form

R_i : if **x** is A_i then $y_i = b_i$

which can be put, if necessary, under the form (4.6)

if **x** is A_i then y is $B_j(C_{i,j})$
using the equivalency between the two formulations.

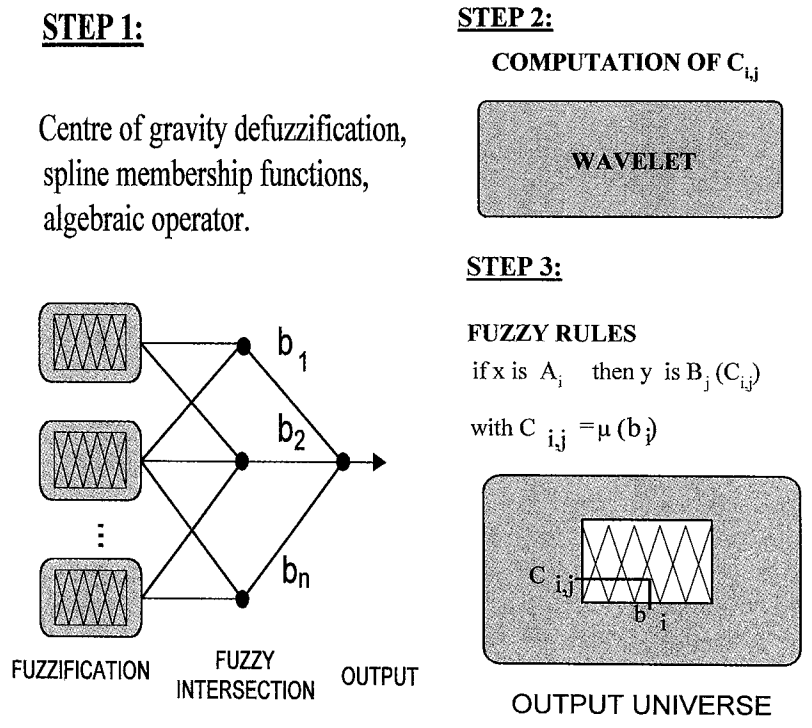


Figure 4.9: The similarities existing between a fuzzy description and B-wavelets are used to determine fuzzy rules and confidence levels describing a set of data.

Soft computing approach to fuzzy-wavelet transform

For a large multivariable dataset, the memory and computing requirements of the wavelet analysis may be too large. This problem can be solved in most cases by introducing a first approximation stage already during the wavelet decomposition. A Haar decomposition is used in the first stages: the first decomposition stages correspond to a simple averaging procedure. Spline-wavelets are introduced only at resolution levels containing most of the signal energy. The reconstruction algorithm uses spline-wavelets at all levels. This corresponds to using for the decomposition a scaling function approximating the

desired membership function and the exact scaling function for the reconstruction. Figure 4.10 shows an example of an approximated scaling function for a triangular membership function. The method is very efficient in dealing with multivariable datasets, even if the data are noisy.

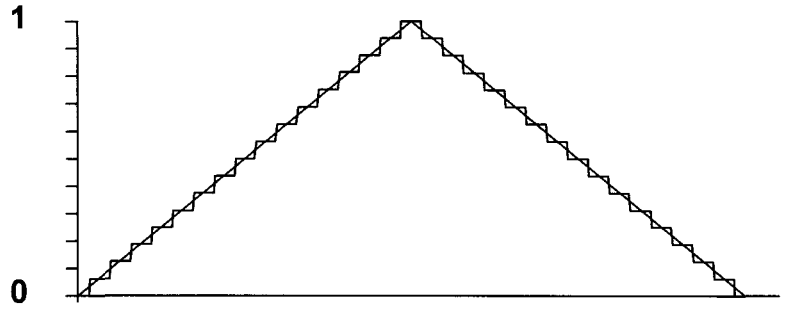


Figure 4.10: Example of a scaling function used for an approximated wavelet decomposition. The scaling function with steps is an approximation of the triangular membership function.

Processing boundaries

There are a number of methods to process boundaries, for instance by folding the data around the end points. If a precise linguistic interpretation of the results is required close to the boundary, then it is recommended to use second generation wavelets to process the end points. Second generation wavelets are wavelets that generalize the wavelet formalism to configurations that were not covered by the standard wavelet approach. In particular, spline-wavelets adapted to processing the end points have been designed with this technique. Depending on the support of the wavelet, a number of different functions must be used to process the boundary. For second order splines, a single wavelet is necessary. Figure 4.11 shows the scaling functions and the associated wavelets for the second order spline for processing the end points. The other points can be normally processed with a biorthogonal spline-wavelet decomposition.

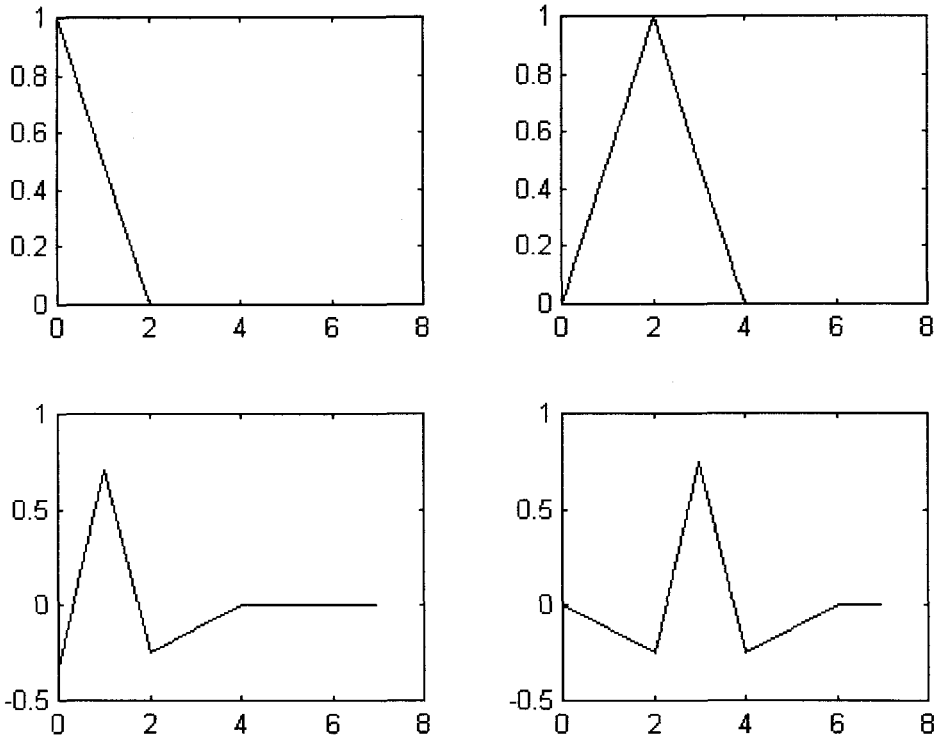


Figure 4.11: Boundaries can be processed with second generation wavelet using the lifting scheme. Right: wavelet associated to the scaling function on the left side. Left: scaling function for processing the last point with a second order spline.

Second generation wavelets have found a number of applications besides the one presented here. Second generation wavelets are used for multiresolution analysis on irregularly spaced grid, or to construct multiresolution on a sphere. We will encounter second generations wavelets again as extensions of the fuzzy-wavelet formalism will be discussed. The reader is referred to the annex for details on the construction of second generation wavelets.

Linguistic interpretation of the rules

The question of the adequacy of the linguistic formulation is especially important, as one is dealing with human experts. If only translated of a single spline are used, then generally through a simple rescaling, the rules can be put under a linguistic form that can be processed by the human expert. For membership functions that are chosen adaptively, the problem of interpretability becomes central. The lack of clear interpretability of many fuzzy systems generated with a neurofuzzy approach is a major drawback. The fuzzy-wavelet approach overcomes this problem quite elegantly by using a multiresolution.

Indeed, a scaling function at a given level of resolution can be expressed as the sum of higher resolution scaling functions by using the two-scales relation which is at the heart of the whole wavelet framework (Part 1):

$$\phi(x) = \sum_k g_k \cdot \phi(2 \cdot x - k) \qquad \textbf{(4.11)}$$

This means that the membership functions can be fused together or split into membership functions at a higher resolution quite easily. A nontrivial example showing two approaches to make the results easily linguistically interpretable will now be presented. The first approach is illustrated in fig. 4.12.

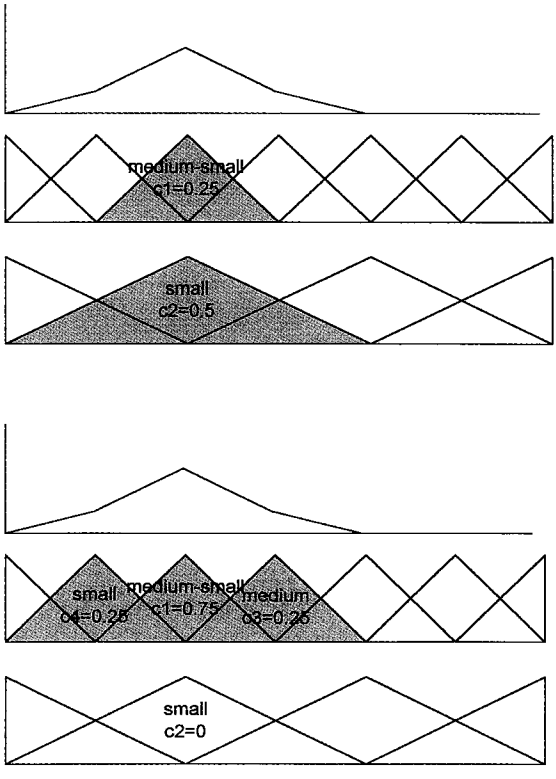


Figure 4.12: The above curve can be expressed differently as a sum of scaling functions. The left decomposition is more compact for implementation, but the right form is linguistically better for a human expert.

The function in fig. 4.12 corresponds to the superposition of two scaling functions. The support of the second scaling function is contained into the support of the first scaling function. The function can be decomposed into the sum of the scaling function corresponding to medium-small and small. In this

example, the function is constructed on purpose such as a linguistic interpretation of the decomposition is not straightforward to a human expert. The degree of membership to *medium-small* is smaller than the degree of membership to small, in spite of the fact that the function has a peak within the medium-small range. For a human expert, such rules are counter-intuitive. In the present case, one may have rules such as *if x is medium-small then y is small; if x is small then y is large.* The two rules are apparently contradicting. So how can this problem be solved? A simple approach consists of splitting the scaling function corresponding to *small* into the sum of the scaling functions at the higher level of resolution.

After splitting the scaling function, the results of the wavelet decomposition can be expressed as linguistically correct fuzzy rules. The degree of membership to medium-small is the largest and the result is also understandable linguistically. The main disadvantage of this representation is that it may be too precise for the human expert. If this is the case, another approach consists of transforming the description into a layered structure as shown in fig. 4.13. This second approach is quite efficient, if the number of levels of resolution is not too large

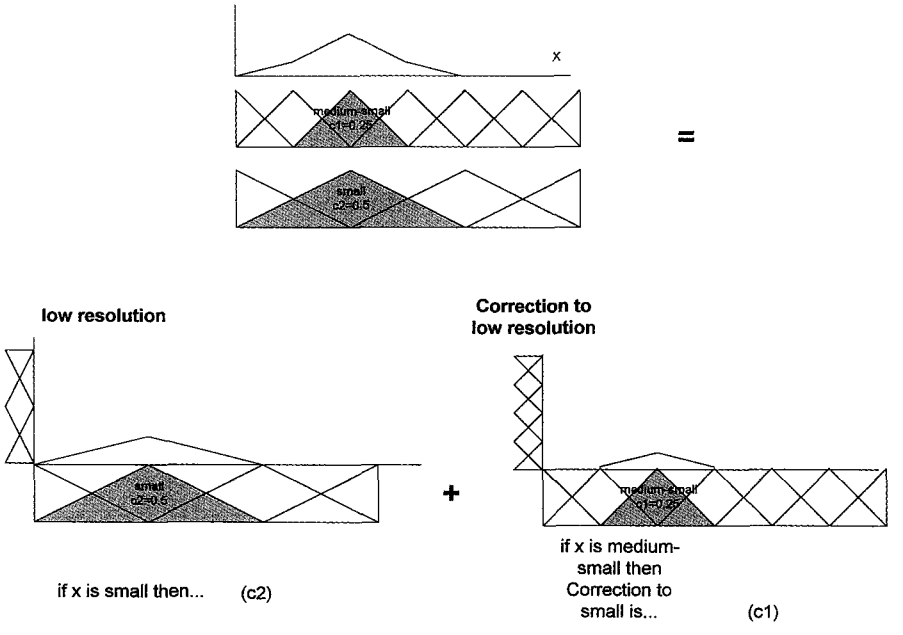


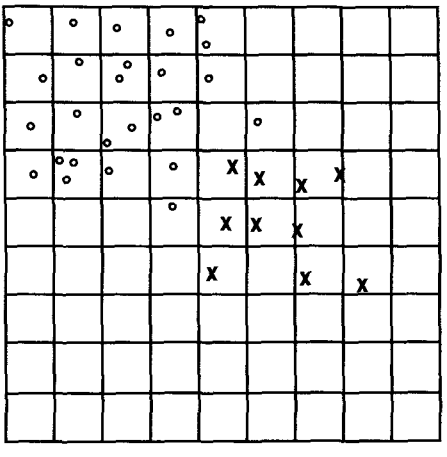
Figure 4.13: The curve above can be decomposed as the sum of a low resolution description and a correction to this description. This representation is a good compromise between compactness of the representation and linguistic clarity.

Fuzzy-wavelet classifier

We have shown above that wavelet theory and fuzzy logic can be combined into a single method. This opens the possibility to develop a fuzzy-wavelet classifier, that is a wavelet classifier with a linguistic interpretation of the classification. Wavelet-based classifiers have found applications in different fields, going from the analysis of cracks to the analysis of seismic data. The basic idea consists of analyzing first the signal with a wavelet decomposition. The coefficients of the wavelet decomposition are then compared with examples in an identification stage. The development of the classifier may use very different techniques going from look-up tables to Learning Vector Quantization (LVQ), Kohonen networks, decision trees or genetic algorithms.

We present here an example of a fuzzy-wavelet classifier. The classifier has been tested successfully on an industrial project: the development of algorithms, integrated into a fire detector, capable of making automatically the distinction between a signal caused by deceiving phenomena and real fires.

The first stage of the algorithm consists of choosing a mesh size h for the observation data. The input space is divided into small hyperboxes H_l of volume h^3. The database containing the examples is also divided into two subsets A and B. Examples for A and B are *deceiving phenomena* or *fire* in the example of a classifier implemented into a fire detector. A value is attributed to each hyperbox. The value 1/2 is given if no example lies within the hyperbox. The value 1 is attributed, if all the examples in the hyperbox belong to the subset A and 0 if they belong to the subset B. If elements of both subsets are found in an hyperbox then either the mesh size is reduced or new definitions for A and B are chosen (for instance A+B=fire).



0	0	0	0	0	0.5	0.5	0.5	0.5
0	0	0	0	0	0.5	0.5	0.5	0.5
0	0	0	0	0.5	0	0.5	0.5	0.5
0	0	0	0	1	1	1	0.5	0.5
0.5	0.5	0.5	0	1	1	1	0.5	0.5
0.5	0.5	0.5	0.5	1	0.5	1	1	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

Figure 4.14: The first stage of the classification corresponds to the coding of the examples after dividing the input space into a number of hyperboxes. Illustration with two input variables. The matrix (right) is used as input to the wavelet decomposition.

As an illustration, let us assume that there are only two input variables. Figure 4.14 shows an example of the input matrix S after coding of the database containing the input data. The classification stage is carried out by making a wavelet analysis of the matrix S using spline-wavelets. The matrix rows are processed first, and then the columns. To each decomposition level correspond three matrices. The first two matrices contain the detail coefficients as elements, while the third matrix corresponds to the result of the low-pass filtering. An interpretation of the coefficients as a function of their values can be done. Let us recall that the fast wavelet decomposition algorithm is carried out with a cascade of filters. At a given level, the detail coefficients represent the high-frequency component of the signal, while the approximation coefficients give the low-frequency part of the signal. The detail coefficients, corresponding to the high-frequency part of the signal, can be used therefore to characterize the boundaries between two domains (recall that the wavelet coefficients are good edge detectors!). The approximation coefficients $c_{m,n}$ give also important information. The approximation coefficients $c_{m,n}$ correspond to the projection of the signal on the scaling function. The scaling functions can be interpreted as membership functions, and the coefficients can be used to compute the confidence levels of the rules defined implicitly by $c_{m,n}$ (Thuillard, 1997b) An approximation coefficient $c_{m,n} = 1$, means that the hyperbox H_l, corresponding to the support of the bounded multivariate spline-wavelet used for the projection, contains only examples corresponding to the subset A. Undefined hyperboxes contribute to reducing the degree of membership.

Off-line learning from irregularly spaced data

In this section, we will discuss the possibility of using fuzzy-wavelet techniques after some nonlinear preprocessing of the input data. First, the datapoints are mapped bijectively onto a regular grid. Then the wavelet decomposition is carried out on the regular grid. Finally, the resulting approximation in terms of splines is mapped back to the original grid. Let us examine how to carry out that program with a very simple example (fig. 4.15). The input space is two-dimensional and contains 16 points. The 16 datapoints are mapped bijectively onto a regular 4x4 matrix. The 4 points with the largest values of x_2 are associated to the first row. After removing these 4 points, the procedure is repeated for the second row. An example with a two-dimensional input space was chosen to illustrate that the mapping does not preserve near-neighbors relationships. A mapping preserving near neighbors relationships is generally computationally very demanding and will not be considered here.

Figure 4.15: Data points can be mapped bijectively on a regular grid. Once mapped on a regular grid, the fuzzy-wavelet methods can be applied.

After the fuzzy-wavelet method is applied on the regular grid, the approximating function can be written as a weighted sum of splines:

$$\hat{f}(\mathbf{x}) = \sum c_{m,n} \cdot \phi_{m,n}(\mathbf{x}) \qquad \textbf{(4.12)}$$

with $\phi_{m,n}(x) = 2^m \cdot \phi(2^m \cdot x - n)$.

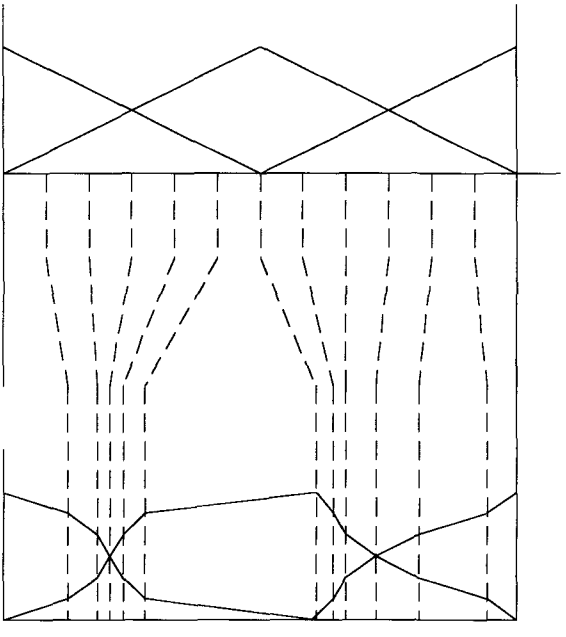


Figure 4.16: The effect of nonlinear mapping is illustrated on second-order splines. The data points correspond to the lower end of the vertical dashed lines.

The inverse mapping does not change the weights, it modifies only the shape of the splines. The method is very easy to implement and a fuzzy interpretation of the results is possible. At one level of resolution, all membership functions sum up to one and form therefore a partition of unity. Also the membership functions do not take negative values. This is shown in fig. 4.16 with a one-dimensional example. The shape of the membership depends on the position and density of the datapoints in the input space. The dependence on the density of points is clearly seen in fig. 4.16. In that sense, the method is highly adaptive. In general, the transparency of the fuzzy rules is smaller than in fuzzy systems using a dictionary of pre-defined fuzzy functions. This is the main drawback of the method, besides the fact that complexity reduction is much more difficult. There has not been at time enough work along that line to discuss realistically the potential of this method. This approach bears nevertheless some promises due to its built-in adaptivity.

Missing data

In most applications, rules validation is probably the most important single step in learning. The validation process is rarely fully automatic, as in most cases some human intervention is necessary. One is very often confronted to the problem that the input data is sparsely populated. Large regions may be even free of any data. The human operator is confronted to a decision on what strategy to follow. There are essentially four alternatives:

-*Empty regions are ignored, because they do not correspond to the definition range of the input space.* This is an acceptable solution, if one can guarantee that the empty regions in the input space do never occur in real applications.

-*New data are collected specifically within the empty regions.* This is a current approach during the development of new sensors.

-*Default rules are added to the system.* Adding default rules is quite common in sensorics or control. The default rules may have the function to guarantee an acceptable response of the system under very difficult conditions.

-*Rules within the empty regions are computed from neighboring regions.* Generalization requires using interpolation and extrapolation techniques. This will be the subject of next section.

Interpolation and approximation methods

In the case of missing data, the fuzzy system can be completed by using different approximation techniques. We will present below a few methods for that purpose. Fuzzy-wavelet methods are suitable to modeling problems with several variables. For this reason, we focus on some methods that work well in a multi-dimensional setting. As the complexity of interpolation and approximation

techniques increases rapidly with the number of variables, only the simplest methods are practicable in real problems. A first possibility consists of using interpolation techniques. Let us remind that an interpolation function is a function ϕ for an ensemble of point (x_i,y_i) if :

$$\phi(x_i) = y_i \quad \forall i \tag{4.13}$$

Spline interpolants

According to Lagrange formula, it is possible to fit an interpolating polynomial of degree N-1 through any curve given by N points. In spline interpolation schemes, one is concerned with a slightly different problem. One tries to fit piecewise polynomials functions to N points (x_i,y_i). Each piecewise polynomial is of order k and the interpolating function is requested to be in C^{k-1}, the set of functions with continuous $(k-1)^{th}$ derivative. The second order spline interpolation corresponds to interpolating between the different points with a continuous function consisting of piecewise linear functions. The cubic spline interpolation is certainly the most popular spline as it represents a good compromise between necessary computing power and smoothness. It is given by the following formula for the interpolating function y:

$$y = A \cdot y_j + B \cdot y_{j+1} + C \cdot y''_{j+1} + D \cdot y''_{j+1} \tag{4.14}$$

with $A=(x_{j+1}-x_j)/(x_{j+1}—x_j)$, $B= 1-A$, $C=1/6\ (A^3-A)\ (x_{j+1}- x_j)^2$,
$D=1/6\ (B^3-B)\ (x_{j+1}- x_j)^2$
The terms y'' are computed through solving the equation:

$$\begin{aligned} &1/6 \cdot (x_j - x_{j-1}) \cdot y''_{j-1} + 1/3 \cdot (x_{j+1} - x_{j-1}) \cdot y''_j + \\ &1/6 \cdot (x_{j+1} - x_j) \cdot y''_{j+1} = \\ &(y_{j+1} - y_j)/(x_{j+1} - x_j) - (y_j - y_{j-1})/(x_j - x_{j-1}) \end{aligned} \tag{4.15}$$

The above equations show that in order to compute cubic spline coefficients, one has to solve essentially a linear problem. This is one of the reason why splines are so popular. The method works well in one dimension. The complexity of the problem increases rapidly with the dimension, so that the method is only recommended at low dimension.

B-splines interpolants can be also used to describe empty regions (de Boor, 1978). B-splines are piecewise polynomial functions with a compact support. They are very often used to interpolate between data.

Suppose N points (x_i, y_i) are known and one looks for an interpolation between these points with piecewise polynomial functions $\phi_k(x)$ with a compact support: $y(x) = \sum c_k \phi_k(x)$ and $y(x_i)=y_i$

Schoenberg and Whitney (1953) have shown that if a point x_i (N>i>1) is within the support of each function $\phi_k(x)$ then the problem has a unique solution. The solution to this problem is obtained through a Gauss elimination method. The shape of the splines depends on the position of the knots. The resolution of this problem necessitates the knowledge and the storage of the N points. For a large number of points, this is a large inconvenience.

Multivariate approximation methods

From the Shannon sampling theorem, one knows that a band-limited function is recoverable from sample points on regular grid provided the sampling rate is large enough. Feichtinger (1990) has shown that the reconstruction is still possible if the sampling points are not on a regular grid if the sampling density is high enough. Practically, the method is difficult to implement for multivariable interpolations as it involves an inverse Fourier transform. From a practical point of view, interpolating in a high-dimensional space is difficult and one generally prefers using an approximation method. An exception is the Delauney interpolating scheme.

At intermediate dimension, the Delaunay triangulation methods can be used to interpolate the data. We will present here the method in a 3-dimensional space. Suppose, one wants to interpolate the value of the function at a point Xg located on a regular grid. The Delaunay interpolation scheme is shown in fig. 4.17.

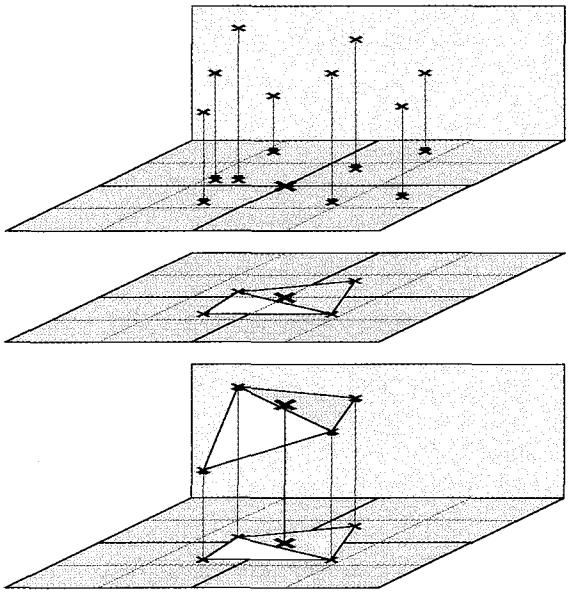


Figure 4.17: Delaunay's triangulation method.

Data points corresponding to the underlying function $y=f(x_1, x_2,...,x_d)$ are projected on the input space. First the Voronoi cell, around the point on the grid

one wants to estimate, is computed. The Voronoi cell corresponds to the ensemble of points closer or at the same distance to the chosen point than to any other point. Triangulation is carried out and d=3 points are selected (For details on the triangulation method see Okabe (1999)). The d points define a plane P (or hyperplane at higher dimension) in $\Re^d$: $y=P(x_1, x_2,...,x_d)$. The value of the point on the grid f(**Xg**) is then computed from the equation of P.

The Voronoi construction is quite general and works in any d-dimensional space. Given a finite set of distinct points in $\Re^d$, the space may be divided into a number of Voronoi cell. The Delaunay diagram can be constructed by connecting the points whose Voronoi cell share a (d-1) dimensional face.

A simple approach that works well at high dimension is the *neighbor-based* interpolation. Each value on the grid is computed from its nearest-neighbor with an averaging procedure. As an example, one may consider the simple estimation:

$$f(\mathbf{X}_g) = 1/N \cdot \sum_{i \in \{N \text{ nearest neighbors to } \mathbf{X}_g\}} y(x_i) \qquad \textbf{(4.16)}$$

If points are now uniformly distributed, the estimation may be improved by a weighted averaging method. We will limit the discussion of a single method, a multiresolution scheme using cardinal B-splines, that is quite reminiscent of the fuzzy-wavelet approach. Consider the problem of estimating a function f(x) as a weighted sum of splines forming a multiresolution:

$$\hat{f}(x) = \sum_{m,n} c_{m,n} \cdot \phi_{m,n}(x) \qquad \textbf{(4.17a)}$$

Assume further that a number of points are known. Finding good values for $c_{m,n}$ is a delicate problem, as the functions $\phi_{m,n}$ are linearly dependant. The problem can be much simplified if the equations are decoupled:

$$\hat{f}(x) = \sum_{m} \hat{f}_m(x) \qquad \textbf{(4.17b)}$$

$$\hat{f}_m(x) = \sum_{n} c_{m,n} \cdot \phi_{m,n}(x) \qquad \textbf{(4.17c)}$$

This permits to approximate the data at a very low resolution and to correct iteratively the approximation function at the higher levels of resolution. The equations are first solved at a low resolution. This can be done either using a neural network, a kernel estimator or singular-valued decomposition. The residue $R(x_k) = f(x_k) - \hat{f}(x_k)$ is computed and the procedure is iterated on the residue at one level of resolution higher. The approximating function is assumed to be a linear combination of cardinal splines of order k:

$$\hat{f}(x) = \sum_{m,n} c_{m,n} \cdot N_m^k(x-n)$$

The approximating function $\hat{f}(x)$ at the lowest level of resolution is computed with a simple average scheme.

The function f(x) is first approximated at the lowest resolution

$$\hat{f}_0(x) = \sum_n c_{0,n} \cdot N_0^k(x-n) \tag{4.18}$$

The residue $R(x_k)$ is computed

$$R(x_k) = f(x_k) - f_0(x_k) \tag{4.19}$$

The procedure is repeated at the next higher level of resolution, using the residues $R(x_k)$ as input data. The residue is approximated by the function $\hat{f}_1(x)$ with

$$\hat{f}_1(x) = \sum_n c_{1,n} \cdot N_1^k((x-n)) \tag{4.20}$$

After L iterations the input data are estimated by

$$\hat{f}(x) = \sum_{m=0}^{L} \hat{f}_m(x) \tag{4.21}$$

The advantage of using splines is that the method can be easily adapted to higher dimensions by taking tensor products of splines of same resolution. Other functions such as radial basis functions might have been chosen as well (gaussian, triangular, poly-harmonic,...). Figure 4.18 shows an example in which a feedforward perceptron using an instantaneous gradient descent was taken. Let us mention that in order to avoid over fitting, a regularization approach may have been used. The above method can be seen as a generalization of the neurofuzzy approach to multiresolution analysis. The method works both for on-line and off-line problems and at any dimension. A further advantage is that it does not require the input space to be defined on a grid. Large empty regions are approximated with low-resolution splines, while small empty regions are approximated with high-resolution splines.

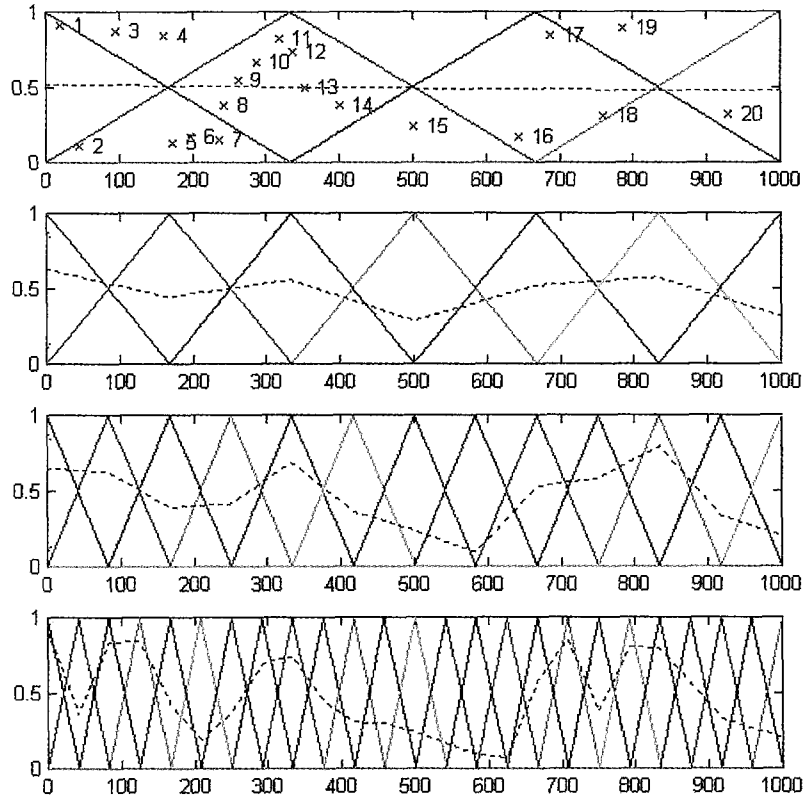


Figure 4.18: An approximation of a function given by 20 points is made iteratively by adding higher resolution cardinal splines that model the residue.

The above enumerated properties makes the multiresolution neurofuzzy method quite attractive for applications. There are two drawbacks of the above method that should be mentioned: the first being the danger of over fitting in data-rich regions. In order to prevent over fitting a cross-validation method is necessary. The second difficulty is the implementation of the method in on-line problems if all data points cannot be stored due to a low memory capacity of the hardware. In part 5, methods will be discussed that overcome that problems by using multiresolution feedforward neural networks with a modified instantaneous gradient method.

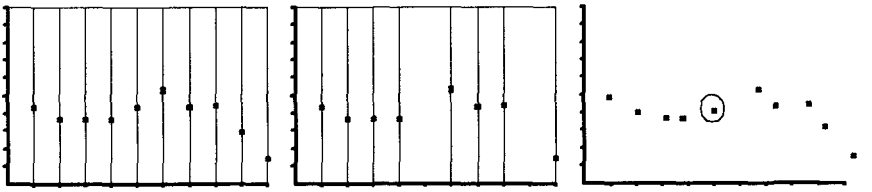


Figure 4.19: The fuzzy-wavelet method applies to data on a regular grid. If a few points are missing, the missing points can be obtained by interpolation or extrapolation techniques. If data are not on a regular grid, then a neural method can be used or the membership functions can be deformed adaptively using the method in part 4. Part 5 will present further techniques that can be applied to on-line learning

In summary, while the fuzzy-wavelet methods presented in the first sections of part 4 dealt with regularly spaced datapoints in the input space, the previous sections did extend the method to random design, and to the situation in which some points are missing. The above methods do not work well in on-line learning. For on-line learning wavelet-based neural networks or estimators are preferably used. This will be the subject of the next chapters.

This page is intentionally left blank

PART V ON-LINE LEARNING

This page is intentionally left blank

5. On-Line Learning

Off-line or batch learning was the main theme in part 4. The two central ideas behind fuzzy-wavelet techniques were explained. First a dictionary of pre-defined membership functions forming a multiresolution is taken. This makes the fusion and splitting of rules simple. As each membership function is a linear superposition of higher resolution membership functions, rule splitting is straightforward. Rules fusion can be carried out through filtering. The approximation coefficients at high resolution are transformed with the low-pass filter associated to the considered wavelet. Second, learning from data can be made using wavelet-based algorithms. The approximation coefficients in the wavelet decomposition correspond to the output value in the singleton Takagi-Sugeno model and the scaling functions are interpreted as membership functions.

Transparency and readability were the most important motivations behind the development of these methods combining fuzzy logic and multiresolution analysis. The interpretation of the rules is greatly facilitated by the fact that most semantic and redundancy problems, usual in most neurofuzzy methods, are solved per design. We have examined in the previous sections, how to handle situations, in which data are missing or data are not on a regular grid. For batch learning, the interpolation and approximation methods in part 4 can be used to estimate values on a grid. An alternative solution is to map the data on a regular grid and to apply fuzzy-wavelet methods to the transformed data.

Part 4 dealt with off-line problems in a deterministic design. In part 5, the fuzzy-wavelet formalism is extended to random designs. Part 5 focuses on on-line learning. It explores the connections between wavelet theory, fuzzy logic, neural networks and approximation theory.

The first section introduces wavelet-based neural networks. In the second section, wavelet networks are extended to biorthogonal wavelets. Within the framework of biorthogonal wavelet networks, multiresolution neurofuzzy methods are proposed. These methods offer a simple solution to the problem of validation during on-line learning. New rules are added to the system, using a simple wavelet-based validation procedure, as new data become available.

Part 5 introduces fuzzy wavelet networks within the more general framework of wavelet networks. For the readers mostly interested in fuzzy logic, fig. 5.1 shows the different sections in which the fuzzy learning methods can be found. Figure 5.1 summarizes also the different approaches, presented in this book, to develop a multiresolution fuzzy system from data:

-For off-line learning with data on a regular grid, appropriate membership functions and rules are determined with fuzzy-wavelet techniques. The most

appropriate rules are chosen based on the decomposition coefficients or by using a matching pursuit algorithm. If some data are missing, the values on a grid can be estimated with standard regression and approximation techniques.

-For a random design, approximation techniques can also be chosen, though an alternative solution consists of mapping the input space onto a regular grid. In that later case, the position and shape of the membership functions depend on the location and density of points in the input space.

-For on-line learning, wavelet-based neural methods (fuzzy wavenets) or multiresolution estimation (fuzzy-wavelet estimators) are the methods of choice.

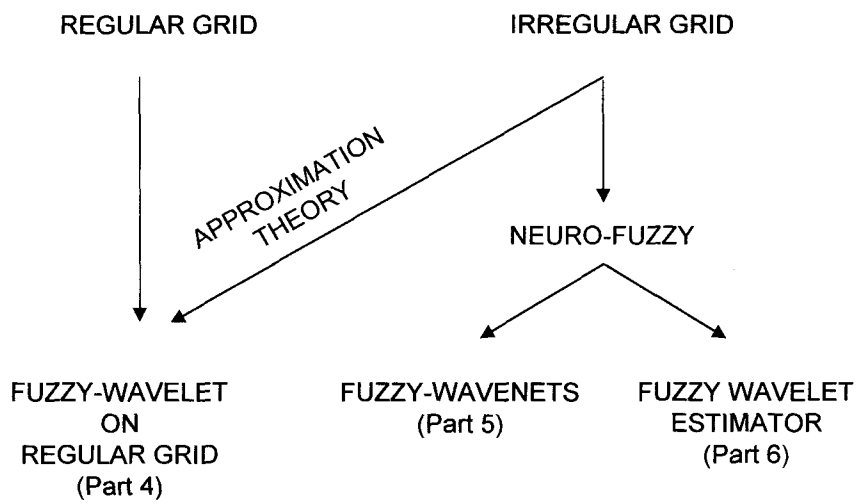


Figure 5.1: Summary of the different methods to develop fuzzy rules from data with wavelet-based approaches.

Wavelet-based neural networks

Wavelet theory has a profound impact on signal processing as it offers a rigorous mathematical approach to the treatment of multiresolution. The combination of neural networks and wavelet theory has lead to a number of new techniques: wavelet networks, wavenets, fuzzy wavenets. In this section, we want to review wavelet-based neural networks. Wavelet analysis and neural networks have been combined in numerous manners. We distinguish two categories of methods. In the first one, the wavelet part is essentially decoupled from learning. A signal is decomposed on some wavelet and the wavelet coefficients are furnished to a neural network. In the second category, wavelet theory and neural networks are

combined into a single method. We limit the scope of this chapter to the second category, which covers wavelet networks, wavenets and fuzzy wavenets.

The introduction of wavelet theory into neural networks has resulted into the development of wavelet networks. Wavelet networks are feedforward neural networks using wavelets as activation function. Wavelet networks have been used in classification and identification problems with some success. The strength of wavelet networks lies in their capabilities of catching essential features in *frequency-rich* signals. In wavelet networks, both the position and the dilation of the wavelets are optimized besides the weights. Wavenet is another term to describe wavelet networks. Originally, wavenets did refer to neural networks using dyadic wavelets. In wavenets, the position and dilation of the wavelets are fixed and the weights are optimized by the network. We propose to adopt this terminology. The theory of wavenets has been generalized to biorthogonal wavelets (Thuillard, 1999a). This extension to biorthogonal wavelets did permit the development of fuzzy wavenets (Thuillard, 2000a). Fuzzy wavenets extend wavelet-based learning techniques to on-line learning. A major advantage of fuzzy wavenets techniques in comparison to most neurofuzzy methods is that the rules are validated, on-line, during learning by using a simple algorithm based on the fast wavelet decomposition algorithm.

The similarities existing between the structure of a feedforward neural network and a wavelet decomposition have been used in so-called wavelet networks. A wavelet network is a 3-layers feedforward neural network in which $\psi(a_i \cdot x + b_i)$ is a wavelet.

The output of the 3-layers neural network is

$$f(x) = \sum_{i=1}^{k} w_i \cdot \psi(a_i \cdot x + b_i) \qquad \textbf{(5.1)}$$

with ψ the activation function and a_i, b_i, w_i the network parameters (weights) that are optimized during learning.

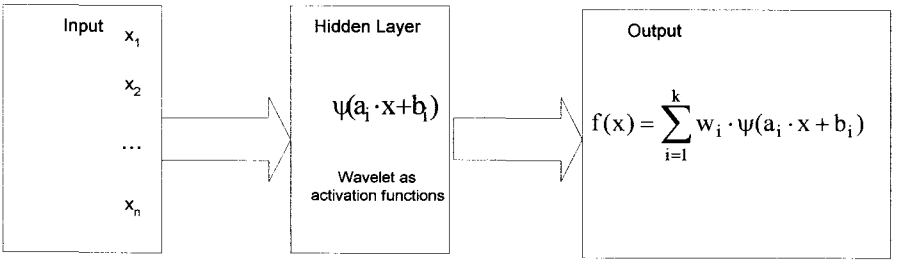


Figure 5.2: The structure of a wavelet network is very often the one of a feedforward neural network.

The dilation, translation and weights are optimized during learning. If the network is properly initialized, then the network can be quite parsimonious. If only the weights are optimized in (5.1), and the activation function is of the form $\psi_{m,n} = \psi(2^m x - n)$,with m,n integers, the network is referred to as a wavenet. A subset of wavelet networks are the so-called fuzzy wavelet networks or fuzzy wavenets. Using the two-scales relation (Mallat, 1998), a wavelet can be decomposed into a sum of scaling functions $\psi(x) = \sum_r h_{n-2r}\phi(2x - r)$. The wavelet network, given by (5.1), can be put under the form:

$$f(x) = \sum_{m,n,r} d_{m,n} \cdot h_{n-2r} \cdot \phi_{m+1,n}(x) + \bar{f} \qquad \textbf{(5.2)}$$

Fuzzy wavenets are wavelet networks based on wavelets with some special properties: the scaling function associated to these wavelets must be symmetric, everywhere positive and with a single maxima. Under these conditions, the scaling functions can be interpreted as fuzzy membership functions. Figure 5.3 summarizes the different wavelet-based neural networks using a feedforward type of networks. Fuzzy wavenets are included within the category of wavelet networks.

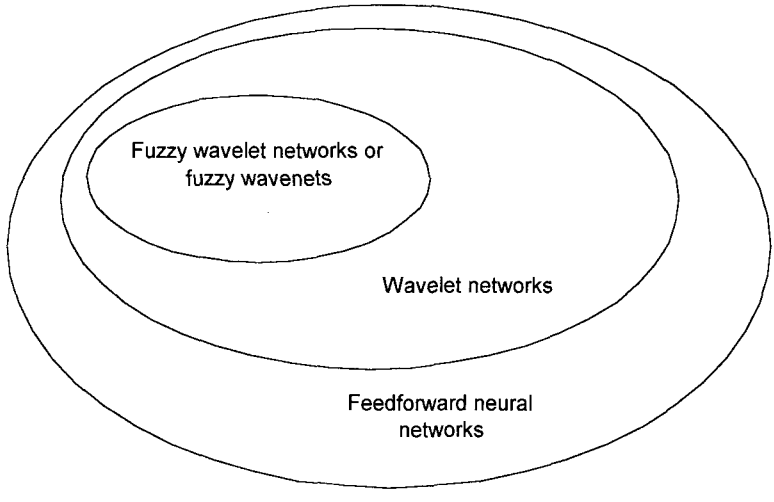


Figure 5.3: The most popular wavelet networks are based on the perceptron structure. Fuzzy wavelet networks, also called fuzzy wavenets, can be regarded as a neurofuzzy model which belongs at the same time to the set of wavelet networks.

Wavelet networks

The origin of wavelet networks can be traced back to the work by Daugman (1988) in which Gabor wavelets were used for image classification. Wavelet networks have become popular after the work by Pati (1991, 1992), Zhang (1992), and Szu (1992). Wavelet networks were introduced as a special feedforward neural network. Zhang et al. did apply wavelet networks to the problem of controlling a robot arm. As mother wavelet, they use the function

$$\psi(x) = (x^T \cdot x - \dim(x)) \cdot e^{-1/2 \cdot x^T \cdot x} \qquad \textbf{(5.3)}$$

Szu et al. take a different function, $\cos(1.75\ t)\ \exp(-t^2/2)$, as mother wavelet for classification of phonemes and speaker recognition. Simple combinations of sigmoids were chosen by Pati (1991). This approach has been generalized by Fernando Marar (1996) to polynomial functions of the sigmoid function.

Wavelet networks using the 3 layers perceptron structure are of the general form:

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i \cdot \det(D_i^{1/2}) \cdot \psi[D_i \cdot \mathbf{x} - t_i] \qquad \textbf{(5.4)}$$

with D the diagonal dilatation matrix and t the translation vector.

For classification the output signal may be further processed with a sigmoid function σ. In that case, the output is given by $\sigma(f(\mathbf{x}))$ (Szu, 1992).

A motivation for using wavelet networks is that there are universal function estimators that may represent a function to some precision very compactly. This follows from the work by Hornik (1989) and Kreinovich (1994). Hornik has shown that an arbitrary continuous function on a compact set can be approximated by a 3-layers neural network within a precision ε. More precisely, assume an arbitrary function f with p continuous derivatives on (0,1) and $\left|f^p(x)\right| \le \Delta$, such that the function is equal to zero in some neighborhood of the end points. The function f: $\Re \rightarrow \Re$ can be approximated by an expression of the type:

$$f(x) = \sum_{h=1}^{H} \beta_h \cdot s(w_h \cdot x + b_h) \qquad \textbf{(5.5)}$$

with H the number of neuron in the hidden layer, w the weight between the input and the hidden layer, and β the weight between the hidden and the output layer. The function s(x) is the transfer function, for instance the sigmoid function. A wavelet network is a particular case of (5.5). Kreinovich et al. (1994) have proven that wavelet neural networks are asymptotically optimal approximators for functions of one variable. Wavelet neural networks are optimal in the sense that they require the smallest possible number of bits to store, for reconstructing a function within a precision ε.

From the practical point of view, the determination of the number of wavelets and their initialization represent two major problems with wavelet networks. A good initialization of wavelet neural networks is extremely important to obtain a fast convergence of the algorithm. A number of methods have been implemented. Zhang et al. (1992) initialize the coefficients with an orthogonal least-squares procedure. As an alternative, the dyadic wavelet decomposition may be used to initialize the network. Echauz (1998) applies a clustering method to position the wavelets. The distribution of points about a cluster permits to approximate the necessary dilation of the wavelet. Echauz (1996) proposes also an elegant method using trigonometric wavelets. He uses functions of the form:

$$\cos\,\mathrm{trap}(x) = \cos(3\pi/2 \cdot x) \cdot \min\{\max\{3/2 \cdot (1 - |x|), 0\}, 1\} \quad \textbf{(5.6)}$$

Trigonometric wavelets can be approximated by polynomials. Fitting of the polynomial is a linear problem that is solved more easily than fitting trigonometric wavelets. The fitting parameters of the polynomials can be used to approximate the initializing parameters of the corresponding wavelets. In Boubez (1993), the network is initialized by positioning and approximating first low resolution wavelets. New higher resolution wavelets are introduced and initialized subsequently to minimize the score. Rao et al. (1993) use the principle of cascade correlation learning architecture to train the network. New wavelets are added one by one and at each step, the network is trained till convergence is reached. Yu et al. (1996) opt for the opposite approach, the wavelet network uses first a large number of functions. The wavelet network is made subsequently as compact as possible using a shrinkage technique to delete not too important nodes.

Backpropagation algorithms, conjugate gradient method (Szu, 1992) stochastic gradient algorithm (Zhang, 1992) or genetic algorithms (Prochazka, 1994) have been used for training the network.

A number of interesting applications have taken advantage of the multiresolution properties of wavelet networks.

Many manufacturing process monitoring systems have the function of detecting abnormal vibrations (Pittner, 1998). For vibration detection and classification, wavelet-based methods represent good alternatives to Fourier analysis. Engine knock detection systems have been developed by PSA-Peugeot-Citroen (Thomas, 1996) on the basis of wavelet networks. Another related application is the detection of vibrations in defective circuit breakers in electric power (Lee, 1999).

Wavelet networks have been implemented with success to identify and classify rapidly varying signals, for instance to identify high risks patients in cardiology (Dickhaus, 1996) or for echo cancellation (Li, 1996).

Major efforts have been undertaken in the field of speech segmentation and speaker recognition following the pioneering work by Szu et al. (Szu, 1992, 1996, 1998). The error rate in continuous speech recognition is of the order of 5%. Speech recognition systems have difficulties to separate mixed sounds, like a

ts into t and s. Reliable acoustic segmentation is regarded as a way to improve speech recognition.

Forecasting and prediction of chaotic signals are two other promising fields of applications. Prediction of chaotic time series with wavelet networks were obtained from a limited number of datapoints. Excellent results were obtained on chaotic times series (Cao, 1995). The multiresolution character of wavelets permits to catch long terms and short terms variations. Applications in forecasting range from economical predictions (Cao, 1996) to prediction of short term load in power station (Chang, 1998) or channel equalization (Chang, 1994).

Wavelet networks have been tested on a number of classical control problems, from the detection of small variations in a plant to the control of robotics arms (Katic, 1997).

Studies on radar applications have dealt with aircraft velocity estimation (Sanchez-Redondo, 1998) or rain forecasting (Yeung, 1996).

Let us mention also two applications in the field of image processing: face tracking (Kruger, 1994) and real environments characterization for haptic display (Miller, 1998).

A new generation of chemical sensors based on micro-hotplate gas sensors have been developed at NIST. The sensor consists of an array of micro-hotplates on a silicon wafer using CMOS technology. The different reaction kinetics of the different gases can be used to enhance the sensor' detection capabilities. The temperature of the sensor is modulated rapidly. The dynamic response of the sensors to different gases are analyzed with a wavelet network using Mexican hat wavelets (Kunt, 1998).

An interesting alternative to wavelet networks consists of using a dictionary of dyadic wavelets and to optimize only the weights w_i. This approach is generally referred to as wave-net or wavenets.

Dyadic wavelet networks or wavenets

Wavenets were first proposed by Bakshi et al. (1994). In its simplest version, a wavenet corresponds to a feed-forward neural network using wavelets as activation functions.

$$f(x) = \sum_{m,n} d_{m,n} \cdot \psi_{m,n}(x) + \bar{f} \qquad (5.7)$$

with $\bar{f}$ the average value of f, $d_{m,n}$ the coefficients of the neural network and ψ the wavelet.

Wavenets have been generalized to biorthogonal wavelets (Thuillard, 1999a, 2000a, 2000b). The principal difference to orthogonal wavelets is that the

evolution equation depends on the dual wavelet. We have proposed the following evolution equation for biorthogonal wavelets.

$$\hat{d}_{m,n}(k) = \hat{d}_{m,n}(k-1) - LR \cdot (f(x) - y_k(x)) \cdot \tilde{\psi}_{m,n}(x) \qquad \textbf{(5.8)}$$

with LR the learning rate and $y_k(x)$ the k^{th} input point. For datapoints that are independent, uniformly distributed copies of a random variable X, the estimated wavelet coefficients $\hat{d}_{m,n}(k)$ converge adiabatically to $d_{m,n}$ in the limit of a very low learning rate.

For orthogonal wavelets, (5.8) reduces to

$$\hat{d}_{m,n}(k) = \hat{d}_{m,n}(k-1) - LR \cdot (f(x) - y_k(x)) \cdot \psi_{m,n}(x) \qquad \textbf{(5.9)}$$

Fuzzy wavenets

Let us recall the framework in which we have worked till now and the situation we have left in part 4. A major challenge to fuzzy logic is the translation of the information contained implicitly in a collection of data points into linguistically interpretable fuzzy rules. Neurofuzzy methods have been developed for this purpose. A serious difficulty with many neurofuzzy methods is that they do often furnish rules without a transparent interpretation; a rule is referred as being transparent if it has a clear and intuitively correct linguistic interpretation. A solution to this problem is furnished by multiresolution techniques. The basic idea is to take a dictionary of membership functions forming a multiresolution and to determine which membership functions are the most appropriate to describe the data points. In order to associate a linguistic interpretation to each membership function, the membership functions are chosen among the family of scaling functions that have the property to be symmetric, everywhere positive and with a single maximum. This family includes among others splines and some radial functions. The main advantage of using a dictionary of membership functions is that each term, such as *small* or *large* is well defined beforehand and is not modified during learning. The multiresolution properties of the membership functions in the dictionary function permits to fuse or split membership functions quite easily so as to put the control surface under a linguistically understandable and intuitive form for the human expert.

In the singleton model, the fuzzy rules are expressed under the form: R_i: if x is A_i then $y = b_i$. Here A_i are linguistic terms, x is the input linguistic variable, while y is the output variable. The value of the input linguistic variable may be crisp or fuzzy. If spline functions N^k are taken, for instance, as membership function $\mu_{A_i}(\hat{x}) = N^k(2^m \cdot \hat{x} - n)$ then the system is equivalent to

$y = \sum_j b_j \cdot N^k(2^m \cdot \hat{x} - n)$. In this particular case, the output y is a linear sum of translated and dilated splines. This means that under this last form the singleton Takagi-Sugeno model is equivalent to a multiresolution spline model. It follows that wavelet-based techniques can be applied here.

For on-line problems, rules validation is the main issue especially if little memory is available. A number of multiresolution methods have been presented in the previous chapters. None of them can suitably tackle the validation problem. Ideally, one would like a system containing a small number of rules as only a few points are available. New rules would be added to the systems as more information is gathered. We will introduce in the next sections, a number of methods, that just do that using a simple and efficient wavelet-based validation procedure. Several variants are presented; all have in common to use the fast wavelet algorithm to validate the new rules and are therefore referred to as fuzzy wavenets.

These fuzzy wavenets methods combine wavelet theory to fuzzy logic and neural networks. They permit to determine and validate adaptively appropriate fuzzy rules in on-line problems. The model is refined as more data are furnished to the system. With only a few datapoints, the information on the underlying surface is small and a low resolution description of the system is appropriate, while with an increasing number of datapoints, a higher resolution may be justified. New rules are added to the description of the surface as more datapoints are processed. The rules are validated by using an automatic procedure based on the fast wavelet decomposition and reconstruction algorithm. Learning is fully automatic and does not require any external intervention, making these methods very useful in practical applications, for instance during field testing of sensors and detectors. The detectors are installed in challenging test objects located all over the world. The sensor' signals are processed autonomously by a low end microprocessor in the detector. The information is processed and stored under a compressed form. The compressed information can be at any time transmitted to the laboratory, for instance through modern communications means (mobile phone, modem,...). This approach permits to carry out large-scale field testing at moderate costs.

Multiresolution identification using spline wavenets have been implemented with success to the modeling of chaotic data (Billings; 1999). Billings et al. use a multiresolution network to model for instance a Duda oscillator. The multiresolution structure of the network permits to catch both long and short-range correlations quite efficiently. The neural network uses a feedforward neural network with an instantaneous gradient descent. The model is constructed by using a method inspired by ASMOD. As learning is off-line, their model complete the list of multiresolution neurofuzzy models presented in part 4 for off-line learning. The main difference, between Billings' approach and the one presented here, lies in a small but important detail in the evolution equation. In fuzzy wavenets, the evolution equation uses the dual scaling functions and not

the scaling function itself. Using the dual permits to give an interpretation of the weights $\hat{c}_{m,n}$ in the expression $\hat{f}(\mathbf{x}) = \sum_{m,n} \hat{c}_{m,n} \cdot \phi_{m,n}(\mathbf{x})$. For locally uniformly distributed data and for small learning rates, the coefficients $\hat{c}_{m,n}$ tend, under some mild conditions on $f(\mathbf{x})$, towards the approximation coefficient $c_{m,n}$ of the wavelet decomposition of $f(\mathbf{x})$. The coefficients in both approaches are not the same. This is already seen from a one-level model: $\hat{f}(\mathbf{x}) = \sum_{n} c_{M,n} \cdot \phi_{M,n}(\mathbf{x})$. The instantaneous gradient method minimizes the locally weighted mean-squares error, while a global method would result in a MSE solution. The wavelet coefficients $\hat{c}_{m,n}$ do not have to be MSE solutions. For orthogonal wavelets, this would hold, but for biorthogonal wavelets it does not hold in general.

Learning with fuzzy wavenets

Figure 5.4 shows the architecture of the learning algorithm. It consists of a series of neural networks, using both wavelets $\psi_{m,n}(x)$ and scaling functions $\phi_{m,n}(x)$ as activation functions. Each neural network takes activation functions of a given resolution.

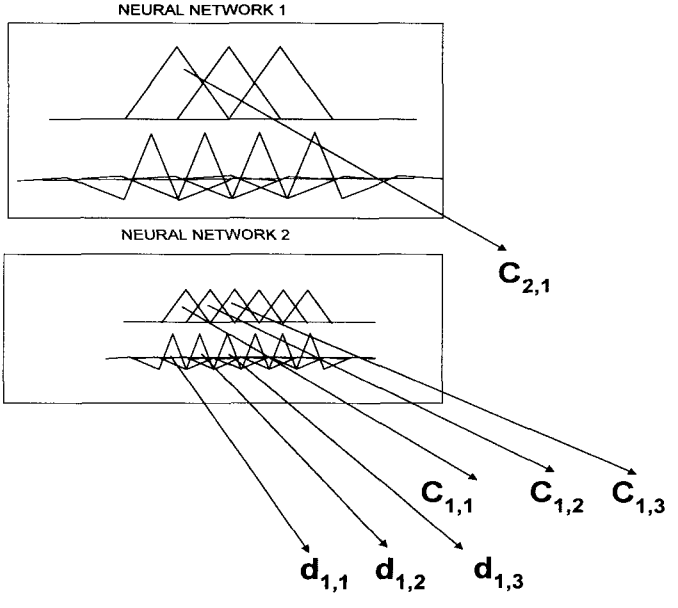


Figure 5.4: Structure of a fuzzy-wavenet. The input signal is approximated at several resolution as a weighted sum of wavelets $\psi_{n,m}$ and scaling functions $\phi_{m,n}(x)$ at a given resolution.

The m^{th} neural network optimizes the coefficients $\hat{c}_{m,n}$ and $\hat{d}_{m,n}$, with $f_m(x)$ the output of the m^{th} neural network.

$$f_m(x) = \sum_n \hat{d}_{m,n} \cdot \psi_{m,n}(x) + \sum_n \hat{c}_{m,n} \cdot \phi_{m,n}(x) \qquad \textbf{(5.10)}$$

The evolution equation for the details $\hat{d}_{m,n}(k)$ and the approximation coefficients $\hat{c}_{m,n}(k)$ at step k are given by

$$\hat{d}_{m,n}(k) = \hat{d}_{m,n}(k-1) - LR \cdot (f_m(x) - y_k(x)) \cdot \tilde{\psi}_{m,n}(x) \qquad \textbf{(5.11)}$$

$$\hat{c}_{m,n}(k) = \hat{c}_{m,n}(k-1) - LR \cdot (f_m(x) - y_k(x)) \cdot \tilde{\phi}_{m,n}(x) \qquad \textbf{(5.12)}$$

with $y_k(x)$, the k^{th} input point and LR the learning rate, $\tilde{\phi}_{m,n}(x)$, $\tilde{\psi}_{m,n}(x)$ the dual functions to $\phi_{m,n}(x)$ and $\psi_{m,n}(x)$. The evolution equations (5.11-12) describe the evolution of $f_m(x)$. Assume datapoints $y_k = f(x_k)$, with x_k uniformly distributed copies of a random variable X. At each step the coefficients $\hat{c}_{m,n}$, $\hat{d}_{m,n}$ are updated by a term which expectation are proportional to

$$E((\hat{f}_m(x) - y_k(x)) \cdot \tilde{\psi}_{m,n}(x)) = <\hat{f}_m(x) - f(x), \tilde{\psi}_{m,n}(x)> = d_{m,n} - \hat{d}_{m,n} \qquad \textbf{(5.13a)}$$

$$E((\hat{f}_m(x) - y_k(x)) \cdot \tilde{\phi}_{m,n}(x)) = <\hat{f}_m(x) - f(x), \tilde{\phi}_{m,n}(x)> = c_{m,n} - \hat{c}_{m,n} \qquad \textbf{(5.13b)}$$

In the adiabatic sense, the expectation of the function $f_m(x)$ converges to the projection of f(x) on the space W_{m+1} under some mild conditions for the function f(x). Since $\psi_{m,n}(x)$ and $\phi_{m,n}(x)$ are independent, it follows that $\hat{c}_{m,n} \to c_{m,n}$ and $\hat{d}_{m,n} \to d_{m,n}$.

Validation methods in fuzzy wavenets

The validation procedure may be explained starting from wavelet theory. For dyadic wavelets, a necessary condition for perfect reconstruction is that the space $V_{m-1} + W_{m-1}$ spanned by the scaling and wavelet functions at level m-1 is equivalent to the space $V_m : V_{m-1} + W_{m-1} \equiv V_m$, which can be symbolically expressed as

It follows that the approximation coefficients at level m can be obtained from the wavelet and approximation coefficients at level m-1. A simple local validation criterion for an approximation coefficient $\hat{c}_{m,n}$ is to request that this coefficient can be approximated from the approximation and detail coefficients $\hat{c}_{m-1,n'}, \hat{d}_{m-1,n'}$ at one lower level of resolution. At each iteration step, the weights from the different networks are cross-validated using a central property of wavelets, namely that the approximation coefficients $c_{m,n}$ at level m can be computed from the approximation and wavelet coefficients at level m-1 using the reconstruction algorithm.

$$c_{m,n} = \sum_r g_{n-2r} \cdot c_{m-1,r} + h_{n-2r} \cdot d_{m-1,r} \qquad \textbf{(5.14)}$$

with g_{n-2r} and h_{n-2r} the filter coefficients for reconstruction (Beware the filter coefficients g, h are to a normalization factor $\sqrt{2}$ identical to the ones defined in part1. The normalization is given here by the relation: $\phi_{m,n}(x) = 2^m \cdot \phi(2^m \cdot x - n)$.) In order for a coefficient to be validated, the difference between the weight of the membership function (model m) and the weight computed from the approximation and wavelet coefficients at one level of resolution lower (model m-1) must be smaller than a given threshold (fig. 5.5).

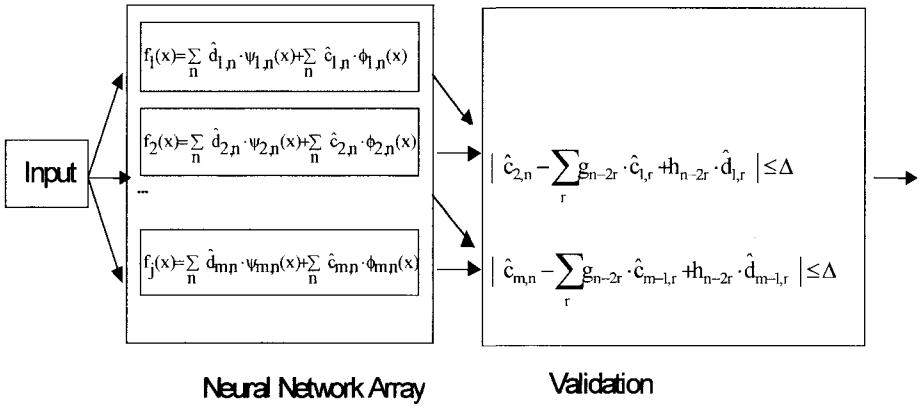


Figure 5.5: The validation module compares the approximation coefficients $\hat{c}_{m,n}(k)$ to the approximation and wavelet coefficients at one level of resolution lower.

As validation criterion for the coefficient $\hat{c}_{m,n}$, we require

$$\left| \hat{c}_{m,n} - \sum_r g_{n-2r} \cdot \hat{c}_{m-1,r} + h_{n-2r} \cdot \hat{d}_{m-1,r} \right| \leq \Delta \qquad \textbf{(5.15)}$$

The most appropriate membership functions and rules are chosen adaptively during learning. With only a few points, not much information on the control surface is known and the control surface is better described with a small number of rules. As the number of points increases, the number of rules is raised if necessary. The method furnishes an automatic procedure to determine adaptively the *best* membership functions and rules. The *best* coefficients are chosen adaptively among the set of validated coefficients. The validated coefficients corresponding locally to the highest resolution are kept (default coefficient= average value).

Learning with wavelet-based feedforward neural networks

The convergence of the fuzzy wavenet method is not too fast, as the method requires for stability reasons to use a small learning rate in comparison to a perceptron. For this reason, one may consider another approach using only scaling functions. The basic structure of the network is similar to the fuzzy wavenets except that the m^{th} neural network optimizes the coefficients $\hat{c}_{m,n}$, with $f_m(x)$ the output of the m^{th} neural network (fig. 5.6).

$$f_m(x) = \sum_n \hat{c}_{m,n} \cdot \phi_{m,n}(x) \qquad \textbf{(5.16)}$$

The evolution equation is given by the following expression

$$\hat{c}_{m,n}(k) = \hat{c}_{m,n}(k-1) - LR \cdot (f_m(x) - y_k(x)) \cdot \tilde{\phi}_{m,n}(x) \qquad \textbf{(5.17)}$$

The validation procedure uses the decomposition algorithm to compare the results at two levels of resolution.

$$c_{m,n} = \sum_k p_{k-2n} \cdot c_{m+1,k} \qquad \textbf{(5.18)}$$

with g the coefficients of the filter associated to the low-pass decomposition filter in the fast wavelet decomposition algorithm. The validation criterion for $\hat{c}_{m,n}$ is then

$$\left| \hat{c}_{m,n} - \sum_k p_{k-2n} \cdot \hat{c}_{m+1,k} \right| \leq \Delta \qquad \textbf{(5.19)}$$

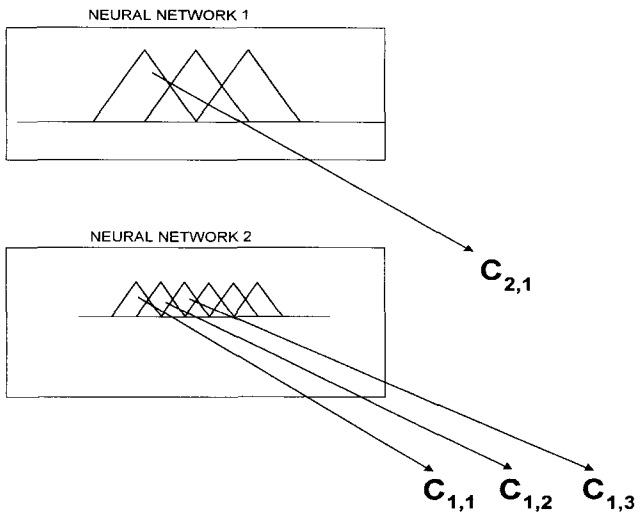


Figure 5.6: Structure of a fuzzy-wavenet. The input signal is approximated at several resolution as a weighted sum of scaling functions $\phi_{m,n}(x)$ at a given resolution.

What are good candidates scaling and wavelet functions at high dimension?

Many problems require the description of a n-dimensional surface with n larger than 2. From the theoretical point of view, there is no limit to the dimension of a wavelet (Kovacevic, 1997; Kugarajah, 1995). An obvious approach to build wavelets in higher dimensions is through tensor products of one-dimensional wavelets (for instance of splines). This approach is versatile enough to describe with sufficient precision many n-dimensional surfaces.

Compactly supported biorthogonal wavelets on any lattice and any dimension can also be generated using the lifting scheme. This approach becomes unpractical at high dimension due to the increasing size of the filter. Radial functions are often very appropriate to deal with high-dimensional spaces. The symmetry of radial functions permits an easy computation of their values. Semi-orthogonal wavelets based on radial functions have been developed by Micchelli (1991). Contrarily to other radial functions (Buhmann, 1996), the construction of Micchelli can be used at any dimension. The scaling functions on which the construction is based are so-called polyharmonic B-splines. Polyharmonic B-splines take the form:

$$f(x) = \|x\|^{2r-d} \cdot \log\|x\|, d \text{ even} \qquad \textbf{(5.20a)}$$

$$f(x) = \|x\|^{2r-d}, d \text{ odd} \qquad \textbf{(5.20b)}$$

with r an integer and d the dimension. The integer must be such that 2r>d
The Fourier transform of the scaling function is given by

$$\hat{\varphi}(\omega) = (\sum_{j=1}^{d} \sin^2(\omega_j/2) / \|\omega/2\|^2)^r \qquad \textbf{(5.20)}$$

with $\omega = (\omega_1, ..., \omega_d)$.

The filter coefficients corresponding to the scaling function can be computed with an inverse Fourier transform. Figure 5.7 shows the scaling function for r=2 and d=2, that is a 2-dimensional scaling function. The wavelet associated to this scaling function is shown below (fig. 5.7b). The wavelet coefficients are similarly obtained by the inverse Fourier transform of the function:

$$\psi(\omega/2) = 2^{-d} \cdot \|\omega/2\|^{2r} |\varphi(-\omega/2)|^2 / \sum_{k \in Z^d} \hat{\varphi}(-\omega/2 + 2 \cdot \pi \cdot k) \qquad \textbf{(5.21)}$$

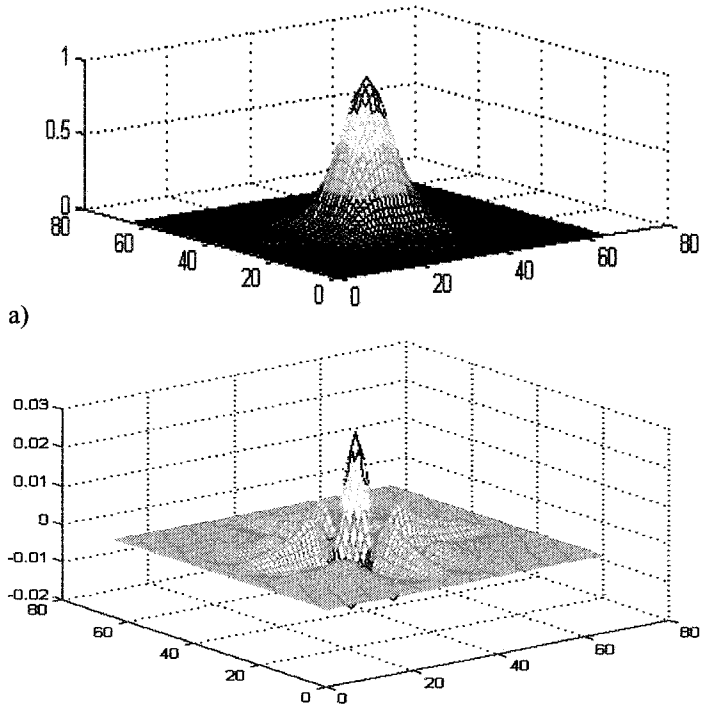


Figure 5.7: Radial scaling functions may be used for multidimensional wavenets as an alternative to tensor products of univariate functions. a) Scaling function in two dimensions, b) associated wavelet.

This page is intentionally left blank

PART VI NONPARAMETRIC WAVELET-BASED ESTIMATION AND REGRESSION TECHNIQUES

This page is intentionally left blank

6. Nonparametric Wavelet-Based Estimation and Regression Techniques

Nonparametric regression and estimation techniques

The main goal of nonparametric regression techniques is to estimate a function from the knowledge of a limited number of points $y_i=y(x_i)$. In many applications, the datapoints are obtained experimentally and may be corrupted with noise. Consider the standard nonparametric regression problem: Let (X,Y) be a pair of random variables with values in $x \in \Re^d$, $y \in \Re$. Assume that $y(x_i) = f(x_i) + \varepsilon_i$ where ε_i are independent $N(0,\sigma)$ normally distributed copies of random variables. A function y = f(x) is the regression function of Y on X if

$$E(Y|X = x) = f(x) \qquad \textbf{(6.1)}$$

For the rest of the discussion, it is important to consider the two typical sampling designs, namely the random sampling and the deterministic sampling.

Random design: The input data x_i are copies of random variables X_i that are independent and identically distributed on [0,1] with density g(x).

Deterministic design: The input variables X_i are non random. The simplest case of a deterministic design is the regular design in which x_i are on a regular grid.

We will limit the discussion to two basic estimation methods: kernel estimators for regression function and density estimation. Due to the particular role of splines in estimation, a small section deals with smoothing splines techniques.

Smoothing and regression are two important problems in which local weighting with kernel functions have found important applications. A smoothing or regression kernel is a positive, generally even, function with unit integral. A regression kernel is generally well localized. Spline functions are often used as kernels. The uniform kernel, the gaussian kernel and the quadratic kernel are also very popular (Eubank, 1999).

Uniform kernel: $\phi(x) = 0.5$, $|x| \le 1$, 0 otherwise

Gaussian kernel: $\phi(x) = (2\pi)^{-1/2} \cdot \exp(-x^2/2)$

Quadratic: $\phi(x) = 0.75 \cdot (1 - x^2)$, $|x| \le 1$, 0 otherwise

Nonparametric kernel estimators permit to estimate a function f(x) from a number of datapoints (x_i, y_i). The estimate $\hat{f}(x)$ is expressed as a weighted sum of translated and dilated kernels:

$$\hat{f}(x) = \sum_{i=1}^{N} S_i(x) \cdot y_i \qquad \textbf{(6.2)}$$

with S_i centered on x_i and N the number of points.

The Watson-Nadaraya and the Müller-Gasser estimators are some of the most popular estimators. They represent two extreme cases in estimation theory (Eubank, 1999). The Watson-Nadaraya estimator is given by

$$\hat{f}(x) = \frac{\sum_{i=1}^{N} \phi(\frac{x_i - x}{h}) \cdot y_i}{\sum_{i=1}^{N} \phi(\frac{x_i - x}{h})} \qquad \textbf{(6.3)}$$

Watson-Nadaraya estimators have some interesting properties. In the case of a random design they can be shown to be bayesian estimators of (x_i,y_i), in which (x_i,y_i) are i.i.d copies of a continuous random variable (X,Y). (In order to simplify the formalism and without loss of generality, we have used 1-dimensional estimators.) The Watson-Nadaraya estimator minimizes also the weighted mean squares error:

$$MSE = \sum_{i=1}^{N} \phi(\frac{x_i - x}{h}) \cdot (y_i - c_i)^2 \qquad \textbf{(6.4)}$$

This is seen by equating the derivative $\partial MSE / \partial c_i = 2 \cdot c_i \cdot \sum_{i=1}^{N} \phi(\frac{x_i - x}{h}) - 2 \cdot \sum_{i=1}^{N} \phi(\frac{x_i - x}{h}) \cdot y_i$ to zero.

The Müller-Gasser estimator is a second very popular estimator defined by the expression:

$$\hat{f}(x) = h \cdot \sum_{i=1}^{N} \int_{\bar{x}_{j-1}}^{\bar{x}_j} \phi(\frac{x_i - x}{h}) \cdot dx \cdot y_i \qquad \textbf{(6.5)}$$

with $\bar{x}_j = (x_{j+1} - x_j)/2$; $\bar{x}_{-1} = x_1$; $\bar{x}_N = x_N$.

Smoothing splines

Smoothing splines can be regarded as an extension of linear regression techniques to accommodate constraints on the smoothness of the fitted function. A natural measure of the smoothness of a one-dimensional function f is an integral function of the m^{th} derivative $f^{(m)}$ of f. Smoothing splines estimators search for a function $\hat{f}(x)$ minimizing the weighted sum of the smoothness measure $\int_0^1 (f^{(m)}(x))^2 \cdot dx$ and the average squared residual $1/N \cdot \sum_{i=1}^{N} (y_i - \hat{f}(t_i))^2$:

$$1/N \cdot \sum_{i=1}^{N} (y_i - f(t_i))^2 + \lambda \cdot \int_0^1 (f^{(m)}(x))^2 \cdot dx \qquad \textbf{(6.6)}$$

When λ is large , smoothness is rewarded and an estimator with large m^{th} derivatives is penalized. In the limiting case, $\lambda \to 0$, the optimized function $\hat{f}(x)$ is a least squares estimator. The estimator is obtained by solving the equation:

$$(\Phi^T \cdot \Phi + n \cdot \lambda \cdot \Omega) \cdot b = \Phi^T \cdot y \qquad \textbf{(6.7)}$$

$\Phi = \{\phi_j(t_i)\}_{i,j=1...n}$ a basis for the set of splines of order 2m.

$\Omega = \{\int_0^1 \phi_i^{(m)} \cdot \phi_j^{(m)} \cdot dt\}_{i,j=1...,n}$

In order to be really nonparametric, the value of λ must be determined. This can be done with trial and error or preferably with cross-validation techniques.

Efficient computing methods have been developed for spline functions. In that case, simple solutions can often be given. For instance, the cubic smoothing spline minimizes the expression:

$$1/N \cdot \sum_{i=1}^{N} (y_i - f(t_i))^2 + \lambda \cdot \int_0^1 (f^{(2)}(x))^2 \cdot dx \qquad \textbf{(6.8)}$$

The method can be extended to functions of p variables by minimizing the expression:

$$1/N \cdot \sum_{i=1}^{N} (y_i - f(t_i))^2 + \lambda \cdot \sum_{k_1+...+k_p} m!/(k_1!...k_p!) \cdot (\int_0^1 ... \int_0^1 (\partial^m f(t)/(\partial t_1^{k_1} ... \partial t_p^{k_p}) \cdot dt_1 ... dt_p$$

(6.9)

The method is known as *thin plate smoothing spline*. Smoothing spline surface fitting technique are described very thoroughly by Wahba (1980). Smoothing splines estimators have found many applications for fitting a curve or a surface to a dataset. A number of computer programs have been developed for

fitting a two-dimensional surface to data. The two-dimensional surface is represented either in a three dimensional space or as a plot in which the surface is given by color coding. Applications to high-dimensional spaces are quite rare, as the number of points necessary to approximate the surface increases rapidly with the dimension (curse of dimensionality).

Wavelet estimators

In the 90's, the statistics community got very interested in wavelet theory. A number of wavelet-based methods were created for nonparametric regression and density estimation. Linear and nonlinear regression methods were developed. By the end of the 90's, the cross-fertilization between the classical wavelet specialists and the statisticians was very significant, partly thanks Donoho and Johnstone work on denoising. This section examines first linear wavelet methods for curve estimation, then describe succinctly some of the methods for density estimation. Finally some nonlinear methods are presented.

Wavelet methods for curve estimation

Most wavelet estimators are based on extensions of the Watson-Nadaraya and Müller-Gasser estimators. Wavelet estimators express the regression function $f_w(x)$ as a weighted sum of wavelets and scaling functions:

$$f_w(x) = \sum_n c_{M_0,n} \cdot \phi_{J_0,n}(x) + \sum_{n,m \geq M_0} d_{m,n} \cdot \psi_{m,n}(x) \qquad \textbf{(6.10)}$$

Different estimators can be used to estimate the values of the coefficients. A wavelet version of the Gasser-Müller estimator was proposed by Antoniadis (1994, 1997) for the fixed design model.

$$\hat{f}_w(x) = \sum_{i=1}^{N} y_i \cdot \int_{A_i} E_m(x,s) \cdot ds \qquad \textbf{(6.11)}$$

where $A_i=[s_{i-1},s_i[$ are intervals that partition $[0,1]$ with $x_i \in A_i$.

The kernel $E_m(x,s)$ is given by

$$E_m(x,s) = 2^m \cdot \sum_n \phi(2^m \cdot x - n)\phi(2^m \cdot s - n) \qquad \textbf{(6.12)}$$

A computationally less demanding estimator is the wavelet version of the Watson- Nadaraya estimator given in the regular design by the expression:

$$\hat{f}_w(x) = c_{M_0,n} \cdot \phi_{M_0,n}(x) + \sum_{m=M_0}^{M_{max}} \sum_n d_{m,n} \cdot \psi_{m,n}(x) \qquad \textbf{(6.13a)}$$

with

$$c_{M_0,n} = \frac{1}{N} \cdot \sum_{i=1}^{N} y_i \cdot \phi_{M_0,n}(x_i)\,;\; d_{m,n} = \frac{1}{N} \cdot \sum_{i=1}^{N} y_i \cdot \psi_{m,n}(x_i) \qquad \textbf{(6.13b)}$$

and $2^{M\max} = N$.

The choice of the low resolution level M_0 determines very centrally the quality of the estimation. A high resolution may lead to a very noisy estimation, while some important signal features may get lost or even artifacts may be created if a too low resolution is taken. For practical applications, cross-validation is often the preferred approach. A simple cross-validation consists of choosing the value M_0 as the minimizer of the error function CV(M) using the leave-one out estimator $\hat{f}_{w,i}$:

$$CV(M) = 1/N \cdot \sum_{i=1}^{N} (y_i - \hat{f}_{w,i})^2 \qquad \textbf{(6.14)}$$

More complicated cross-validation methods have been used, as for instance Wahba (1980) generalized cross-validation procedure.

Biorthogonal wavelet estimators

Wavelet estimators use generally orthogonal wavelets and consequently do not belong to the class of kernel estimators. An interpretation of orthogonal wavelet estimators within the framework of kernel estimation is therefore not possible. In order to create kernel wavelet estimators, wavelet estimators are generalized in the next section to biorthogonal wavelets. Biorthogonal wavelet estimators differ from the standard kernel estimators in the way that the coefficients $\hat{c}_{m,n}$ are obtained. The coefficients $\hat{c}_{m,n}$ are computed using the dual scaling functions. For instance, we will show that the coefficients can be computed with a modified Watson-Nadaraya estimator using the dual spline scaling functions $\hat{\phi}_{m,n}$. For the Müller-Gasser wavelet estimator, the kernel becomes:

$$E_m(x,s) = 2^m \cdot \sum_n \phi(2^m \cdot x - n) \cdot \tilde{\phi}(2^m \cdot s - n) \qquad \textbf{(6.15)}$$

with $\tilde{\phi}(2^m \cdot s - n)$ the dual function of $\phi(2^m \cdot s - n)$.

The generalization for the wavelet equivalent Nadaraya-Watson estimator is given by

$$\hat{f}_w(x) = c_{M_0,n} \cdot \phi_{M_0,n}(x) + \sum_{m=M_0}^{M_{max}} \sum_n d_{m,n} \cdot \psi_{m,n}(x) \quad \textbf{(6.16a)}$$

with

$$c_{M_0,n} = \frac{1}{N} \cdot \sum_{i=1}^{N} y_i \cdot \tilde{\phi}_{M_0,n}(x_i); \; d_{m,n} = \frac{1}{N} \cdot \sum_{i=1}^{N} y_i \cdot \tilde{\psi}_{m,n}(x_i) \quad \textbf{(6.16b)}$$

For orthogonal wavelets, (6.16) is equivalent to (6.13).

Density estimators

Estimation of density functions can be made with the Parzen-Rosenblatt estimator defined as

$$\hat{f}(x) = \frac{1}{N \cdot h^d} \cdot \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right) \quad \textbf{(6.17)}$$

with d the dimension.

The naive density estimator is obtained by using the uniform kernel. At one dimensional, this corresponds to forming the curve histogram. The same kernels as for regression may be used, for instance splines, gaussian or quadratic kernels. Wavelet-based methods for density estimation are quite similar to wavelet-based regression methods. The density function f_{dw} is approximated as a weighted sum of scaling functions and wavelets:

$$\hat{f}_{dw}(x) = c_{M_0,n} \cdot \phi_{M_0,n}(x) + \sum_{m=M_0}^{M_{max}} \sum_n d_{m,n} \cdot \psi_{m,n}(x) \quad \textbf{(6.18)}$$

The coefficients may be approximated by using their empirical values:

$$c_{M_0} = \frac{1}{N} \cdot \sum_{i=1}^{N} \tilde{\phi}_{M_0,n}(x_i) \quad \textbf{(6.19a)}$$

$$d_{m,n} = \frac{1}{N} \cdot \sum_{i=1}^{N} \tilde{\psi}_{m,n}(x_i) \quad \textbf{(6.19b)}$$

Wavelet denoising methods

Wavelet denoising methods already to the standard signal processing toolbox and have found a large range of applications. Wavelet denoising methods are classified into two large categories, thresholding and shrinkage. In thresholding, a wavelet coefficient is set to zero if its value is below a given threshold value. Nonlinear denoising methods were developed by Donoho (1994). In the hard

thresholding method, all coefficients below a certain threshold are set to zero. In the soft thresholding method, the wavelet coefficients are reduced by a factor α. The coefficients after thresholding are given by the expression:

$$\hat{w}_{m,n} = \text{sign}(\hat{d}_{m,n}) \cdot \max(0, |d_{m,n}| - \alpha) \qquad \textbf{(6.20)}$$

Practically, shrinkage corresponds to multiplying some wavelet coefficients by a level-dependant positive factor smaller than one. From the theoretical point of view, an interpretation can be given to linear shrinkage methods. A wavelet decomposition use perfect reconstruction filters, consisting of a low-pass filter T_{low} and a high-pass filter T_{high} fulfilling the power complementarity condition: $|T_{low}|^2 + |T_{high}|^2 = 1$. Linear shrinkage is equivalent to replacing the two filters with two new filters T'_{high}, T'_{low} with $|T'_{low}|^2 = \alpha |T_{low}|^2$ and $|T'_{high}|^2 = \alpha(1-\varepsilon) \cdot |T_{high}|^2$. The coefficients are such that $\alpha<1$ and ε is generally small and positive. Roughly, linear shrinkage results into damping the high-frequencies more than the low-frequency signal components. A linear shrinkage method has been proposed by Antoniadis (1994). The method generalizes smoothing splines to wavelet. The minimizer of the expression

$$\|f_w - f\|^2 + \lambda \cdot ((\sum_n (c_{m,n})^2)^{0.5} + (2^{j \cdot s} \cdot \sum_{m=J_0}^{\infty} (\sum_n (d_{m,n})^2))^{0.5} \qquad \textbf{(6.21)}$$

is searched for on [0,1]. The solution to the variational problem is

$$\hat{f}_w(x) = c_{M_0} \cdot \phi_{M_0,n}(x) + \sum_{m=M_0}^{M\max} \sum_n \hat{\beta}_{m,n} \cdot \psi_{m,n}(x) \qquad \textbf{(6.22)}$$

$$\textbf{with } \hat{\beta}_{m,n} = \frac{d_{m,n}}{1 + \lambda \cdot 2^{2sm}} \qquad \textbf{(6.23)}$$

The wavelet coefficients are reduced by a factor proportional to the level m. This method is generally referred to as the linear shrinkage method. Nonlinear shrinkage methods have been developed by a number of authors (Abramovich, 2000).

Fuzzy wavelet estimators

Fuzzy wavelet estimators within the framework of the singleton model

Wavelet estimators based on orthogonal wavelets do not have a linguistic interpretation. A linguistic interpretation of kernel estimation requires that the regression curve is approximated as a weighted sum of functions partitioning the unity and having only positive values. A solution to that problem consists of using biorthogonal wavelet estimators. Symmetric scaling functions that can be interpreted as membership functions are taken as local functions. Except for the Haar wavelet, no orthogonal scaling function has positive values everywhere and is symmetric. So the solution consists of taking scaling functions associated to biorthogonal wavelets. The fuzzy wavelet estimator computes first the values of $\hat{f}_m(x_n)$ on a regular grid with the dual function $\tilde{\phi}_{m,n}$ as kernel:

$$\hat{f}(x) = \frac{\sum_{i=1}^{N} \tilde{\phi}(\frac{x_i - x}{h}) \cdot y_i}{\sum_{i=1}^{N} \tilde{\phi}(\frac{x_i - x}{h})} \qquad (6.24)$$

For symmetric functions, the expression simplifies greatly if the function $f(x)$ is only estimated at regularly spaced points (fig.6.1). In this case, one obtains

$$\hat{f}(k \cdot h) = \frac{\sum_{i=1}^{N} \tilde{\phi}(\frac{x_i - k \cdot h}{h}) \cdot y_i}{\sum_{i=1}^{N} \tilde{\phi}(\frac{x_i - k \cdot h}{h})} \qquad (6.25)$$

with k an integer.

For uniformly distributed input data, $(\sum_{i=1}^{N} \tilde{\phi}_{m,n}(x_i) \cdot y_i) / \sum_{i=1}^{N} \tilde{\phi}_{m,n}(x_i)$ is a good approximation of $< f(x), \phi_{m,n} >$. Eq.(6.26) furnishes therefore an estimation of $\hat{c}_{m,n}$ in $f_m(x) = \sum \hat{c}_{m,n} \cdot \phi_{m,n}(x)$:

$$\hat{c}_{m,n} = (\sum_{i=1}^{N} \tilde{\phi}_{m,n}(x_i) \cdot y_k) / \sum_{i=1}^{N} \tilde{\phi}_{m,n}(x_i) \qquad (6.26)$$

In the limit of infinitely many points, $\hat{c}_{m,n}$ equals $c_{m,n}$, if the function $f(x)$ is regular enough. The second step of the algorithm is simple. The Watson-

Nadaraya estimator is taken to interpolate between the points on the regular grid (fig. 6.2). The function $\phi_{m,n}$ is used this time as kernel.

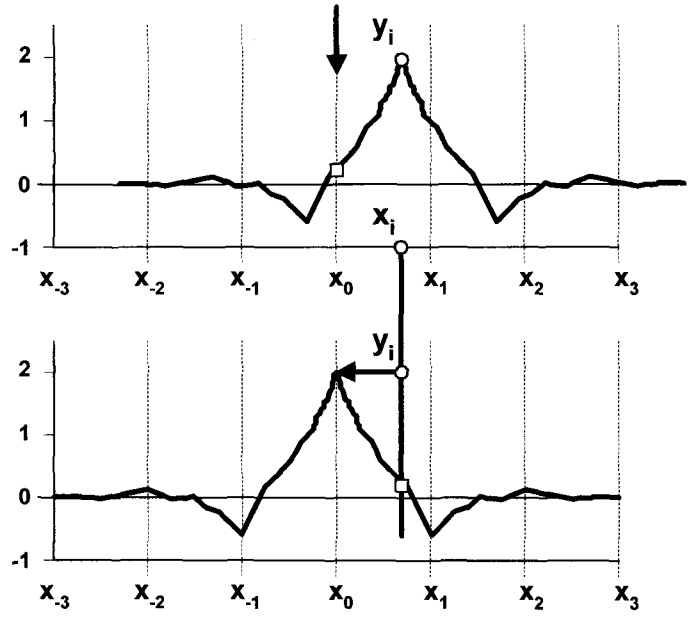


Figure 6.1: Multiresolution spline estimators use dual spline estimators based on the functions $\tilde{\phi}_{m,n}(x)$ to estimate the coefficients $\tilde{c}_{m,n}$ in. $f_m(x) = \sum \hat{c}_{m,n} \cdot \phi_{m,n}(x)$.

In summary, the fuzzy wavelet estimator is given by

$$\hat{f}_m(x) = \sum_n \left(\left(\sum_{i=1}^{N} \tilde{\phi}(2^m x_i - x_n) \cdot y_i \right) / \left(\sum_{i=1}^{N} \tilde{\phi}(2^m \cdot x_i - x_n) \right) \cdot \phi_{m,n}(x) \right) \tag{6.27}$$

with x_n on a regular grid .

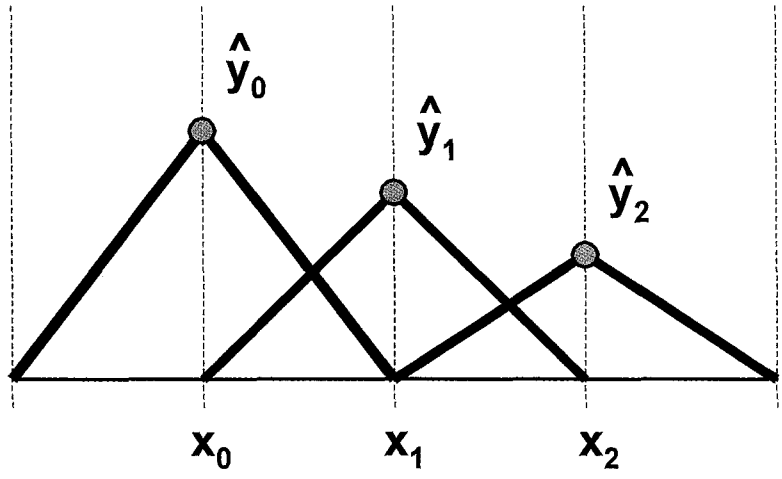


Figure 6.2: After having computed the coefficients $\hat{c}_{m,n}$, a function f(x) is estimated with $f_m(x) = \sum \hat{c}_{m,n} \cdot \phi_{m,n}(x)$.

Multiresolution fuzzy wavelet estimators: application to on-line learning

The above method is easily generalized to a multiresolution (fig. 6.3) by using an ensemble of estimators. Also a fuzzy interpretation can be given if, for instance, splines are taken as scaling functions. In the multiresolution setting, the choice of appropriate rules is carried out by using a method quite similar to the one implemented in fuzzy wavenets. The estimation of the surface at one level of resolution is compared with the estimation at one lower level of resolution. This is done by decomposing the approximation coefficients with the low-pass filter associated to the fast wavelet decomposition algorithm. In order to validate the coefficient $\hat{c}_{m,n}$, two validation conditions are necessary:

$$\left| \hat{c}_{m,n} - \sum_k p_{k-2n} \cdot \hat{c}_{m+1,k} \right| < \Delta \qquad \textbf{(6.28)}$$

with the filter coefficients p corresponding to the low-pass decomposition coefficient for splines. Further, one requires also that

$$\left| \sum_{i=1}^{N} \tilde{\phi}_{m,n}(x) \right| > T \qquad \textbf{(6.29)}$$

to prevent divisions by a very small values.

In many on-line problems, the signal processor is capable of making some computations but has too little memory to store many datapoints. Under these conditions, most cross-validation methods are not implementable and the above

method is very appropriate (For reviews on wavelet-based estimators see Abramovich (2000) or Antoniadis (1997)).

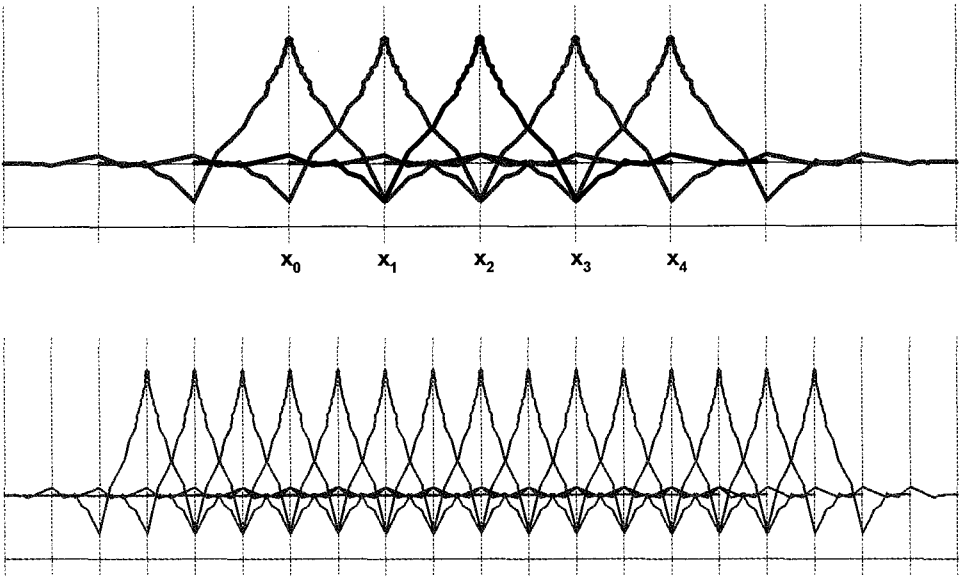


Figure 6.3: Multiresolution fuzzy wavelet estimation is carried out by using an ensemble of estimators.

The strength of the above approach is that the computation of a coefficient $\hat{c}_{m,n}$ requires only the storage of two values: the denominator and the nominator in (6.27).

A probabilistic approach to fuzzy-wavelet

Fuzzy logic is far from presenting a unified picture. Not only do the different approaches differ at the technical level, but also several quite different conceptions of fuzzy logic coexist and sometimes cross-fertilize! The view has been sometimes expressed that fuzzy logic is a reformulation of kernel estimation theory. This view is by far too simplistic even if kernel estimators are excellent to develop fuzzy systems from data. After this warning, let us see how to develop fuzzy systems from data with multiresolution approaches. A linguistic interpretation of kernel estimation is given by associating a linguistic term to each function $\phi(x-n)$. Watson-Nadaraya estimators are often the method of choice.

In the past chapters, it has always been assumed implicitly, that the data points in $\Re^m$ are close to an underlying hypersurface. A good description of a surface y=f(**x**) was searched for. After learning with the fuzzy-wavelet technique,

the hypersurface f(**x**) is approximated by a number of linguistic rules of the form:

R_i : if **x** is **A** then $y = b_i$ with b_i a real number, and $\mathbf{x}=(x_1,...,x_{m-1})$.

The membership functions are chosen among the family of scaling functions that have the property to be symmetric, everywhere positive with a single maximum. This family includes among others some radial functions, splines, tensor products of splines and some radial functions. In this approach, only the input space is granulized. Using splines as membership functions for the output space, the product as AND operator, and a center of gravity defuzzification, the fuzzy rule can be written under the form: if **x** is A then y is B. The transformation of the rule is a kind of pseudo-granularization of the output space. The supports of the membership functions are chosen independently of the distribution of points. If both the resolution of the input and output space must be adaptively determined, then a different approach must be taken to determine the right resolution of both input and output membership functions. Consider a set of data points in $\Re^m$: $\mathbf{x}=(x_1,...,x_{m-1}, y)$ and fuzzy rules of the form:

x is A (C) **(6.30)**

If one wants to privilege the variable y, (6.30) can be written equivalently under the form

If x is A then y is B (C) **(6.31)**

The confidence levels C can be computed from an estimation of the probability density function F(**x**,y). With this approach, any defuzzification method is allowed. The probability density function is estimated with a biorthogonal wavelet density estimator.

PART VII DEVELOPING INTELLIGENT PRODUCTS

This page is intentionally left blank

7. Developing Intelligent Products

The word *intelligence* applies today not only to human performance, but is used in connection to a large number of product features. So-called *intelligence* can be found in a very large variety of products starting from cameras to automatic guidance systems in helicopters. What is generally meant here by intelligence is a different type of intelligence than the human intelligence, that is often referred as computational intelligence. In this section, we will explain how a product can be made computationally more intelligent by using soft computing techniques.

From the industrial point of view, the success of fuzzy logic has much to do with the fact that it permits the translation of knowledge into linguistically expressed mathematical expressions. The translation of this knowledge into a fuzzy system is not as simple, as it had been originally claimed at the beginning of the *fuzzy wave*. Development engineers have learned that the fine tuning of a fuzzy system can be quite time-consuming if the number of implemented rules describing the system is large. Therefore, probably the most significant development in the field has been the appearance of new methods to train fuzzy systems automatically. There are essentially three main soft computing methods to train fuzzy systems: neural networks, genetic algorithms and multiresolution based techniques. Fuzzy logic is well suited to fusing information from data and human experts. It is generally easier for a human operator to express his knowledge under the form of linguistically rules than under a purely mathematical form. Fusing the two sources of information and testing the compatibility between the fuzzy rules formulated by a human expert and from a databank are two of the main issues in learning.

Transparency

The main motivation behind using fuzzy rules is to keep the human factor within the loop. For self-learning systems without any human supervision, better methods than fuzzy logic can be applied. Fuzzy learning is justified when a human expert is part of the modeling or the validation process. Fuzzy learning provides methods to fuse human expert knowledge to experimental knowledge generally under the form of measurement points. It is generally easier for a human operator to express his knowledge under the form of linguistic rules than under a purely mathematical form. Fuzzy logic is well suited to fusing information from human experts and databanks. Depending on the confidence

that the operator has on human expertise, the rules generated by the human expert, can be integrated as hard or soft constraints. During automatic learning, data in contradiction with hard constrained rules are simply eliminated. Soft constrained rules can be included under different forms. For instance, a number of data points can be generated from the rules formulated by the expert and added to the experimental data (fig. 7.1). Another possibility is to look for a compromise solution between the rules obtained from experimental data and the rules formulated by human experts.

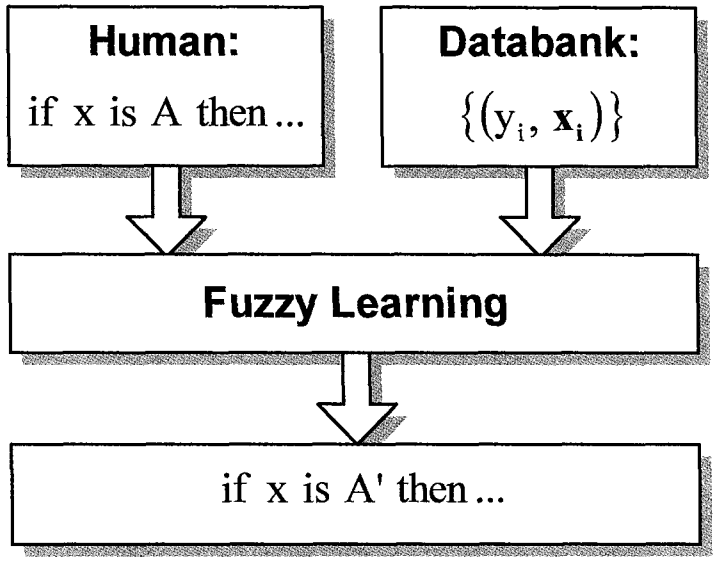


Figure 7.1: The linguistic formulation of fuzzy logic simplifies the fusion of human expert rules with information extracted of a databank. The transparency of the resulting rules is of crucial importance. Transparency has been a main motivation behind the development of learning methods combining multiresolution analysis to fuzzy logic.

Some of the most successful fuzzy learning methods are formally equivalent to a classical modeling technique. For instance, spline modeling can be used to develop fuzzy systems within the Takagi-Sugeno formalism. The central contribution of the fuzzy approach lies in a series of methods that have as goal to ensure either an intuitive interpretation of the rules by human experts or at least an easy inclusion of expert knowledge to the experimental knowledge under the form of a databank. As mentioned above, fuzzy logic cannot be separated from its linguistic context. Therefore, a good compromise must be found between accuracy, transparency and complexity. Let us discuss that aspect of fuzzy modeling within the context of function approximation.

The first limitation of fuzzy modeling is that the dictionary of local functions for modeling is restricted to functions having a linguistic interpretation.

Functions with negative values do not have a simple linguistic interpretation. A fuzzy approximation is therefore sub optimal what the accuracy and complexity are concerned.

For a given accuracy, the lowest complexity description of a fuzzy system is achieved by optimizing the shape and position of the membership functions. For a good interpretability, the constraint that the membership functions form a partition of unity is very often necessary. This condition is rarely fulfilled by the lowest complexity solution, therefore adding that constraint results into an increase of the complexity of the solution.

Multiresolution-based fuzzy methods furnish a new approach to the problem of transparency and linguistic interpretability. Linguistic interpretability is included per design by using pre-defined membership functions forming a multiresolution. Membership functions are chosen among a dictionary and describe terms such as *very small* or *large* that do not change during learning. A fuzzy system developed with that method consists of a number of rules using membership functions with clear linguistic interpretations. Linguistic interpretability and transparency are two slightly different concepts. Linguistic interpretability refers to the quality of rules to have a natural linguistic interpretation, such as *if temperature is low then heater is on*. Transparency describes the quality of a system to be understood by the human operator. A preliminary condition to transparency is a natural linguistic interpretability of rules. A second condition is that the number of rules and the number of different levels in a hierarchical fuzzy system is still manageable by human experts. In other words, the results should be put under a form, that is not too complex and linguistically transparent to give enough insight into the results.

We have seen that complexity and accuracy are often contradictory. In many systems, transparency is lost if a high accuracy is required. A possible solution consists of using two fuzzy modeling results, the first one at a low accuracy that preserves transparency for human validation and a second very accurate model for computational validation. Using a fuzzy-wavelet approach, the high resolution approximation can be obtained from the low resolution description by adding a number of rules that represent small corrections to the rules.

Transparency can be significantly increased by removing unnecessary terms through fusion of rules or by using constructive methods. The fusion of rules in wavelet-based methods is simple as the membership functions form a multiresolution. A low resolution approximation can be computed from the approximation coefficients at high resolution using the fast wavelet decomposition. The energy contained in the wavelet coefficients characterizes the error introduced by lowering the resolution. The resolution can be chosen locally based on conditions on the maximal tolerated local error.

Man, sensors and computer intelligence

The multiresolution learning methods presented in the last two parts have been implemented during the development of several real world projects. Implementations were in the domain of sensorics and more precisely in the domain of fire detection. Sensorics is especially interesting, as its represents a typical case of distributed intelligence. Learning starts from a databank containing knowledge on fires and deceiving phenomena, under the form of signal recording, obtained through field testing and laboratory (fig. 7.2).

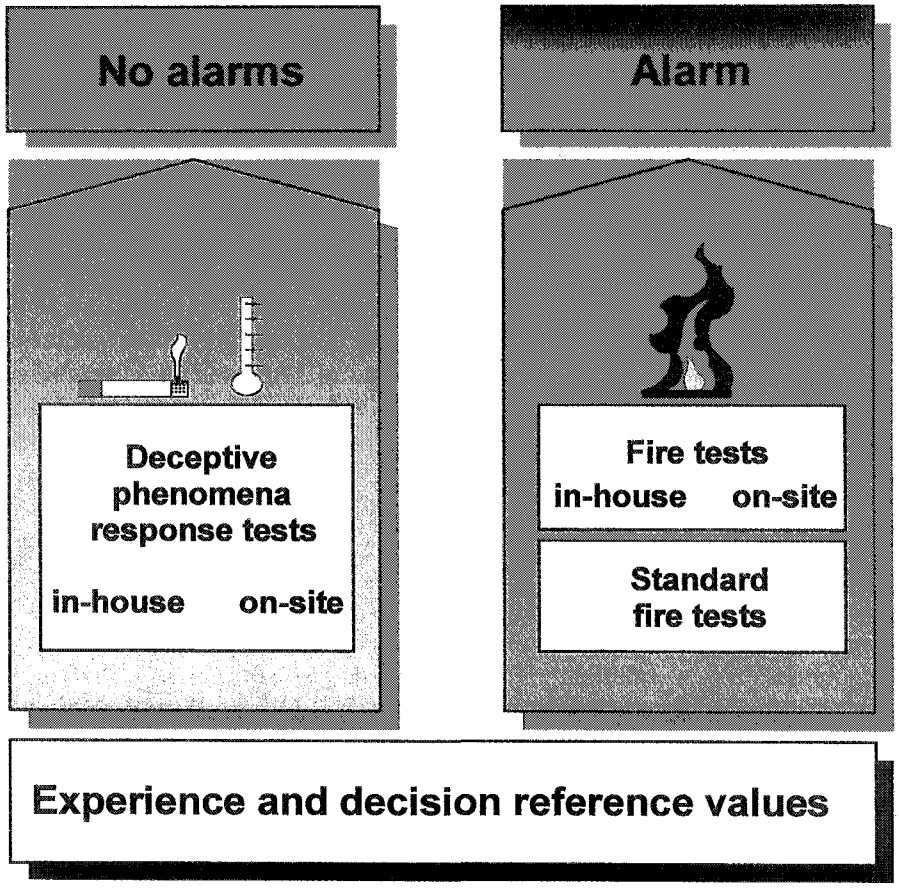


Figure 7.2: The development of intelligent fire detectors is carried out by extracting information from large databanks containing measurements of fire and non-fire situations.

An alarm surface separating the alarm from the non-alarm conditions is extracted from data for fire and non-fire situations. The alarm surface is put under the form of a number of fuzzy rules.

The different rules are then checked against human experts by using a computer assisted development tool. The program checks for the consistency between experimental data and expert know-how. The program proposes also compromise solutions in case of incompatibility.

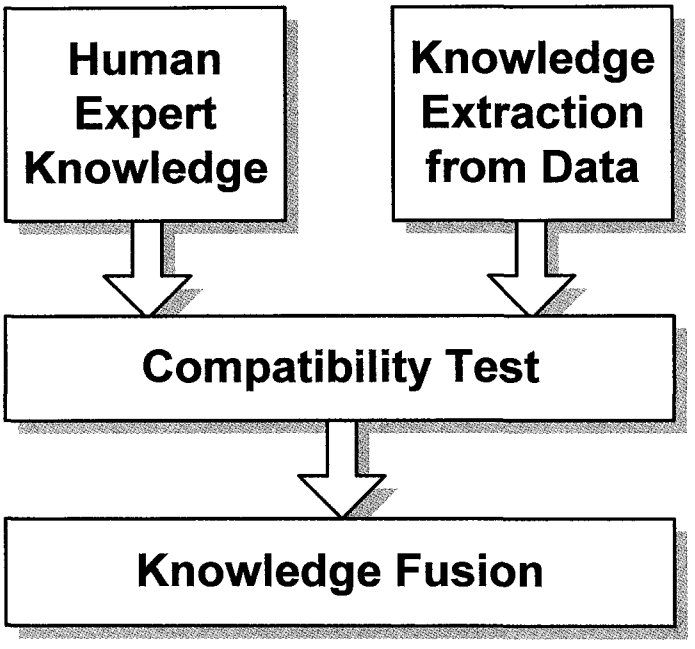


Figure 7.3: The different rules are checked for compatibility before knowledge from different experts is fused.

One of the most interesting stage in the development process corresponds to the fusion of the knowledge extracted from data to the already existing expert knowledge. The compatibility between the different experts must be checked prior to knowledge fusion (fig. 7.3). The comparison of expert knowledge is not too difficult. It relies indeed on a geometrical interpretation of knowledge, based on the fact that a series of linguistic fuzzy rules can always be expressed as surfaces (in n-dimensions surfaces for n input variables). In this geometrical approach information processing from different sources corresponds to comparing surfaces, study how they complete eachother and how they overlap. Therefore, the compatibility between fuzzy rules generated by human operators and from data can be assessed by comparing the corresponding hypersurfaces.

Suppose that we have two experts that express linguistically their knowledge on a certain process under the form of a number of fuzzy rules. The first possibility is that the two experts have knowledge on two different parts of the process. In this case their knowledge complete eachother as shown in fig. 7.4a. This corresponds mathematically to fusing two different surfaces into a single

surface. If the two experts have knowledge on the same part of the process, their information might partially contradict. In this case, the computer might start dialoguing with the experts by making proposals to reconcile the contradicting information. This is done by deforming the surfaces corresponding to the different expert knowledge with a minimum of pre-defined operations, such as then surfaces become comparable (Thuillard, 1998b). Each deformation results into a penalty and the surface minimizing the total penalty is proposed as a compromise solution. The deformed surface can then be translated into linguistic expressions, that are submitted to the experts. The process can be iterated till an agreement between the experts is reached.

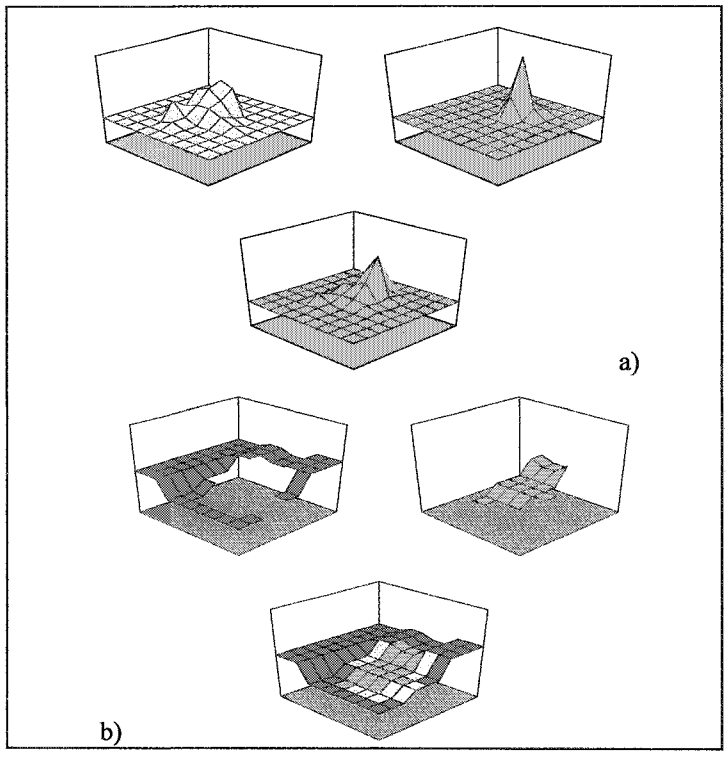


Figure 7.4: a) The knowledge of two experts can be analyzed by a deformable template method using as input the fuzzy-wavelet coefficients describing the control surface. b) The partial knowledge of two experts can be easily summarized by using approximation methods based on multiresolution analysis.

Our approach is by some aspects similar to wavelet-based deformable templates techniques. Expert knowledge is expressed as a surface with as few spline coefficients as reasonable. The expert knowledge' surfaces are deformed using 3 operations on spline functions:

Translation by n an integer

Dilation by 2^m m integer

Change the value of a spline coefficient

A penalty corresponds to each deformation. Given two deformed surfaces T_1 and T_2,

the score Sc is defined by:

$$\mathbf{Sc} = \max(\sum_{\mathbf{x}} (T_1(\mathbf{x}) - T_2(\mathbf{x}))^2, K) \cdot \pi(T) \qquad \textbf{(7.1)}$$

with $\pi(T)$ the total penalty function and K a constant. The best compromise between the penalty function associated to surfaces' transformations of two original surfaces S_1 and S_2 and their dissimilarities characterized by $\max(\sum_{\mathbf{x}} (T_1(\mathbf{x}) - T_2(\mathbf{x}))^2, K)$ is obtained by minimizing the score function Sc with for instance evolutionary-based techniques or Monte-Carlo methods. The result is then interpreted with the following scheme:

1) If $\max(\sum_{\mathbf{x}} (T_1(\mathbf{x}) - T_2(\mathbf{x}))^2, K) = K$ then the synthesis between the two experts can be expressed by the average value:

$T_a(\mathbf{x}) = (T_1(\mathbf{x}) + T_2(\mathbf{x}))/2$.

If $\max(\sum_{\mathbf{x}} (T_1(\mathbf{x}) - T_2(\mathbf{x}))^2, K) > K$ then one considers that the two experts contradict eachother. The reasons for the contradiction can be searched among several possibilities:

One expert is wrong

Expert's knowledge is not described with sufficient precision

Local model failure.

In summary, the deformable surface approach permits to reconcile somewhat different experts' knowledge and to detect problematic regions.

Learning is dynamic and field testing is a central part in the development process. As the experts' knowledge expands, new field testing locations are added to collect information on missing parts of the control surface. Field testing is also used to validate the rules.

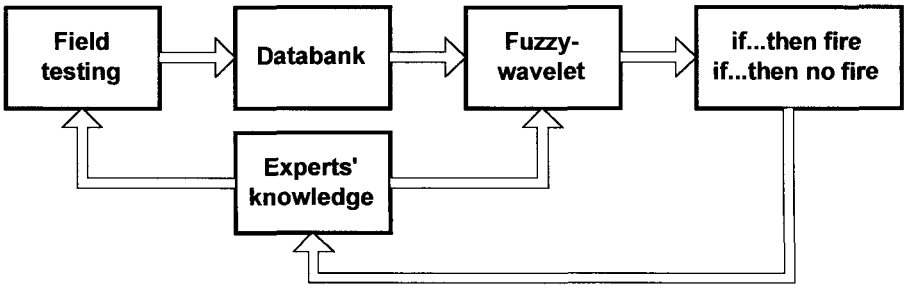


Figure 7.5: Field testing is an essential part in development. Field testing is used to collect new information on the sensors and to validate the fuzzy rules describing the alarm surface.

A considerable advantage of such a working process is that it takes advantage of the available computing power, while always keeping the human in the loop. The human experts are not run over by the computer (fig. 7.5). Since at the end, the information is under a linguistic form, human cross-checking of the computer results is possible. Such control of the computer results by human experts is absolutely necessary, as one should never forget that the results furnished by the computer can only be as good as the data supplied to it.

In conclusion, computer assisted development of *intelligent* products is today a reality. This has been principally possible by recognizing two things. First, linguistic expressions are much easier to process by a pool of experts than mathematical expression. This is particularly true in today's multi-disciplinary working environment. Second, computing power both during the development of a product and in the product itself is still today a serious limitation. Soft computing is an answer to this problem. Excellent results are obtained in many industrial projects by giving up some unnecessary precision. We have been using computer assisted development programs to compare and fuse the information from the different sources during the development of several products (optical beam detector, flame). The experience was very positive and the method is applied now routinely to new developments.

Constructive modeling

Constructive methods can be also be used to reduce the complexity of the model. Adaptive Spline Modeling of Observational Data (ASMOD) has been used with success to many modeling problems, such as ship docking, or helicopter guidance (Harris,1999a). The ASMOD scheme (Adaptive Spline Modeling of Observation Data) has been proposed by Kavli (1994) as an answer to the *curse of dimensionality*. With ASMOD, the model structure is constructed iteratively

through successive model refinements. The resulting model is a linear sum of several low-dimensional sub-models. Let us describe first ASMOD within the setting of neurofuzzy modeling. There are a number of ways of refining the model. A new input variable can be added to the system. Multivariate submodels can be formed by tensor products. New basis functions can be formed by adding new knots. The performance measure is chosen such as to balance the increase complexity associated with the refinement and the reduction of the MSE. The Bayesian and Akaike' s information criterion have been proposed to measure the compare the performance of the different refinements in order to choose the most appropriate one.

ASMOD does also apply very well to fuzzy-wavelet modeling. The model refinement starts with a very simple model, typically a one-dimensional submodel. Each refinement step corresponds to choosing among 3 methods, the refinement procedure decreasing the most the information measure. Two refinement procedures are part of the ASMOD scheme: adding new one-dimensional sub-models and forming tensor products submodels. The third refinement procedure consists of splitting a membership function into memberships functions at one higher level of resolution.

This page is intentionally left blank

PART VIII GENETIC ALGORITHMS AND MULTIRESOLUTION

This page is intentionally left blank

8. Genetic Algorithms and Multiresolution

The main purpose of this part is to show some important connections between genetic algorithms and multiresolution. In problems using binary coding of integers, genetic algorithm may be related to multiresolution analysis. A first major step in that direction was made by Bethke (1981) with the introduction of Walsh partition functions in the field of genetic algorithms. Important insights into the working of genetic algorithms, and in particular on the building block hypothesis have been unraveled using Walsh functions. For some genetic algorithms, the building block hypothesis is better captured by Haar wavelets than Walsh partition functions. As an example, a simple wavelet-based genetic algorithm is constructed and discussed within the framework of Haar wavelets. The algorithm uses a single operator that shares some of the features of the crossover and the mutation operators. The wavelet-based genetic algorithm is methodologically interesting. The wavelet-based genetic algorithm is simple enough to furnish some analytical results, while preserving some essential features of genetic algorithms.

The standard genetic algorithm

The standard algorithm is certainly not the algorithm of choice in applications nowadays. It is nevertheless consistently referred to as an important prototype algorithm for genetic algorithms (Goldberg, 1991). The standard genetic algorithm uses strings described by a binary alphabet B={0,1}to encode possible solutions to an optimization or search problem. At each generation, a number of strings are taken as candidate solutions. The different strings form what is called a population. At each generation, strings are selected according to their fitness. Some strings are modified with the crossover and mutation operators before they are included in the next generation.

The standard genetic algorithm is often explained on the basis of the fundamental theorem of genetic algorithms. This theorem furnishes a bound to the probability of a schema H. A schema H is defined on the alphabet S={0,1,*}. The alphabet S corresponds to the alphabet B with the addition of the symbol *. The symbol * may represent *any symbol in B*. As an example, consider the schema H= (1,0,*,0). This schema contains the two strings (1,0,0,0) and (1,0,1,0).

Two operators are used in the standard genetic algorithm: crossover and mutation.

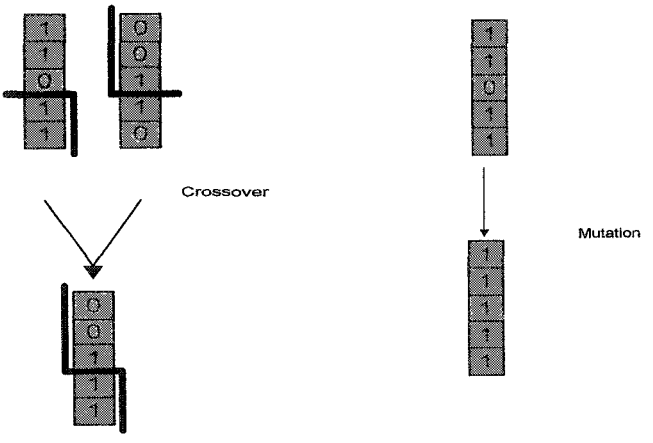


Figure 8.1: The standard genetic algorithm uses the crossover and the mutation operator.

The crossover operator splits two strings at a given point and exchanges one segment with the other string. The two new strings are made of one segment of the original string and a new segment. Crossover tends to preserve compact and short substrings.

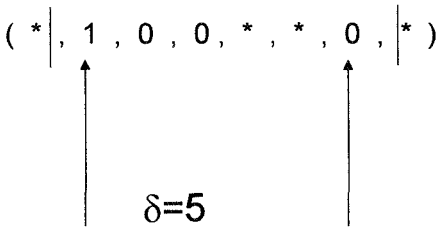


Figure 8.2: A schema consists of a string of symbols belonging to the alphabet S={0,1,*}. The useful length of the string δ is the distance (or number of positions) between the first and the last symbol in the schema belonging to the alphabet B={0,1}. The number of symbols in the schema belonging to the alphabet B is called the order O of the schema.

The smaller the useful length is, the smaller is the chance that a crossover will disrupt a schema (The useful length δ is the distance, or number of positions, between the first and the last symbol in the schema belonging to the alphabet B). In the above example (fig. 8.2), crossover will not disrupt the schema only if the splitting point is at one of the two positions given by the line. A crossing point at any other location may result into a disruption of the schema.

The mutation operator modifies one bit (also called allele by analogy to biology) in the string. A schema is disrupted by a mutation only if the symbol is different from * . The number of symbols in the schema belonging to the alphabet B is called the order O of the schema. Mutations tend to preserve low order schemata.

In the standard version of the genetic algorithm, the survival probability of a string s is chosen proportional to f(s)/f with f(s) the fitness of the string s and f the average fitness over all strings. The fundamental theorem of genetic algorithms states that the expectation of the number of instances of a given schema H, m(H, t), can be written under the form of the following inequality (Holland, 1975):

$$E(m(H,t+1)) \geq m(H,t)\cdot f(H)/f\cdot[1-p_c\cdot\delta(H)/(l-1)-p_m\cdot O(H)] \quad \textbf{(8.1)}$$

with p_c the crossover probability, p_m the mutation probability, f(H) the average fitness of the schema and f the average fitness over all strings.

Short schemata with an above average fitness and a short useful length will increase in number very rapidly, while high order schemata with below average fitness values will be rapidly destroyed. Genetic algorithms find, in many optimization problems, very good solutions by assembling parts of good solutions. This hypothesis is often called the building block hypothesis. Short, low-order schemata with above-average fitness combine to form better solutions. Despite their many successes, genetic algorithms are confronted to a number of important challenges. One of them is to find out which problems do fulfill the building block hypothesis. An interesting approach to this question has been furnished by the use of partition functions (Bethke, 1981; Horn, 1995; Roya, 1998). Using the Walsh partition functions, functions were constructed that do not fulfill the building block hypothesis (Goldberg, 1989). These so-called deceptive functions offer an interesting insight into the working of genetic algorithms.

Walsh functions and genetic algorithms

Walsh functions

Walsh functions are functions taking only a value of 1 or —1 on a support of length L. Figure 8.3 shows the Walsh functions with L=8.

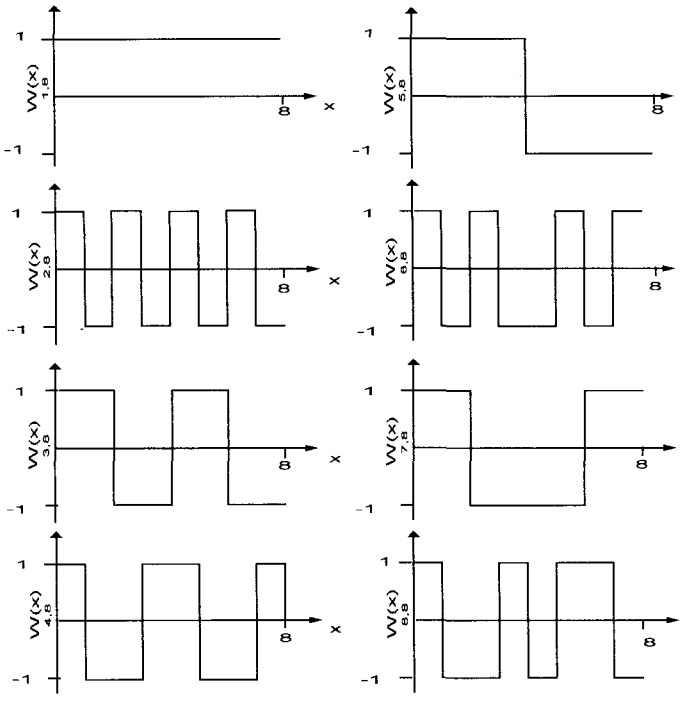


Figure 8.3: Walsh functions on a support of length L=8.

Walsh functions can be put under a matrix form. The Walsh transform W of a vector x is given by

$$W = M \cdot x \tag{8.2}$$

The matrix M is defined by the following expression:

$$M_{i,j} = -1^{bc(i,j)} \tag{8.3}$$

with bc(i,j) the number of 1 set in the string defined by the expression bit AND (i,j).

Walsh partition functions form an orthonormal basis. A vector can therefore be decomposed on the Walsh functions and reconstructed losslessly. The decomposition is given by

$$\omega_i = (M \cdot x)_i \tag{8.4}$$

The Walsh coefficient ω_i corresponds to the projection on the i^{th} Walsh partition function. The original vector can be reconstructed from the Walsh coefficients:

$$x = 1/(i_{max} + 1) \cdot (\sum_i \omega_i \cdot M_i) \qquad \textbf{(8.5)}$$

Example:

The Walsh basis functions for i,j ={0,1,2,3} is:

$$M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & (-1) & 1 \end{pmatrix}$$

Let us verify this with (i=2,j=3, circle). The expression bit AND(2,3) is $bit\ AND[(1,0),(1,1)] = (1,0)$. It follows that the number of 1 in the bit AND expression is bc(i,j)=1 and M(2,3)= -1.

Consider as an example the decomposition of the vector x=(1,0,1,0)

The Walsh coefficients ω_j (j=0;...,3) corresponding to the projection on the different Walsh functions are given by $\omega_i = (M \cdot x)_i$. One obtains ω_0= 2; ω_1= 2; ω_2= 0 ; ω_3=0 and it is easily verified that ¼(2 M_1+ 2 M_2)= x with M_i the i^{th} column of M corresponding to the i^{th} Walsh partition function.

An alternative description of the Walsh functions using the formalism of wavelet packets

Walsh functions are related to Haar wavelets. Walsh functions can be deduced from the multiresolution Haar decomposition independently from (8.3). The next paragraph introduces the Walsh functions using the formalism of wavelet packets. Recall that the dyadic wavelet decomposition can be represented as a tree composed of a cascade of low-pass and high-pass filters. Figure 8.4 shows an example giving the tree representation of a 3 levels wavelet decomposition.

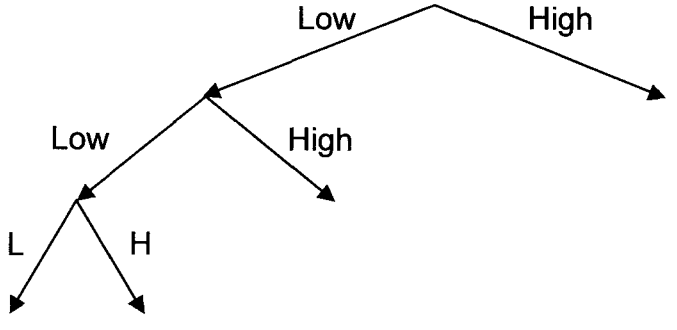


Figure 8.4: Tree representation of a wavelet decomposition.

The wavelet decomposition can be generalized to wavelet packets. A wavelet packets decomposition may be represented by a subtree of the complete decomposition tree. Recall that the complete decomposition tree is given by all possible dyadic decompositions of a signal with two filters fulfilling the power complementarity condition. Figure 8.5 shows the complete tree for a 3-levels decomposition.

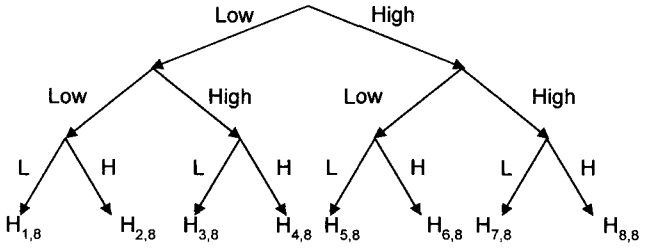


Figure 8.5: Full decomposition tree.

The Walsh wavelet packets are associated to the complete decomposition tree for the Haar function. At the J [th] level of decomposition, each function, on which the signal is projected, corresponds to one basis Walsh function of support $L=2^J$.

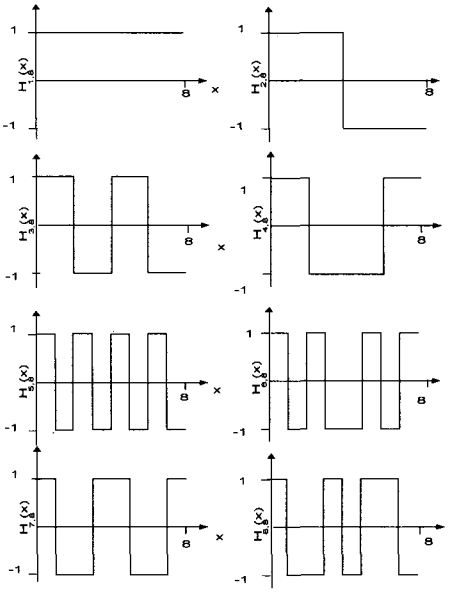


Figure 8.6: Functions corresponding to the end nodes of the complete decomposition tree in fig. 8.5 with L=8.

Assume a signal of length L and a full decomposition tree using a Haar wavelet as mother wavelet. The L functions of support L corresponding to the end nodes of the full tree are the Walsh functions. Figure 8.6 shows the functions corresponding to the end nodes for J=3. One can verify that the Walsh functions are the same as in fig. 8.3. This holds for Walsh functions of any order. From the construction of the Walsh functions in terms of a cascade of filters fulfilling the power complementarity condition, it follows that a signal can be decomposed with Walsh packets without any loss of information. Starting from the coefficients at the last level of decomposition, the original signal can be reconstructed perfectly. A vector of length L can be projected onto the Walsh basis function of support L. The reconstruction proceeds without any error.

On deceptive functions in genetic algorithms

Binary coding permits to transform a number into a string. The integer i is coded as a L bits string in base 2: $<a_1,...,a_L>$ with

$$i = \sum_{k=1}^{L} a_k \cdot 2^{L-k} \qquad \textbf{(8.6)}$$

Assume that a fitness function $f(a_1,...,a_L)$ can be associated to all strings $(a_1,...,a_L)$. The average fitness value over all strings f(*,...,*) is given by the expression:

$$f(*,...,*) = 1/2^L \cdot \sum_{\text{all strings}} f(a_1,...,a_L) \qquad \textbf{(8.7)}$$

The above expression can also be given in terms of the first Walsh function:

$$f(*,...,*) = 1/2^L \cdot \sum_{\text{all strings}} f(a_1,...,a_L) \cdot H_{1,L} \qquad \textbf{(8.8)}$$

Similarly, the average fitness function of strings of the form (0,*,...,*) is

$$f(0,*...,*) = 1/2^{L-1} \cdot \sum_{\text{all strings}} f(a_1,...,a_L) \cdot (H_{1,L} - H_{2,L}) \qquad \textbf{(8.9)}$$

Figure 8.7 shows an example for the schema (0,*,*). Additional terms are included in the Walsh sum as schemata become increasingly specific. Recalling that the standard genetic algorithm tends to preserve a low order schema of high average fitness, one understands that the standard genetic algorithm is *guided* in its search by low order schemata. A genetic algorithm may be deceived by a function having low order schemata of high average fitness, containing no optimal solution. Problems that are intrinsically difficult for genetic algorithms have been designed using the Walsh functions (Goldberg, 1989, 1991). For such

problems, the building block hypothesis does not hold and genetic algorithms are generally less better than a random search.

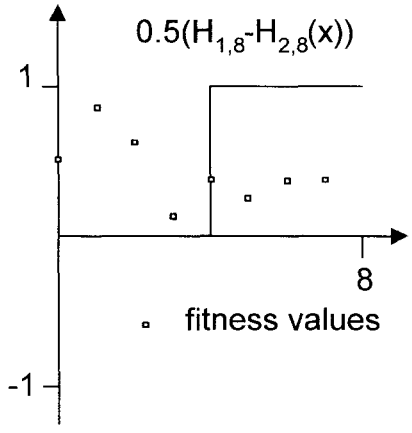


Figure 8.7: The average fitness of a scheme can be computed using the Walsh functions. Example for the schema (0,*,*).

As a final remark, one should mention that the fitness of the different schemata could have been discussed as well with other orthogonal basis than the Walsh functions. Many subtrees of the full decomposition tree in fig. 8.5 would have been as suited for discussing the building block hypothesis. In particular the Haar wavelet basis defined by the subtree in fig. 8.4 is more adapted to the discussion of some genetic algorithms. Such a genetic algorithm is presented in the next section.

Wavelet-based genetic algorithms

A very simple genetic algorithm is introduced in this section. The algorithm uses an operator that combines in one operator some of the main features of the crossover and the mutation operator. This genetic algorithm is conceptually interesting as mathematical expressions for the expectation of schemata of the type, $<a_1,...,a_i,*,...,*>$, can be computed exactly in terms of a Haar wavelet decomposition. Contrarily to the standard genetic algorithm, the proposed genetic algorithm is simple enough to be explained with multiresolution analysis. Despite its simplicity, the algorithm still captures the essence of the standard genetic algorithm.

As in the previous sections, binary coding is assumed. Figure 8.8 shows the binary coding of a one dimensional axis on which a function f is defined. The function $f(x_i)$ is a function of the integer i and can be interpreted as the fitness of

a_i. In the example of fig. 8.8 the value i is coded as $a_i=<a_1,...,a_4>$ with $i = \sum_{k=1}^{4} a_k \cdot 2^{4-k}$.

The multiresolution character of the schema formulation is shown in fig. 8.8. The fitness of a L bits schema $<a_1,...,a_k,*,*,*>$ corresponds to the average value of the fitness function on 2^{L-k} adjacent values. The average fitness can be computed using a 2^{L-k} dilated Haar function.

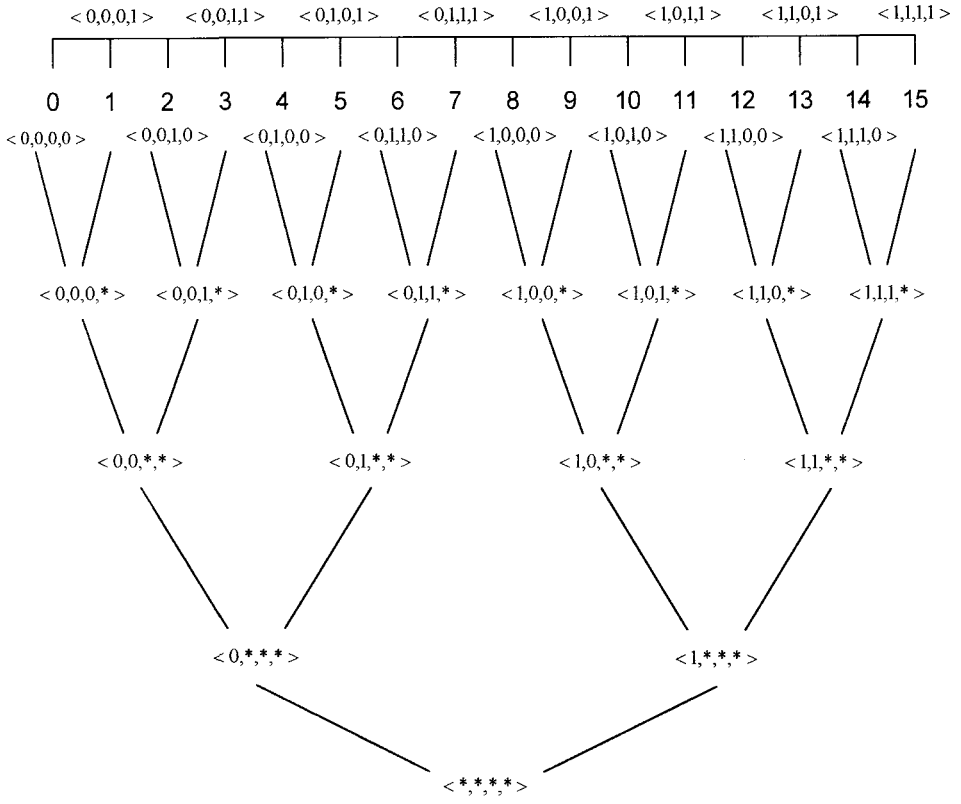


Figure 8.8: The multiresolution structure of schemata $H=<a_1,..,a_j,*,...*>$ is illustrated for L=4. Integers in the range $[0,2^L-1]$ are expressed in base 2 .

The wavelet-based genetic algorithm works as following. A string of fitness f_i is replicated on average $\gamma \cdot f_i$ times. Each string in the new generation is modified with probability P_m by an operator $O_m, (m = \{0,1,..., L\})$. The operator O_m replaces randomly the last m bits in the strings. If m=0 then all bits are randomly replaced,

while the string is kept unchanged for m=L (L is the number of bits in the string). The probability of splitting the string at position m is described by the values P_m, satisfying $\sum_{m=0}^{L} P_m = 1$.

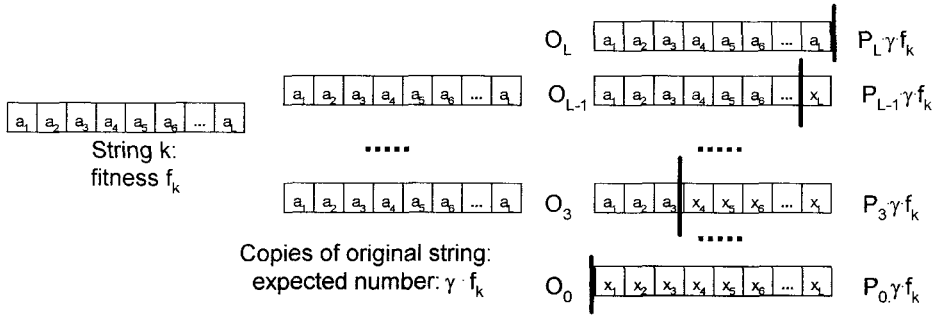


Figure 8.9: A simple wavelet-based genetic algorithm is designed by replacing the crossover and the mutation operators in the standard genetic algorithm by a single operator. The operator O_m replaces the last m bits randomly.

The wavelet-based genetic algorithm in the Haar wavelet formalism

Understanding when and why do genetic algorithms work well is a very difficult question. On the one hand, numerous successful applications of genetic algorithms tend to prove the efficiency of genetic algorithms in many situations. On the other hand, theoretical considerations have lead to a number of *no free lunch theorems* that show quite clearly that the blind application of genetic algorithms on a randomly chosen problem has a chance of less than 50% to be better than a random search. In order to escape that apparent paradox, much research has been done to understand emergent behaviors in genetic algorithms. This line has been followed by several researchers, and here one should mention in particular the work by Vose (2000). Vose has dedicated a book to a mathematical discussion of the so-called simple genetic algorithm. This model is very similar to the standard genetic algorithm from Holland. The main difference is that only one gene is kept after crossover. This small simplification makes the system solvable. The expectation of the number of strings at generation (p+1) can be computed from the expectation at step p by applying an operator G. The iterative application of the operator G defines the trajectory of the expected population. Different behaviors may be expected, depending on the number and the type of fixed points of the transform. If the matrix is irreducible and the coefficients of the matrix G are everywhere positive, then there exists a unique

eigenvector with positive coefficients. In the limit of an infinitely large population, the population follows the trajectory defined by G and converges towards a fixed population. If the vector G has several eigenvectors, then it can be shown, that in the limit of an infinite population, the system will spend most of the time in the vicinity of the fixed points. Despite it many successes, the study of the simple genetic algorithm did not contribute much to reducing the gap between applications and theory. The main reason is that the dynamics of the simple genetic algorithm is often so complex that it is difficult to draw general conclusions. As soon as the system has several positive eigenvectors, the relevance of the results to understanding small population dynamics is questionable. This problem leads us to consider a simpler algorithm, the wavelet-based genetic algorithm described in the previous section. The discussion of the wavelet-based algorithm furnish results that may serve as a guidance in the choice of the free parameters.

A lesson of past years is that it is necessary to define very precisely the coding method of the solutions as well as the genetic operators. The efficiency of the genetic algorithm does generally depend very centrally on the choice of the coding method (Reeves, 1999). In the following, the wavelet-based genetic algorithm will be explained using binary coding of integers. This multiresolution subdivision of the search space defines proximity relationships between the strings: Two close strings belong to the domain of definition of a well-localized, high-resolution Haar function, while distant strings belong only to the common domain of low-resolution Haar functions. Let us point out here, that most results, in the subsequent sections, are not specific to Haar wavelets. Many of the results apply to a large class of wavelet-based algorithms. These wavelet-based algorithms have in common that a proximity relationship based on multiresolution subdivision can be defined. A non-trivial example is given in annex B, based on a nonlinear wavelet construction.

In the two following sections, we will relate the wavelet-based genetic algorithm to wavelet theory and filtering theory in the framework of the infinite population approach. Assuming an infinite population simplifies the analysis. Due to the simpler nature of the algorithm in comparison to the simple genetic algorithm, the behavior of infinite populations permits to furnish a general framework to understand qualitatively the more difficult, and practically only relevant, case of finite population sampling.

The probabilities P_m determine the evolution of the population. The evolution equation of a schema of the form $H=a_1a_2...a_k*...**$ can be computed. Without limiting the generalization and in order to simplify the notations let us assume that $a_1=a_2=...=a_k=0$. The expected number of strings n_H with $H=a_1a_2...a_k*...**$ at generation p+1 is

$$E(n_{a_1a_2a_3\ldots a_k*\ldots**}) = \gamma\cdot[(\sum_{m=0}^{k-1}\sum_{i=1}^{N\cdot 2^{-m}} 2^{m-k}\cdot f_i\cdot n_i)\cdot P_m + \sum_{m=k}^{L}(\sum_{i=1}^{N\cdot 2^{-m}} f_i\cdot n_i)\cdot P_m] \qquad \textbf{(8.10a)}$$

For k=0, one obtains

$$E(n_{*\ldots***}) = \gamma\cdot(\sum_{i=1}^{N} f_i\cdot n_i) \qquad \textbf{(8.10b)}$$

in which n_i is the number of strings at generation p representing the solution with integer value i, f_i the corresponding fitness value and $N=2^L$. After some manipulations, one obtains the first main result, namely an expression relating the expectation of schemata to the detail coefficients $d_{k,j}$ of the Haar wavelet decomposition of $f(a_i)\cdot n(a_i)$ at the previous generation:

$$E(n_{a_1a_2\ldots a_k=0*\ldots**}) - E(n_{a_1a_2\ldots a_k=1*\ldots**}) = d_{k,j}\cdot\gamma\cdot\sum_{m=k}^{L} P_m \qquad \textbf{(8.11a)}$$

$$E(n_{*\ldots**}) = \gamma\cdot(\sum_{i=1}^{N} f_i\cdot n_i) = \gamma\cdot c_{1,1} \qquad \textbf{(8.11b)}$$

From (8.11a), one concludes that the expectation of $n_{a_1a_2a_3\ldots a_k=0*\ldots**} - n_{a_1a_2a_3\ldots a_k=1*\ldots**}$ is proportional to the corresponding Haar detail coefficient $d_{k,j}$ of $fn(a_i) = f(a_i)\cdot n(a_i)$.

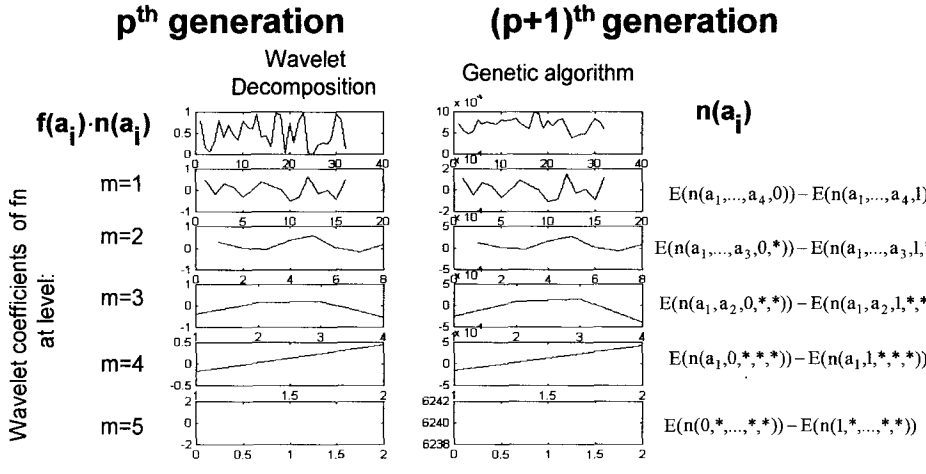


Figure 8.10: The expectation of a schema $<a_1,\ldots a_j,*,\ldots*>$ at the $p+1^{th}$ generation can be inferred from the wavelet coefficients of the function $fn(a_i) = f(a_i)\cdot n(a_i)$ at generation p.

Figure 8.10 illustrates this result with one example. In this example, the disruption point is given by an uniform probability distribution: (P_0=...=P_{k+1}=1/(k+1)). At generation p, a population with a random distribution of the values $fn(a_i) = f(a_i) \cdot n(a_i)$ is taken. Using this given population, the population at step p+1 was obtained by using the genetic algorithm with a uniform distribution of disruptions points. The distribution of the different schemata was estimated from the average of a large number of simulations. The empirical distribution was subsequently decomposed into its wavelet coefficients and compared to the wavelet coefficients of $fn(a_i) = f(a_i) \cdot n(a_i)$. Both wavelets coefficients are to a factor identical.

Connection between the wavelet-based genetic algorithm and filter theory

We have just shown that the expectation of a schema at the $p+1^{th}$ generation can be inferred from the wavelet coefficients of the function $fn(a_i) = f(a_i) \cdot n(a_i)$ at generation p, with a_i the string coding for the number i. The relation to filter theory is straightforward, as one recalls that the Haar wavelet decomposition can be carried out by filtering the signal. The dyadic wavelet decomposition with Haar wavelets can be represented as a filter tree. Figure 8.11 represents this procedure graphically. The expectation value of a given string at the $p+1^{th}$ generation can be computed by first decomposing the function $fn(a_i) = f(a_i) \cdot n(a_i)$ with Haar wavelets. In a second step, the wavelet coefficients are multiplied by a level-dependant factor W_k. This factor equals $W_k = \sum_{m=k}^{L} P_m$. In a last step, the signal is reconstructed using the wavelet reconstruction algorithm. As the weighting factor increases at lower resolution, $1 = W_0 \geq W_1 \geq ... \geq W_L$, the resulting effect corresponds to filtering the fitness-weighted distribution fn with a low-pass filter. This leads us to the second main result:

Low order schemata with high fitness are on average privileged by wavelet-based genetic algorithm. The low-frequency part of the population distribution fn is weighted more than the high-frequency part in the population of a new generation. More precisely, the weighting factor W_i increases at lower resolution: $1 = W_0 \geq W_1 \geq ... \geq W_L$.

The building block hypothesis can now be formulated for this particular genetic algorithm model. The search is guided by low order schemata of high average fitness. The wavelet-based genetic algorithm privileges regions with high average fitness values. It is therefore expected that the algorithm is a reasonable method for functions with maximal fitness values corresponding to regions with a high average fitness. The search algorithm may be deceived by functions for which an optimum belongs to a region with a low average fitness at low resolution (This topic is developed further below).

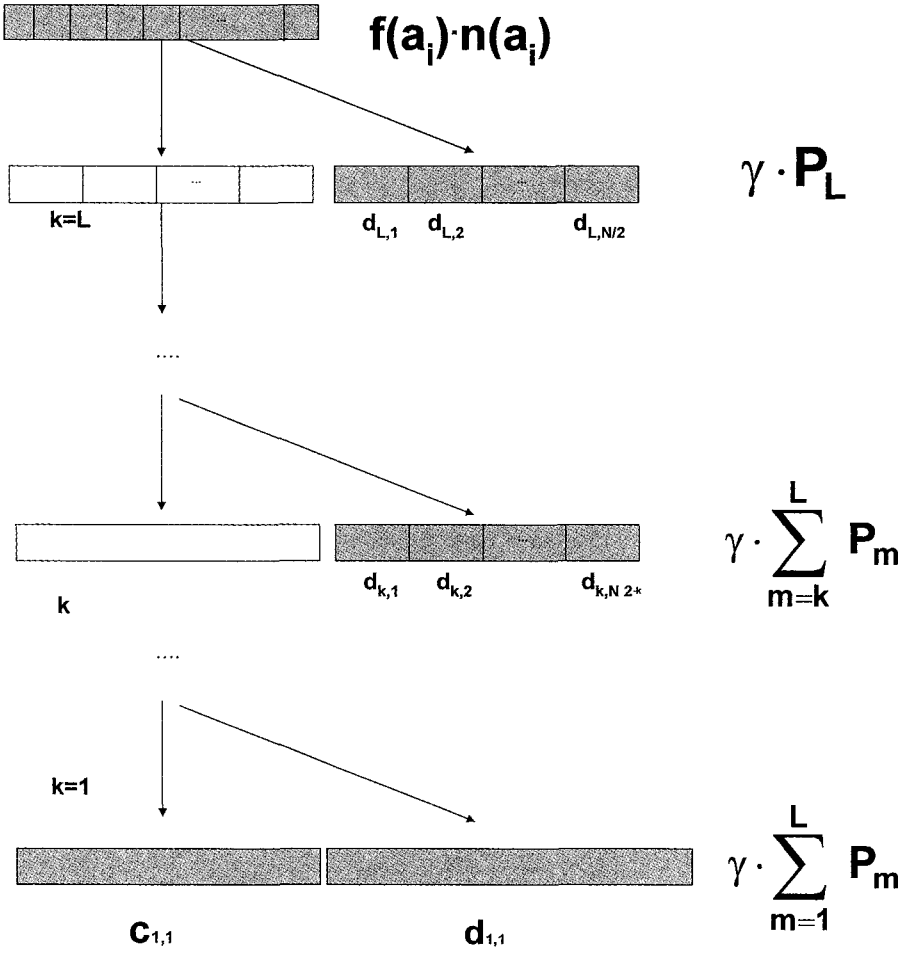


Figure 8.11: The expectation value of a given string at the p+1th generation can be computed by first decomposing the function $fn(a_i) = f(a_i) \cdot n(a_i)$ with Haar wavelets, then multiplying the wavelet coefficients by a level dependent factor (right) to reconstruct finally the signal from the weighted wavelet coefficients. This whole process corresponds to a low-pass filtering of the original signal.

Let us illustrate eq. (8.11) with a second example showing the central influence of the disruption probabilities P_m on the search. The following disruption probabilities P_m were chosen: $P_0=P_1=P_3=P_4=0$; $P_2=1$. At generation p, a population with a random distribution of the values $fn(a_i) = f(a_i) \cdot n(a_i)$ was taken. Using this given population, the population at step p+1 was obtained by using the genetic algorithm with a fixed disruption point after the second bit. The expectation of the different schemata was estimated from the average of a large

number of simulations (fig. 8.12). One observes as expected that on average the sampling rate is almost constant over each of the four domains.

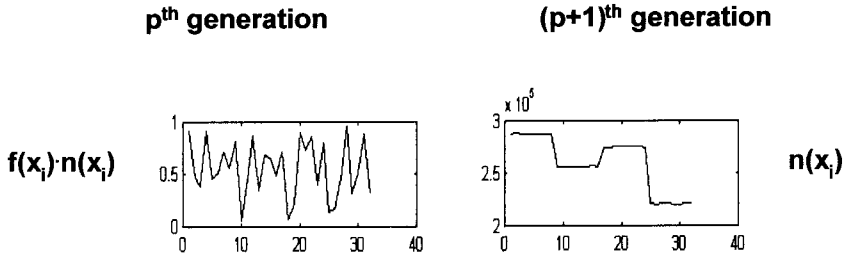


Figure 8.12: Example showing the distribution of the schemata at the $(p+1)^{th}$ generation in function of the fitness-weighted number of strings $f(x_i) \cdot n(x_i)$ at the p^{th} generation. The operator replaces at each generation the last 3 bits randomly (P_2=1).

The disruption probabilities determine very centrally the order of the schemata *guiding* the search. Low-resolution sampling generally privileges exploration over exploitation. The probability P(not a) of a string *a* not being drawn after S samples in a search space of dimension 2^N has a low bound given by

$$P(\text{not } a) \leq S \cdot P_0 / 2^N \qquad \textbf{(8.12)}$$

The probability P_0 sets therefore a lower limit to exploration. Large values of the disruption probability at low resolution tend to flatten the expected distribution curve. Figure 8.12 shows this very clearly. The difference in sampling rates between the different regions is indeed quite small in spite of the large differences in the fitness of individual strings.

On the opposite, if very low values of the disruption probabilities P_m for large m are chosen, then high fitness strings with a small neighborhood of high average fitness are sampled predominantly. In that case, exploitation is generally privileged over exploration, as the algorithm explores mostly solutions in a small neighborhood.

The efficiency of a wavelet-based genetic algorithm can be discussed starting from the evolution equation (8.11). Eq. (8.11) can be put under the following form (In order to simplify the notations, we will give here the equation for the first string only):

$$n_1(k+1, f_1) =$$

$$\gamma \cdot (P_L \cdot n_1(k) \cdot f_1 + P_{L-1} \cdot 1/2 \cdot \sum_{i=1}^{2} n_j(k) \cdot f_j + \ldots + P_0 \cdot 1/2^N \cdot \sum_{i=1}^{2^N} n_i(k) \cdot f_i)$$

(8.13)

As the trajectory converges for large k, (8.13) can be approximated for large k by

$$0 \approx \alpha \cdot n_1(k+1, f_1)/\bar{n}(k+1) - \alpha \cdot n_1(k, f_1)/\bar{n}(k) =$$
$$\gamma \cdot ((P_L \cdot f_1 - \alpha/\gamma) \cdot n_1(k, f_1)/\bar{n}(k) + \qquad \textbf{(8.14a)}$$
$$1/\bar{n}(k) \cdot (P_{L-1} \cdot 1/2 \cdot \sum_{i=1}^{2} n_j(k) \cdot f_j + \ldots + P_0 \cdot 1/2^N \cdot \sum_{i=1}^{2^N} n_i(k) \cdot f_i))$$

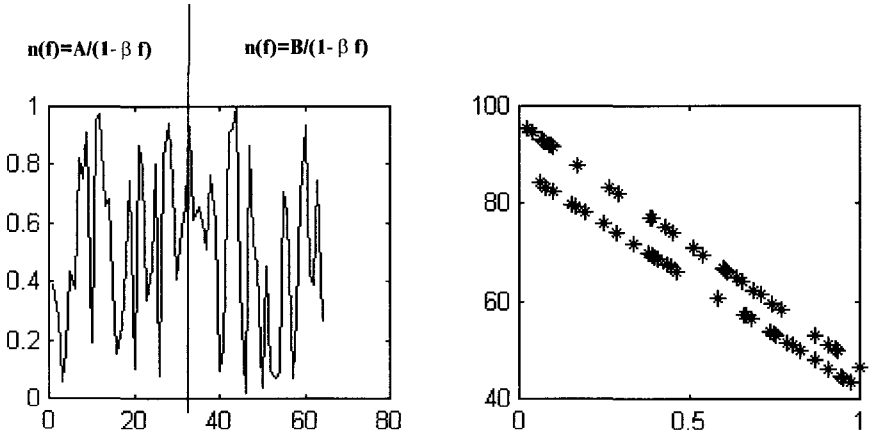
with $\alpha = \lim_{k \to \infty} \bar{n}(k+1)/\bar{n}(k)$ and subsequently

$$n_1(k, f_1)/\bar{n}(k) \cdot \approx 1/(\beta - P_L \cdot f) \cdot$$
$$(P_{L-1} \cdot 1/2 \cdot \sum_{i=1}^{2} n_j(k)/\bar{n}(k) \cdot f_j + \ldots + P_0 \cdot 1/2^N \cdot \sum_{i=1}^{2^N} n_i(k)/\bar{n}(k) \cdot f_i)$$

(8.14b)

with $\beta = \alpha/\gamma$.

Eq.(8.14) shows that at equilibrium, the distribution of the population follows locally a $\Delta_i/(\beta - P_L \cdot f)$ law, with Δ_i a constant depending on the disruption probabilities. The efficiency of the search is determined by the form of the function Δ_i. A constant value of Δ_i over the whole search space makes the distribution of a string independent of its neighbors. In that case, it is not possible to extract information from past samples and the wavelet-based genetic algorithm is worst than a random search. Generally speaking the wavelet-based genetic algorithm is only efficient, if the objective string(s) is within a region of high average fitness. In that case, meaningful information can be extracted from past samples to direct the search to those areas. In other words, for the search to be efficient, the objective string(s) must coincide with regions having a value of Δ_i well above average. A second condition for an efficient search of the objective string is that the values of the denominator $(\beta - P_L \cdot f)$ are small in comparison to the denominator Δ_i. Figure 8.13 illustrates this with a simple example. In that example, sampling is limited to the first and the last two resolution levels. Figure 8.13b represents the inverse of the sampling probability at equilibrium for the fitness distribution in figure 8.13a. For each of the two regions in fig. 8.13a, the $\Delta_i/(\beta - P_L \cdot f)$ relationship predicted by (8.14) is observed. In the example of fig. 8.13, the search will be quite inefficient, whatever the objective is, because Δ_i in both regions are quite similar and therefore the distribution function is essentially given by the denominator.



a) b)

Figure 8.13: The fitness function in a) is sampled with P_0=0.3; P_1=0.4; P_7=0.3; P_2=...=P_6=0. b) The inverse of the distribution function at equilibrium is plotted as a function of the fitness value. The points are aligned on two lines, each line corresponds to one of the two segments in a).

Population evolution and deceptive functions

The concept of deceptivity is central to genetic algorithms. The minimal objective of a genetic algorithm is to perform on average better on a class of problems than a random search. If this minimal condition does not hold, we will say that the problem is deceptive for the considered algorithm. In order to discuss deceptivity, we will first explain how to estimate the expectation of the different strings analytically.We have seen that the expectation $E^{(1)}(a_i)$ of a string a_i after a single generation is of the form

$$E^{(1)}(a_i) = G[f(a_i)] \qquad \textbf{(8.15)}$$

with $G[f(a_i)] = \gamma \cdot (f_0 + \sum_{m,n} d_{m,n} \cdot (\sum_{j=m}^{L} P_j) \cdot \psi_{m,n}(i))$

with $\psi_{m,n}(i)$ the value of the corresponding Haar wavelet at the position of the string a_i. If the initial population is drawn from a uniform population then $d_{m,n}$ corresponds to the wavelet coefficients of the filtered fitness-weighted distribution function $f(a_i) \cdot n(a_i)$. In the limit of a very large population, the outcome of the wavelet-based genetic algorithm can be computed. The expectation at generation p is given recursively by

$$E^{(p)}(a_i) = G[f \cdot E^{(p-1)}(a_i)] \qquad \textbf{(8.16)}$$

We will now show that $E^{(p)}(a_i)$ converges with p towards an equilibrium distribution. The convergence of the iteration process can be discussed within the framework of heuristic random search (Vose, 1999). First one puts (8.16) under a matrix form:

$$z[p+1] = G_M(z[p]) \tag{8.17}$$

with $z[p] = n(x_i)/(\sum_{i=1}^{Pop} n(x_i)/Pop)$, the normalized population distribution expected at generation p.

Stable fixed points z_s of the transform G_M correspond to eigenvectors of G_M:

$$G_M[z_s] = \lambda \cdot z_s \tag{8.18}$$

The matrix G_M can be put under the form:

$$G = M_L \cdot F \text{ with } F = \begin{pmatrix} f_1 & 0 & \dots & & 0 \\ 0 & f_2 & 0 & & \dots \\ \dots & 0 & \dots & 0 & \\ & & 0 & & 0 \\ 0 & \dots & & 0 & f_{2^L} \end{pmatrix} \tag{8.19}$$

The matrix M may be defined recursively. For a vector of length $N=2^L$, M_L is given by the set of equations:

$$M_{N,0} = P_L \tag{8.20a}$$

$$M_{N,2^k} = \begin{pmatrix} M_{N,2^{k-1}} & 0 \\ 0 & M_{N,2^{k-1}} \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \cdot \frac{P_{L-k}}{2^k} \tag{8.20b}$$

$$M_L = M_{N,2^L} \tag{8.20c}$$

(0 and 1 represent 2^{k-1} x 2^{k-1} matrices with only zeros, respectively 1.)

The matrix G_M has only positive coefficients and is irreducible provided $P_0 \neq 0$ and $f_i > 0, \forall i$. It follows that the conditions for the Frobenius-Perron theorem are fulfilled (Gantmacher, 1977). The Frobenius-Perron theorem states that an irreducible matrix with only positive coefficients has a positive eigenvalue that corresponds to an eigenvector with only positive values. A corollary of the Frobenius-Perron theorem is that there cannot be more than two linearly independent eigenvectors with only positive values (For a demonstration, see for instance Gantmacher, 1977).

As there is a unique eigenvector with only positive values, the iterative application of the transform G_M converges towards a single stable point z_s. Figure 8.14 provides an example, showing a fitness function, the normalized eigenvector

of the transform G_M and the distribution obtained from the iterative application of (8.17). For comparison, figure 8.14b shows the equilibrium stable fixed distribution function obtained by solving the eigenvector problem (8.18). Both populations are identical.

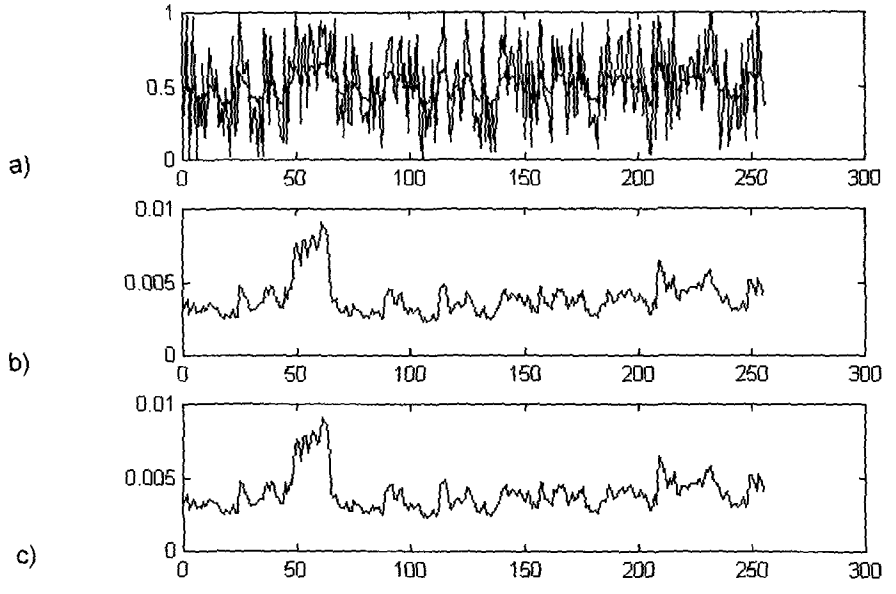


Figure 8.14: Example showing the stable distribution of the population ($P_0 = ... = P_8 = 1/9$): a) Fitness of the different strings, the second line shows the expected distribution after one generation; b) Population after 100 iterations using (8.17); c) Stable population computed with (8.18).

Let us introduce here a working definition of deceptiveness. A genetic algorithm is deceptive for a given objective if the search of strings, fulfilling the objective, is on average less efficient than a random search. Different objectives may be thought of. The objective may be to find a string or a number of strings with a fitness above a given threshold. A reasonable objective may also be to discover regions of high average fitness. This latter objective is relevant to many control problems in which stability of the solution is an issue. In summary, deceptivity can only be defined in relation to an objective. It is generally easier to show that for some objective, the wavelet-based genetic algorithm is deceptive than the inverse. If, in the infinite-population limit, the algorithm is less efficient than a random search at each generation then simulations show that the finite-population case is also deceptive if a reasonable number of samples are drawn

(Proving or refuting this conjecture may be interesting, though we will not embark on this!). In other words, we postulate that the condition below is a very strong indication of deceptiveness, also in the finite-population case.

$$E^{(p)}(a_i)/(\sum_i E^{(p)}(a_i)) < 2^{-N} \qquad \forall p \tag{8.21}$$

with 2^{-N} the size of the search space.

Giving criteria for the non-deceptiveness is even more difficult. The reason is that having a large expectation for the objective string is not sufficient to guarantee that the search is more efficient than a random search. If the exploration rate of new solutions is too small, then the probability of discovering better solutions is also small. Multi-sampling of a string reduces the efficiency of the algorithm. Multi-sampling is not so much a limitation in low-resolution sampling (i.e. $P_m>0$ only for very small m and a large search space). In that special case, the algorithm is generally quite efficient if the condition

$$E^{(p)}(a_i)/(\sum_i E^{(p)}(a_i)) > 2^{-N} \tag{8.22}$$

holds at each generation.

The deceptiveness of a fitness function depends also on the chosen free parameters. The disruption probabilities $P_0,...,P_L$ are the main parameters relevant to the wavelet-based genetic algorithms. Their choice determines the performance of the algorithm. A fitness function may be deceptive for a given objective at a certain resolution and non-deceptive at another resolution. Figure 8.15 illustrates this. It shows the equilibrium distribution for 4 sets of disruptions probabilities using in each case the same fitness function to start with. Low-resolution sampling leads to the second largest peak being mostly sampled. In that case the highest peak is *washed out* through filtering. Sampling of the highest peak is only significant for the disruption probabilities corresponding to the two smallest low-resolution disruption probabilities. In that particular case, the fourth resolution is the best to find the maximum of the fitness function. From this example, one understands that changing the resolution during learning may have in some instances a positive effect on the efficiency of the search. In particular, changing the disruption probabilities during the search may be recommended in multi-modal fitness landscapes for which high-fitness regions correspond to large basins. A low-resolution search with the wavelet-based algorithm permits to localize, with a limited number of samples, regions of high average fitness values. Sampling at higher resolution permits to focus the search on these high-fitness regions. Needless to say, that an optimization of the disruption parameters P_m necessitates some preliminary knowledge on the shape of the fitness landscape.

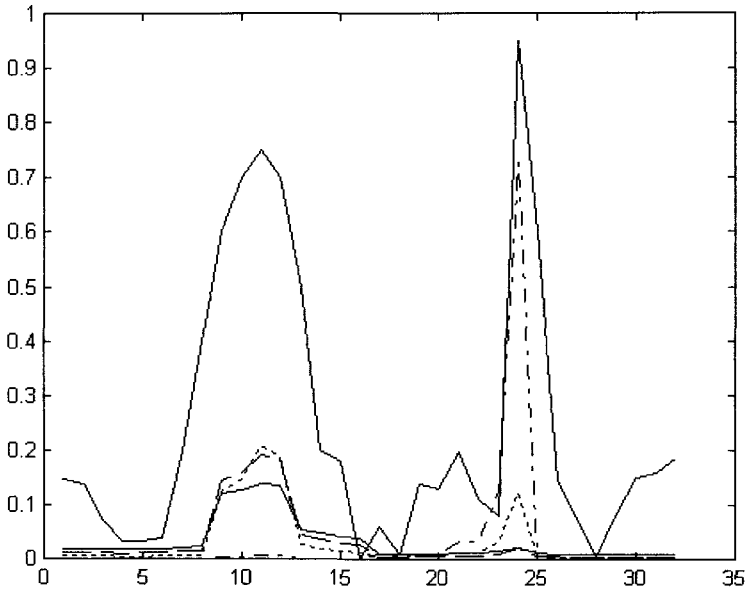


Figure 8.15: Equilibrium distribution function of the distribution of strings defined by fitness function given by the upper solid line. The disruption probabilities are chosen proportional to $P_m \propto 1/(2+(L-m))^{\alpha}$: a) $\alpha = 0$ (solid line); b) $\alpha = 1$ (long dash); c) $\alpha = 2$ (short dash); $\alpha = 3$ (dash-dot). The high-frequencies are less filtered as one goes from a) to d).

Before addressing the question of the relevance of the infinite-population description to the finite-population case, let us first summarize the above discussion.

-We have presented a wavelet-based genetic algorithm that makes explicit some of the connections between Haar wavelets and genetic algorithms. The algorithm uses a single operator that tries to catch some of the main features of the crossover and mutation operators. The simplicity of the model allows the derivation of analytical results, a somewhat rare case in genetic algorithms. In particular, the expected population can be computed in terms of the wavelet coefficients of the fitness function. As wavelet theory has an equivalent formulation within filter theory (subband coding), the results can be expressed also in terms of filtering.

-The disruption probabilities determine to a very large extent the respective importance of exploitation and exploration. Large probabilities for low resolution sampling results into exploration being privileged over exploitation, while the reverse holds if sampling is performed in the neighborhood of the fittest strings. The efficiency of the algorithm depends centrally on the existence of a good correlation between the objective strings and the regions of high average fitness. Roughly speaking, the algorithm is efficient if the search is guided towards the

objective string through the sampling of regions having high average fitness. The performance of the search depends critically on the disruption probabilities. We have presented an example (fig. 8.15) in which a fitness function is deceiving for a particular setting of the disruption probabilities and a given objective string. In that particular case, a slight modification of the disruption probabilities leads to a very efficient algorithm.

Let us discuss now the finite population case. In the infinite-population model, the distribution converges towards a stable equilibrium distribution. The dynamics of the wavelet-based genetic algorithm is therefore much simpler than the one of the simple genetic algorithm by Vose. The dynamics of the simple genetic algorithm can be indeed very rich and often the relevance of the infinite-population distribution to the standard genetic algorithm in *real-world applications* is not clear. The existence of an equilibrium distribution in the wavelet-based genetic algorithm represents a basis for discussing the finite population model. The equilibrium distribution in the infinite-population model can be used as a useful guidance for understanding the algorithm in the only practically relevant case of a finite population. A central but very difficult question is to know to which extent the equilibrium distribution characterizes the population well enough. In the finite, but large, population case, the distribution moves in many generations from a random distribution given by the initial population to an average distribution described approximately by the equilibrium distribution. The larger the population, the better is the average distribution approximated by the equilibrium distribution. If the equilibrium distribution is qualitatively very different from the transient infinite-population distributions, any discussion of the algorithm' performance in the finite population case is extremely speculative. The closer is the first generation to the infinite population distribution the more representative is the infinite population behavior to the small population case. So a legitimate question is to determine to which extent the first generation distribution is correlated to the equilibrium distribution. Simulations show that the correlation is not perfect, but that a significant trend does exist. Let us show this with an example. Let us consider fitness functions with values drawn from a uniform distribution. After generally a single generation, the maximum value $\max_i E^{(p)}(a_i)$ locks on a given position i (fig 8.16a).

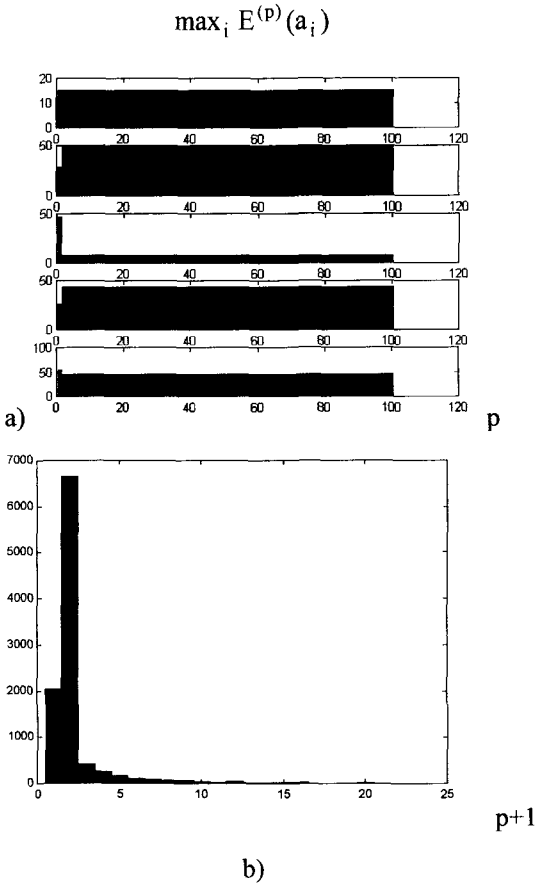


Figure 8.16: Examples of numerical simulations using eq.(8.16). a) The maximum value of $E^{(p)}(a_i)$ is reported as a function of the generation p . b) Distribution of the generation at which the string with the largest probability at equilibrium corresponds to the maximum of $E^{(p)}(a_i)$. The statistics was made with 10000 fitness functions with values chosen randomly in a uniform distribution between zero and one (P_0=...=P_7=1/7).

In above 85% of the cases, the genetic algorithm does promote the string with the largest value of the fitness after filtering, that is the string given by (8.15), and not the maximum fitness value. Only 20% of the fittest strings at equilibrium coincide with the string of maximum fitness. The above results were found to be quite representative. It implies that the expected distributions, after one generation and at equilibrium, are often well correlated and are generally both relevant to a qualitative discussion of the algorithm performance.

Multiresolution search

The wavelet-based genetic algorithm, presented in the last sections, captures much of the essence of the standard genetic algorithm, while being analytically tractable. Besides being simple, the wavelet-based algorithm is also interesting as it can be regarded as a prototype algorithm of a larger class of search algorithms, that are presented below. This last section is conceptually important as it extends wavelet-based genetic algorithms to a broader class of search algorithms based on the application of multiresolution techniques. We propose to describe these techniques by the generic term of multiresolution search. We will give below two working definitions of *multiresolution search.* The first one is quite general and the second one is a restriction of the definition of multiresolution search to wavelet-based multiresolution search.

Multiresolution search algorithms allow searches in both the continuous and the discrete domain. Let us start by explaining how to extend the simple wavelet-based genetic algorithm presented in the last sections to a continuous search space parameter. This example will permit to grasp already some of the main ideas behind the concept of multiresolution search.

The wavelet-based genetic algorithm can be expressed under a more general form: Suppose that at the p^{th} generation, a finite subset of the search space $\{x_1,...,x_n\}$ has been tested. Each element in the subset has a fitness $f(x_i)$. The finite subset of candidate solutions to the search problem at the next generation is determined according to the following procedure.

1) A solution of fitness f_i is replicated on average $\gamma \cdot f_i$ times.

2) Each element of the subset created in (1) is modified by an operator Θ_m (m=0,...,L). The probability of using the m^{th} operator m is P_m ($\sum_m P_m = 1$).

The operator Θ_m transforms an element x_i into x with a probability density function $\theta_m(x_i \to x)$.

$$\theta_m(x_i \to x) \propto 1/2^{L-m} \cdot \sum_n H_{m,n}(x_i) \cdot f(x_i) \cdot H_{m,n}(x) \qquad \textbf{(8.23)}$$

with $H_{m,n}$ the scaling function associated to the Haar wavelet (The normalization of $H_{m,n}$ is chosen here such that $H_{m,n}$ takes values either zero or one). It is not difficult to show that, for binary coding, the operator Θ_m is equivalent to the operator O_m , the operator that replaces the last m bits in the strings. The cumulative effect of the different operators Θ_m can be described by the operator Θ that transforms an element x_i into x with a probability density function $\theta(x_i \to x)$ given by:

$$\theta(x_i \to x) \propto \sum_m P_m \cdot 1/2^{L-m} \cdot \sum_n H_{m,n}(x_i) \cdot f(x_i) \cdot H_{m,n}(x) \qquad \textbf{(8.24)}$$

Based on the above example, we propose a general definition of what is meant with *multiresolution search.*

Definition:
A multiresolution search is defined on a search space (continuous or discrete). At the p^{th} generation, a finite subset of the search space $\{x_1,...,x_n\}$ is tested for fitness. A new subset of elements is created by replicating the elements x_i on average $\gamma \cdot f_i$ times, with f_i the fitness of the element x_i The value γ may be a constant or modified from generation to generation. Each element of the new subset is subsequently transformed by an operator Θ_m (m=0,...,L). The probability of using the m^{th} operator m is P_m ($\sum_m P_m = 1$). The operator Θ_m transforms an element x_i into x with a probability density function of the form $\theta_m(x_i \to x) \propto \sum_n F_{n,m}(x_i) \cdot f(x_i) \cdot G_{n,m}(x)$ with

$F_{n,m}(x) = F(2^m \cdot x - n); G_{n,m}(x) = G(2^m \cdot x - n)$.

The resulting subset defines the search subspace at the next generation.

Figure 8.17 summarizes the algorithm.

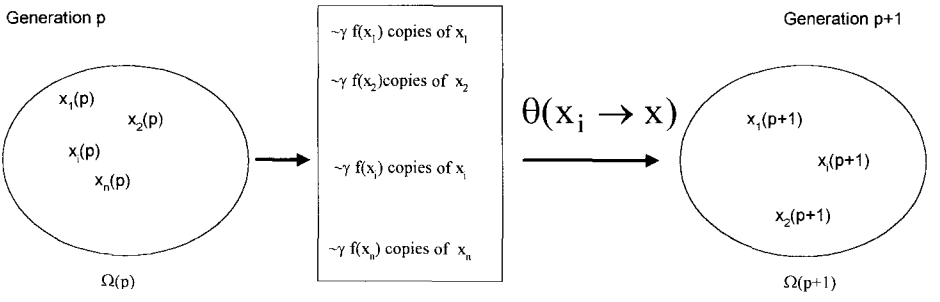


Figure 8.17: General description of a multiresolution search.

The above formulation of the wavelet-based genetic algorithm in terms of a multiresolution search does not only extend the algorithm to a continuous search space but permits also to include multiresolution searches based on different wavelet constructions than the Haar wavelets.

Definition:
A wavelet-based multiresolution search is a multiresolution search using an operator Θ that transforms an element x_i into x with a probability density function of the form:

$$\theta\ (x_i \to x) \propto \sum_m P_m \cdot 1/2^{L-m} \cdot \sum_n \tilde{\phi}_{m,n}(x_i) \cdot f(x_i) \cdot \phi_{m,n}(x) \qquad \textbf{(8.25)}$$

The functions $\tilde{\phi}$ and ϕ are respectively the dual scaling functions and the dual scaling functions corresponding to the biorthogonal wavelets $\tilde{\psi}$ and ψ. The probabilities P_m must be appropriately chosen such as to ensure non-negative values of θ. The reason is that either $\tilde{\phi}$ or ϕ have negative values for any scaling function except the Haar scaling function. In order to obtain non-negative density functions, a modified probability density function must be used, assuming for simplicity that $\min(\phi_{m,n}(x)) \geq 0$ one may write (8.25) under the form:

$$\theta_m(x_i \to x) \propto 1/2^{L-m} \cdot \sum_n (\tilde{\phi}_{m,n}(x_i) + \delta) \cdot f(x_i) \cdot \phi_{m,n}(x), m > 0 \qquad \textbf{(8.26a)}$$

$$\theta_0(x_i \to x) = 0 \qquad \textbf{(8.26b)}$$

If $\delta \geq -\min(\tilde{\phi}_{m,n}(x))$ then the values of the density functions are always non-negative. In that formulation, an element x_i is transformed into x with a probability density function $\theta(x_i \to x)$ proportional to:

$$\theta(x_i \to x) \propto \sum_{m \neq 0} P'_m \cdot 1/2^{L-m} \cdot \sum_n (\tilde{\phi}_{m,n}(x_i) + \delta) \cdot f(x_i) \cdot \phi_{m,n}(x) \qquad \textbf{(8.27)}$$

Eq. (8.27) is almost equivalent to (8.25). The difference is that the lowest resolution sampling is directly included into the higher resolution sampling by adding the constant δ (fig. 8.18). By equating (8.25) to (8.27), the probabilities P_m can be computed from P'_m and δ:

$$P_0 = \delta/(1+\delta); \quad P'_m = P_m / (\sum_{m \neq 0} P_m \cdot (1+\delta)) \qquad \textbf{(8.28)}$$

The value of δ determines the amount of random sampling. A value of $\delta = 0.2$ means, for instance, that random sampling is chosen in about 16 % of the case. For some orthogonal wavelets, the value of δ is quite small.

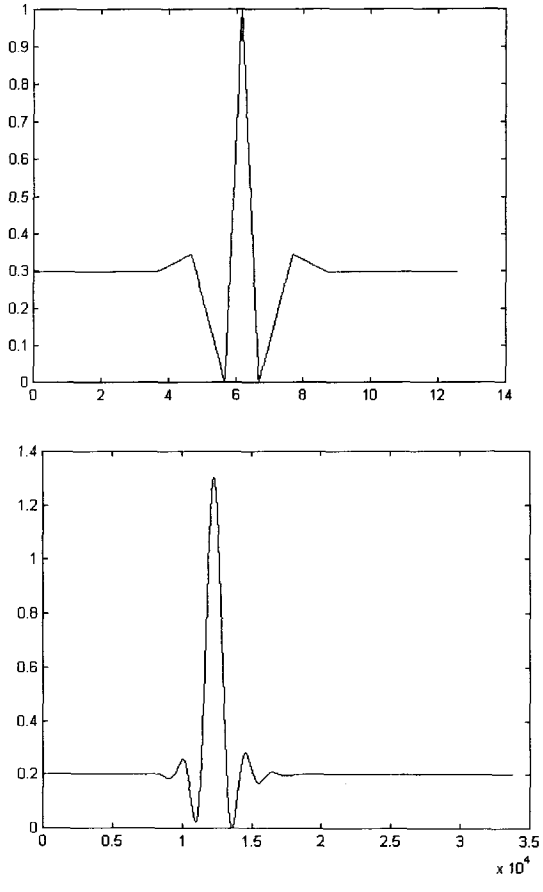


Figure 8.18: In order to avoid negative probabilities, an offset is added to the scaling functions, that are after normalization interpreted as a probability density. The offset determines the proportion of randomly chosen elements at each generation. Left: biorthogonal 4.2 spline (Its main advantage: boundary scaling functions can be built using lifting), Right: Coifman 8 (the offset corresponds to a small amount of random search of the order of 16%).

The main results of the previous sections can be translated to the continuous case (in the limit of an infinitely small quantization step!) and we will limit the discussion to stating two main results in the infinite population limit:

- The probability density function $N(x,p)$ of a candidate solution x at generation p can be estimated recursively. The probability density function converges with p towards an equilibrium distribution.
- The probabilities P_m determine to a very large extent the respective importance of exploitation and exploration. Large probabilities for low resolution sampling results into exploration being privileged over exploitation, while the reverse holds if sampling is performed in the neighborhood of the fittest strings. The efficiency of the algorithm

depends centrally on the existence of a good correlation between acceptable solutions and the regions of high average fitness. Roughly speaking, the algorithm is efficient if the search is guided towards good solutions through the sampling of regions having high average fitness.

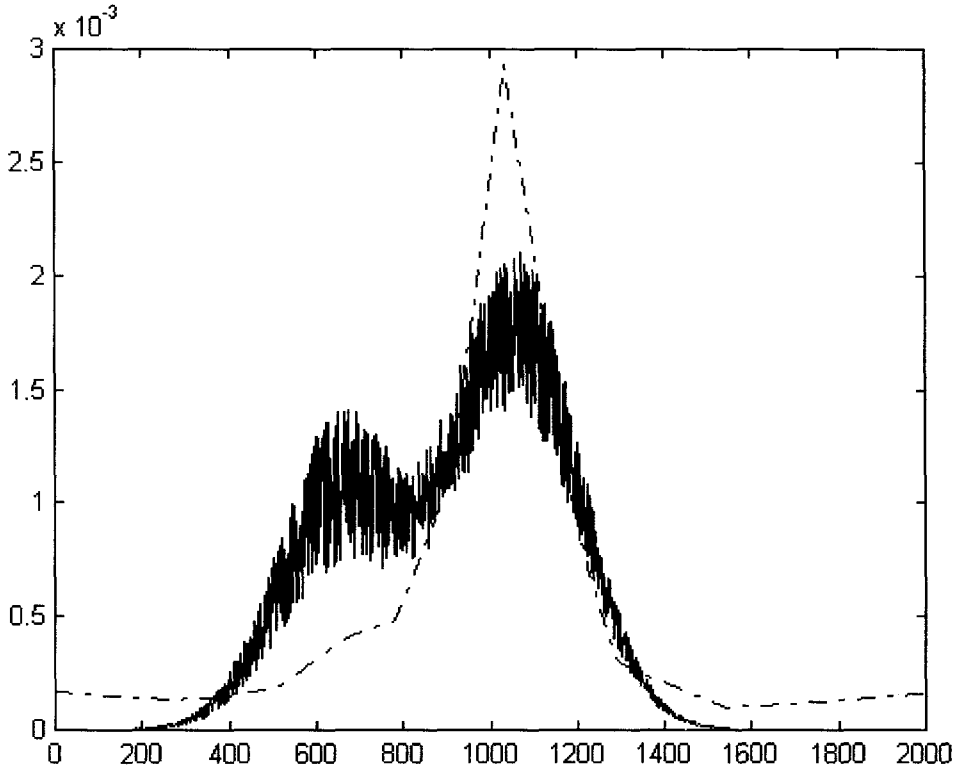


Figure 8.19: Example showing a normalized fitness function (solid line) and the sampling probability distribution on the search space at equilibrium (dash) for P_0=0.3; P_6=7/40; P_5=7/40; P_4=7/40; P_3=7/40 (P_k corresponds to the projection on the 2^{L-k} dilated scaling function; P_0 to the contribution of random sampling (Search space[0,2^L=2048]).

ANNEXES LIFTING SCHEME, NONLINEAR WAVELETS

This page is intentionally left blank

Annexes

Lifting Scheme

The Fourier approach has been for quite some time the main method to construct wavelets. The situation has changed with the discovery of the lifting scheme (Sweldens, 1995). In the lifting scheme, wavelets are derived in the spatial space. The lifting scheme has lead to the development of wavelets defined on a sphere, wavelet constructions to process boundaries and to multiresolution schemes on irregular intervals. All these new constructions are regrouped under the concept of second generation wavelets. All wavelets constructions obtained in the Fourier domain can be derived in the spatial domain by using the lifting scheme (Daubechies, 1998). For that reason, the lifting scheme is often considered as a generalization of wavelet theory, and therefore the denomination of *second generation wavelet* was introduced.

In this introduction on second generation wavelets, the goal is to give the flavor of the method and to present constructions that have been used in this book. The lifting scheme is quite intuitive. Consider a function $y_n = f(x_n)$ with 2^n samples. The purpose of the lifting scheme is to decompose this function into the sum of a coarse approximation together with a correction to the coarse approximation. Up to this point, the lifting scheme is in essence similar to the fast wavelet decomposition algorithm. The particularity of the lifting scheme is that the decomposition is carried out by filtering alternatively the function at odd and even locations. In its simplest version, a decomposition with the lifting scheme is carried out by cascading a prediction and an update stage. The prediction stage estimates the value of $f(x_n)$ at odd locations (n=2k+1 with k an integer) from the points at even locations (n= 2k with k an integer). The correction to the predicted values furnishes the output of the prediction stage. The update stage modifies the values at even location to preserve average.

Let us take the example of the Haar wavelet decomposition. A function f(x) can be estimated at an odd location from its value at the previous even location:

$$\hat{f}(x_{2k+1}) = f(x_{2k}) \qquad \textbf{(A1)}$$

The correction to this prediction is given by

$$\Delta f(x_{2k+1}) = (f(x_{2k+1}) - \hat{f}(x_{2k+1})) = f(x_{2k+1}) - f(x_{2k})) \qquad \textbf{(A2)}$$

Introducing the notation $Odd_k = f(x_{2k+1})$ and $Even_k = f(x_{2k})$, (A1-A2) can be put under the form

$$\hat{f}(x_{2k+1}) = Even_k \quad \textbf{(A3)}$$

$$\Delta f(x_{2k+1}) = Odd_k - P(Even_k) \quad \textbf{(A4)}$$

with $P = f(x_{2k})$ a lazy function of the points at even location.
Eq.(A3-A4) can be summarized by the wiring diagram in fig. A1.

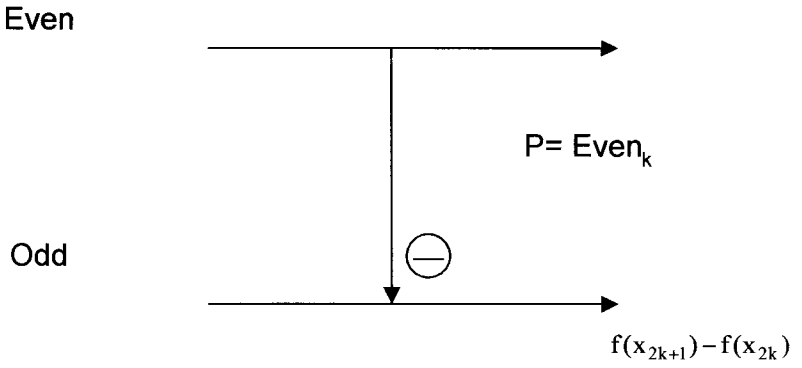


Figure A1: The wiring diagram represents schematically the transform given by equations(A3-A4).

The second part of the algorithm consists of updating the points at even locations, so that the average value on the points at even locations equals the average value of the function f(x):

$$\sum_{k=0,1...n} f(x_{2k}) + U(f(x_{2k+1}) - f(x_{2k})) = 1/2 \sum_{k=0,1...n} f(x_{2k+1}) + f(x_{2k}) \quad \textbf{(A5)}$$

This condition is necessary as a wavelet decomposition must preserve the average of a function. The average is preserved with the following update function U:

$$U = 1/2(f(x_{2k+1}) - f(x_{2k})) \quad \textbf{(A6)}$$

Setting $Even_k = f(x_{2k})$, $Odd_k = f(x_{2k+1})$ and $Oddp_k = Odd_k + P(Even_k)$ the wavelet decomposition with the Haar wavelet is given in the lifting scheme framework by the wiring diagram in fig. A2. The output points at odd locations correspond to the detail coefficients, while the output points at even locations are the approximation coefficient. A wavelet decomposition can be carried out by

cascading several diagrams, using at each level the output even datapoints as input to the next decomposition level.

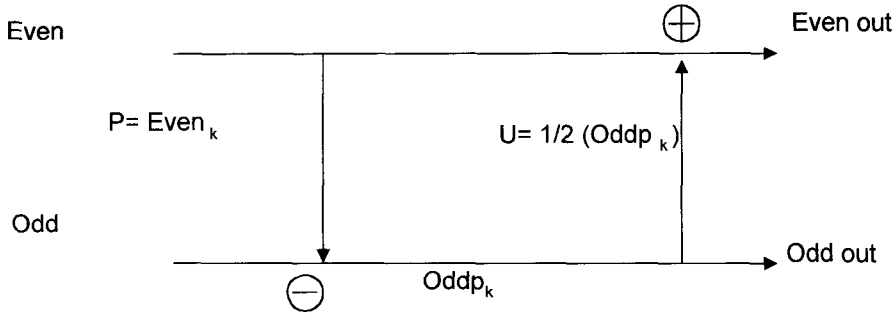


Figure A2: Wiring diagram in the lifting scheme corresponding to a one level Haar decomposition. The output points at odd locations correspond (to a factor) to the detail coefficients, while the output points at even locations are the approximation coefficients.

In the next example, we will show how to construct biorthogonal spline wavelets with the lifting scheme. The wavelet obtained with this construction corresponds to a Cohen-Feauveau-Daubechies biorthogonal wavelet (Cohen, 1992). This example will show the usefulness of the lifting scheme to construct wavelets. It will also make clear that the lifting scheme is also an efficient algorithm for a multiresolution analysis. The lifting scheme is in many cases even more efficient than the fast wavelet algorithm.

Biorthogonal spline-wavelets constructions with the lifting scheme

The lifting scheme associated to biorthogonal spline wavelets bears many similarities to the lifting scheme for Haar wavelets presented in the introduction. In the first stage, the values of f(x) are estimated at odd locations from the points at even locations and the difference between the true value and the prediction corresponds to the detail coefficient. The value at an odd location is predicted by taking the average of the two values of f(x) at the two neighboring location points.

The corresponding wiring diagram is given by

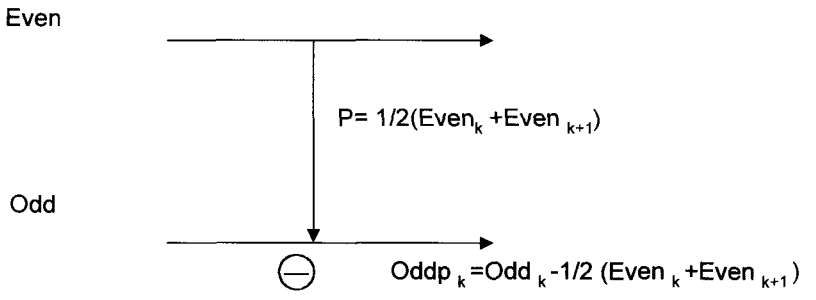


Figure A3: Prediction stage for (2.2) biorthogonal spline wavelets.

The update stage is designed such as to preserve the average value of the function after the decomposition stage. An update operator that fulfills this condition is

$$U = 1/4 \cdot (\mathrm{Oddp}_k + \mathrm{Oddp}_{k+1}) \qquad \textbf{(A7)}$$

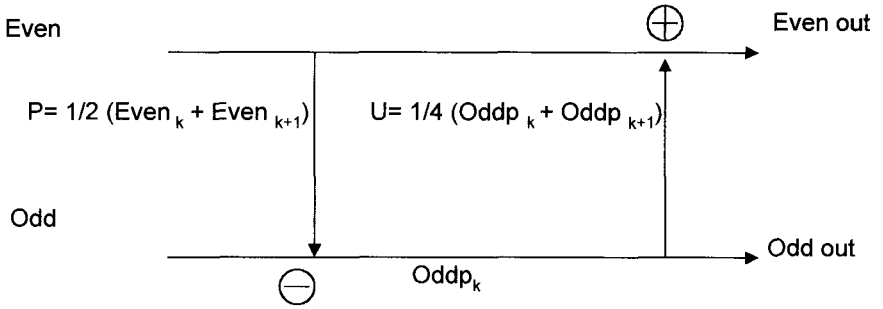


Figure A4: Wiring diagram for (2.2) biorthogonal spline wavelets.

By substituting the operator P in the above expression, one obtains the approximation coefficients in terms of the input signal:

$$\mathrm{Evenp}_k = -1/8 \cdot (\mathrm{Even}_{k-1}) + 1/4 \cdot \mathrm{Odd}_{k-1} + 3/4 \cdot \mathrm{Even}_k + 1/4 \cdot \mathrm{Odd}_k - 1/8 \cdot \mathrm{Even}_{k+1}. \qquad \textbf{(A8)}$$

Similarly the detail coefficients can be written

$$\mathrm{Oddp}_k = -1/2 \cdot \mathrm{Even}_k + \mathrm{Odd}_k - 1/2 \cdot \mathrm{Even}_{k+1} \qquad \textbf{(A9)}$$

Eq.(A8-A9) correspond to the filter coefficients of the biorthogonal (2,2) spline-wavelets. The reconstruction algorithm is obtained by inverting the wiring diagram as shown in fig. A5.

The ease with which one can invert the wiring diagram from the reconstruction to the decomposition algorithm is certainly one of the strong points of the lifting scheme. This property is intrinsic to the lifting scheme, as at each step, one half of the coefficients are recalculated from the other half of the coefficients. This very construction makes each stage of the construction invertible. Let us take the example of the prediction stage. The prediction stage corresponds to the operation:

$$Oddp_k = Odd_k + P(Even_k) \qquad \textbf{(A10)}$$

This stage is inverted quite simply :

$$Odd_k = Oddp_k - P(Even_k) \qquad \textbf{(A11)}$$

As the even coefficients are not changed during the operation, the expression P(Even) can be computed and the operator inverted. Similarly all other stages in the wiring diagram can be inverted, making the wiring diagram perfectly invertible.

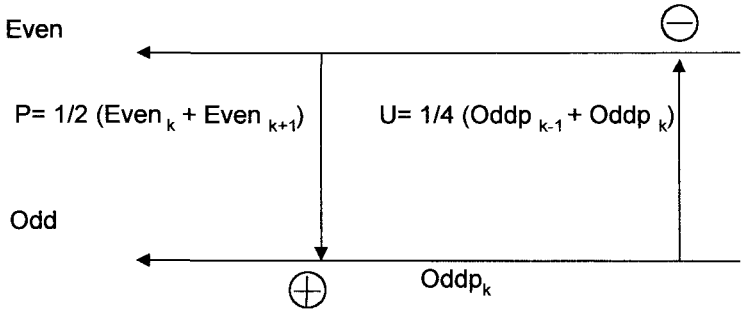


Figure A5: The wiring diagram in fig. A4 can be inverted for reconstructing the original signal losslessly from the transformed data points.

The wavelet and scaling functions can be constructed quite easily. The wavelet function is constructed by putting all zeros but a one in the reconstruction diagram at the odd wire. By cascading several reconstruction diagrams, a good approximation of the wavelet function is obtained.

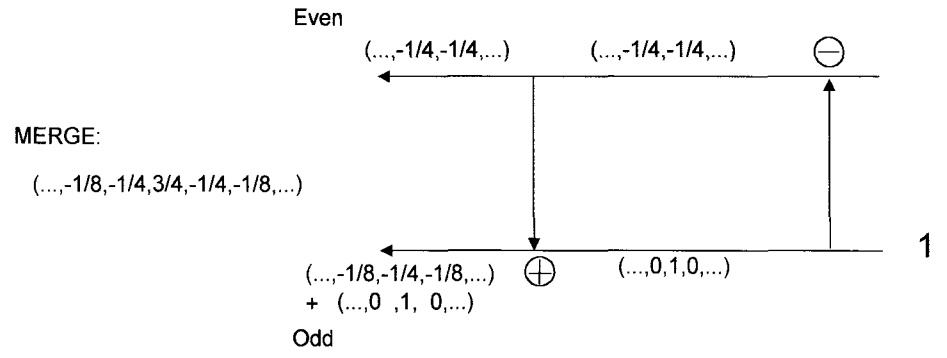


Figure A6: The (2.2) biorthogonal wavelet can be obtained by cascading the above wiring scheme corresponding to the reconstruction diagram.

After merging, one obtains the coefficients 1/8(-1,-2,6,-2,-1) which corresponds to the filter coefficients in the reconstruction algorithm with the fast wavelet reconstruction algorithm for the biorthogonal (2,2) construction. This gives an example of the equivalency between the fast wavelet decomposition and the lifting scheme.

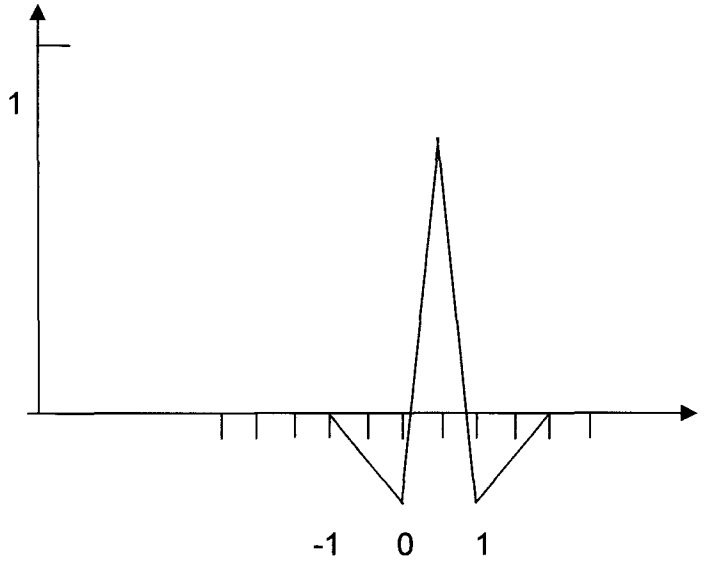


Figure A7: The (2.2) biorthogonal wavelet was obtained by cascading the wiring diagram in fig. A6.

The choice of the scaling function does not determine univocally the wavelet function. Different wavelets can be obtained by changing the update in the wiring diagram. The scaling function can be obtained with a similar procedure, putting a

one in the *Even* wire as shown below. The reconstruction coefficients for the scaling functions are obtained: (0.5, 1, 0.5).

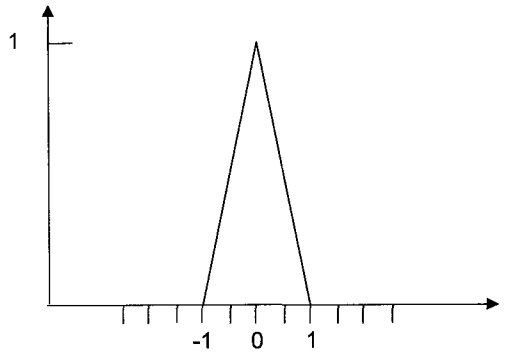


Figure A8: Scaling function associated to the (2.2) biorthogonal spline construction. The scaling function is obtained by putting all zeros but one *1* in the even wire of fig.A6.

Nonlinear wavelets

The decomposition of a function in a sum of wavelets is by nature a linear method, as the wavelet coefficients are obtained by using a cascade of linear filters. Wavelet theory is by essence a linear method, even if some wavelet-based methods, for instance denoising, may use nonlinear aspects. In the linear case, the above reconstruction stage can be put under the form:

$$x[n] = H(\{d_{m,n'}\}) + G(\{c_{M,n'}\}) \qquad \textbf{(B1)}$$

Nonlinear wavelet decompositions, or critically decimated nonlinear filter banks, as there are sometimes called in the filter literature, have been proposed by many authors (Egger, 1995; Claypole, 1997; Queiroz, 1998, Heijmans, 1998). Nonlinear wavelets are characterized by a reconstruction stage of the form:

$$x[n] = R[\{d_{m,n'}\}, \{c_{M,n'}\}] \qquad \textbf{(B2)}$$

We will limit the discussion to giving a number of nonlinear constructions.

Said and Pearlman wavelets

We will first examine the nonlinear wavelet construction based only on the so-called S transform.

In a first step, the average of two successive values is computed and rounded off to the next integer:

$$l[n] = \left\lfloor \frac{x(2n) + x(2n+1)}{2} \right\rfloor \quad \textbf{(B3)}$$

The high-frequency part of the signal is given by

$$h[n] = x(2n+1) - x(2n) \quad \textbf{(B4)}$$

The inverse transformation is given by

$$x[2n] = x_l[n] + x_h[n] \quad \textbf{(B5)}$$

with

$$x_l[2n] = l[n]$$

$$x_l[2n+1] = l[n]$$

$$x_h[2n] = \left\lceil \frac{h[n]+1}{2} \right\rceil$$

$$x_h[2n+1] = -\left\lceil \frac{h[n]}{2} \right\rceil$$

The rounding off is obviously a nonlinear operation. It has the great advantage to require only integer values. The S stage is therefore well suited to an efficient computation in a microprocessor. The compression of an S-transformed image does not give convincing results, due to aliasing. For that reason, Said and Pearlman (1996) did introduce a first transform, the P transform, to suppress aliasing. The P transform is also invertible.

Morphological Haar wavelets

The morphological Haar wavelet uses the max operator (Heijmans, 1999). The wavelet decomposition is carried out by using two operators. The first operator corresponds to the approximation stage in the linear wavelet. It is given by

$$\chi_1[n] = \max(x[2n], x[2n+1]) \quad \textbf{(B6)}$$

The second operator is the equivalent of the high-pass filter for the nonlinear case.

$$\delta_1[n] = x[2n] - x[2n+1] \quad \textbf{(B7)}$$

The decomposition stage is invertible, allowing a lossless signal decomposition. Morphological Haar wavelets preserve the edges of objects better than Haar wavelets. The Haar wavelet tends to smooth out edges, while morphological Haar wavelets preserve the edge, as the maxima are preserved by

the max operator. More precisely, global maxima are kept, while local maxima may be removed by a higher local maxima at lower resolution. The morphological Haar wavelets can be generalized to higher dimensions. Also the procedure can be extended to more complex decompositions. The general max-lifting scheme is such a method, a promising method for segmentation problems in combination to thresholding methods.

Wavelets constructions for genetic algorithms

The wavelet-based genetic algorithm in part 8 was specifics to binary coding. In this annex, we will show that results similar to the ones in part 8 can also be obtained without making the assumption of binary coding. A slightly modified genetic algorithm is used and explained within the framework of a nonlinear wavelet model. Let us start by describing a single stage of the nonlinear wavelet decomposition.

Consider a string $(a_1,...,a_n)$ with $n=3^J$. The nonlinear wavelet decomposition transforms a triplet (a_1,a_2,a_3) according to table I.

triplet	Approximation coefficient	Detail coefficients
(0,0,0)	0	(0,0)
(1,0,0)	0	(1,0)
(0,1,0)	0	(0,1)
(0,0,1)	0	(1,1)
(1,1,1)	1	(0,0)
(0,1,1)	1	(1,0)
(1,0,1)	1	(0,1)
(1,1,0)	1	(1,1)

Table I

The approximation coefficients is related to the number of *0* and *1* in the triplet. If there is a majority of bits set to one, the approximation coefficient is one, while if the majority of bits is set to zero, the approximation coefficient is zero. The detail coefficient indexes the position of the minority bit if any. The wavelet decomposition is invertible as seen from table I. Further if all detail coefficients are set to (0,0), then the original signal is given by the string (1,...,1).

A nonlinear wavelet decomposition is obtained by cascading the nonlinear wavelet decomposition given in table I, as depicted in fig. B1.

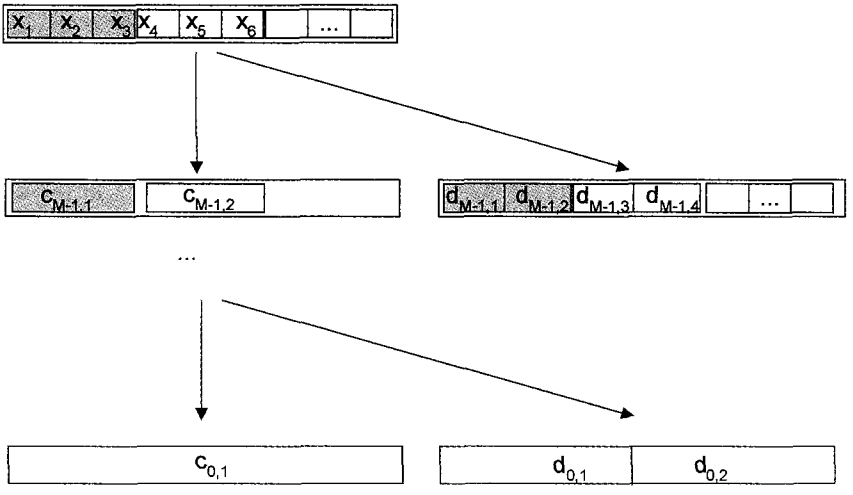


Figure B1: Nonlinear wavelet decomposition defined by table I.

The wavelet-based genetic algorithm works quite similarly to the wavelet-based algorithm in part 8. Strings are first reproduced according to their fitness. The operator O transforms the strings according to the following scheme: A string is kept unchanged with probability P_{Le} with Le corresponding to the number of decomposition levels. With probability $P_{Le\text{-}M}$, the low resolution part of the string at level $P_{Le\text{-}M}$ ($Le \geq M$) is kept and the wavelet coefficients at levels Le,..Le-M are randomly chosen. The new string is obtained using the wavelet reconstruction algorithm defined in table I.

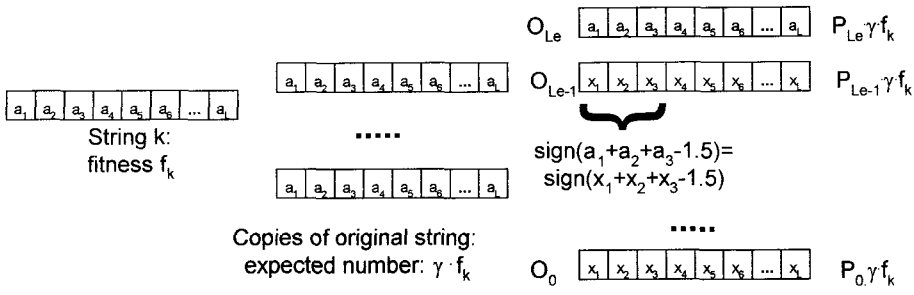


Figure B2: Wavelet-based genetic algorithm based on the wavelet decomposition in table I.

Similarly to the approach in part 8, the expectation of a string, after the first generation, can be computed from the wavelet and approximation coefficients of the wavelet decompositions. One obtains:

$$E^{(1)}(a = a_1,...,a_k) = \gamma \cdot (\sum_{m=0}^{L} c_{m,n} \cdot P_m \cdot H_{m,n}(a)) \qquad \textbf{(B8)}$$

with $c_{m,n}$ the approximation coefficients and $H_{m,n}$ the nonlinear low-pass projection defined by the left column in table I.

This page is intentionally left blank

References

Abbod, M.F., Linkens, D.A., "Anaesthesia monitoring and control using fuzzy logic fusion," Biomedical Engineering, Applications Basis Communications **10**, 225-35 (1998).

Abramovich F., Bailey T., Sapatinas T., "Wavelet analysis and its statistical applications," The Statistician—Journal of the Royal Statistical Society **D 49**, 1-29 (2000).

An, P.E., Harris, C.J., "An intelligent driver warning system for vehicle collision avoidance," IEEE Trans. Systems, Man and Cybernetics **26**, 254-61 (1996).

Antoniadis, A. and Grégoire, G. McKeague, I., "Wavelet methods for curve estimation," J. Am. Stat. Assoc. **89**, 1340-53, (1994).

Antoniadis, A., "Wavelets in statistics: a review," J. Italian Stat. Soc. **6**, 97-130 (1997).

Ashenayi, K., "A review of applications of intelligent systems technology in power engineering," Proc. 30th Annual Frontiers of Power Conference, Stillwater 27-28 Oct. 1997, XIII-11 (1997).

Babuska, R., *Fuzzy modeling for control*, Kluver Academic Press, Boston (1998).

Bakshi, B.R.., Koulanis, A.K., Stephanopoulos, G., "Wave-nets: novel learning techniques, and the induction of physically interpretable models," SPIE **2242**, 637-48 (1994).

Bakshi, B.R., "Multiscale analysis and modeling using wavelets," J. Chemometrics **13**, 415-34 (1999).

Baldi, P and Hornik, K., "Neural networks and principal component analysis: learning from examples without local minima," Neural Networks **2**, 53-58 (1989).

Baras, J.S., Wolk, S.I., "Model based automatic recognition from high range resolution radar returns," Proc. SPIE **2234**, 57-66 (1994).

Bartels, R.H, Beatty, J.C., and Barsky,B.A., *Splines for use in computer graphics*, Morgan Kaufmann (1987).

Battle, G., "A block spin construction of ondelettes. part I: Lemarié functions," Comm. Math. Phys. **110**, 601-15 (1987).

Bellman, R., *Adaptive control Processes: a guided Tour*, Princeton University Press (1961).

Berger, J., R. Coifman, R., Goldberg, M., "Removing noise from music using local trigonometric bases and wavelet packets, "J. Audio Eng. Soc. **42**, 808-18 (1994).

Bethke, A.D., "Genetic algorithms as function optimizers," PhD. Work, Michigan University (1981).

Betti, A., Barni, M, Mecocci, A., "Using a wavelet-based fractal feature to improve texture discrimination on SAR images," Proc Int. Conf. on Image Processing, IEEE Comp. Soc., Los Alamitos, vol. 1, 251-54 (1997).

Billings, S.A., Coca, D., "Discrete wavelet models for identification and qualitative analysis of chaotic systems," International Journal of Bifurcation and Chaos in Applied Sciences and Engineering **9**, 1263-84 (1999).

Bezdek, J., *Pattern recognition with fuzzy objective function algorithms*, Plenum Press (1981).

Bossley, K.M., *Neurofuzzy modelling approaches in system identification,* Ph.D Thesis, University of Southampton (1997).

Boubez, T.I., Peskin, R.L., "Wavelet neural networks and receptive field partitioning," Proc. 1993 IEEE Int. Conf. on Neural Networks (ICNN'93), San Francisco, vol. 3, 1544-49 (1993).

Breiman, L.J., Friedman,J., Olshen, R., Stone, C., *Classification and regression trees* , Wadsworth (1984) .

Brislawn, C., "Fingerprints go digital," Notices AMS **42**, 1278-83 (1995).

Brito, A.E., Kosheleva, O.M., Cabrera, S.D., "Multi-resolution data processing is optimal: case study of detecting surface mounted devices," Proc. Conf. on Intelligent Systems and Semiotics: A Learning Perspective, Gaitersburg, 22-25 Sept. 1997, 157-61 (1997).

Brown, M., Harris, C, *Neurofuzzy adaptive modelling and control*, Prentice Hall, New York (1994).

Buhmann, M., "Pre-wavelets on scattered knots and from radial function spaces: a review," Mathematics of Surfaces VI, G. Mullineux, ed., IMA Conference Proceedings Series, Oxford University Press, Oxford, 309-24 (1996), also Research Report N° 94-08, Eidgenössische Technische Hochschule, Switzerland: http://www.sam.math.ethz.ch/Reports/1994-08.html.

Burke Hubbard, B., *The world according to wavelets*, A K Peters, Wellesley, MA (1996).

Cao, L., Hong, Y., Fang, H., He, G., "Predicting chaotic time series with wavelet networks," Physica D, 225-38 (1995).

Cao, L., Hong, Y., Zhao, H., Deng, S., "Predicting economic time series using a nonlinear deterministic technique," Computational Economics **9**, 149-78 (1996).

Carr, J.C., Gee, A.H., Prager, R.W. and Dalton, K.J., "Quantitative visualization of surfaces from volumetric data," Proc. WSGC'98-The sixth int. Conf. in central europe on computer graphics and visulization, Plzen, 57-64 (1998).

Chalermwat, P., El- Ghazawi, T., "Multi-resolution image registration using genetics", Proc. 1999 Int. Conf. on Image Proc. Kobe, Japan, 24-28 Oct. 1999, vol. 2, 452-6 (1999).

Chang, C.S., Weihui Fu, Minjun Yi, "Short term load forecasting using wavelet networks," Engineering Intelligent Systems for Electrical Engineering and Communications **6**, 217-23 (1998).

Chang, P.R., Yeh, B.F., "Nonlinear communication channel equalization using wavelet neural network," Proc. IEEE Int. Conf. on Neural Networks: IEEE World Congress on Computational Intelligence, IEEE, 3605-10 (1994).

Cheng, D.C., Cheng, K., "Multiresolution based fuzzy c-means clustering for brain hemorrhage analysis," Proc. 2nd Int. Conf. on Bioelectromagnetism, IEEE, New York, 35-36 (1998).

Christopoulos, C., Skodras, A., "JPEG 2000 the next generation still image compression standard," Tutorial IEEE Int. Conf. on Image Proc. (ICIP'99), Kobe (1999).

Chui, C.K., *An introduction to wavelets*, Academic Press, New York (1992).

Claypole, R., Davis, G., Sweldens, W., Baraniuk, R, "Nonlinear wavelet transforms for image coding," Proc. 31st Asilomar Conf. Signals, Systems, and Computers **1**, 662-67 (1997).

Cohen, A., Daubechies, I., Feauveau J.C., "Biorthogonal bases of compactly supported wavelets," Commun. On Pure and Applied Math. **45**, 485-560 (1992).

Coifman, R.R., Wickerhauser, M.V., "Entropy-based algorithms for best basis selection," IEEE Trans. Info Theory **38**, 713-18 (1992).

Croisier, A., Esteban, D., Galand C., "Perfect channel splitting by use of interpolation/-decimation/ tree decomposition techniques," Proc.. Conf. on Inform. Sciences and Systems, Patras August 1976, 443-46 (1976).

Daubechies, I., Sweldens, W., "Factoring wavelet transforms into lifting steps," J. Fourier Anal. Appl. **4**, 247-69 (1998).

Daubechies, I., *Ten lectures on wavelets*, SIAM, Philadelphia (1992).

Daugmann,J., "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," IEEE Trans. Acoust., Speech, Signal Proc. **36**, 1169-79 (1988).

Davis, G., Mallat, S., Zhang, Z., "Adaptive time-frequency decompositions," Opt. Eng. **33**, 2183-91 (1994).

Dawn, T., "How to tell a click from a burp from a whistle (wavelet analysis)," Noise and Vibration Worldwide **24**, 12-14 (1993).

de Boor, C., *A practical guide to splines*, Springer-Verlag (1978).

de Vel, O., S. Wangsuya, D. Coomans, "On thai character recognition," Proc. IEEE Int. Conf. On Neural Networks, IEEE, New York, vol. 4, 2095-98 (1995).

Deschenes, C.J., Noonan, J., "Fuzzy Kohonen network for the classification of transients using the wavelet transform for feature extraction," Information Sciences **87**, 247-66 (1995).

Diaconis, P. and Freedman, D., "Asymptotics of graphical projection pursuit," Annals Stat. **12**, 793-815 (1984).

Dickhaus, H., Heinrich, H., "Identification of high risks patients in cardiology by wavelet networks," Proc. 18th IEEE Engineering in Medicine and Biology, 31 Oct-3 Nov. 1997, IEEE, vol. 3, 923-24 (1996).

Diercks, P., *Curve and surface fitting with splines* , Oxford Press (1995).

Donoho, D, Johnstone, I., "Ideal denoising in orthonormal basis chosen from a library of bases," C.R. Acad. Sci. **39**, Serie I, Paris, 1317-22 (1994).

Doyle, R.S., Harris, C.J., "Multisensor data fusion for helicopter guidance using neurofuzzy estimation algorithms," Royal Aeronautical Society Journal, July, 241-57 (1996).

Echauz J. and Vachtsevanos G., "Elliptic and radial wavelet neural networks," in Proc. Second World Automation Congress (WAC'96), Montpellier, France, vol. 5, 173-79 (1996).

Echauz, J., "Strategies for fast training of wavelet neural networks," 2nd International Symposium on Soft Computing for Industry, 3rd World Automation Congress (WAC'98), Anchorage, Alaska, May 10-14, 1-6 (1998).

Egger, O., Li, W., Kunt, M., "High compression image coding using an adaptive morphological subband decomposition," Proc. IEEE **83**, 272-87 (1995).

Eubank, R.L., *Nonparametric regression and spline* , Statistics: Textbook and Monographs 157, Marcel Dekker, New York (1999).

Feichtinger, H.G., "Coherent frames and irregular sampling," in J.S. Byrnes and J.L. Byrnes, editors, Proc. Conf. Recent Advances in Fourier Anal. and Its Appl. NATO ASI Series C, Vol. .315, 427-40. Kluwer Acad. Publ. (1989).

Feng, G.C., Yuen, P.C., Dai, D.Q., "Human face recognition using PCA on wavelet subband," J. Electronic Imaging **9**, 226-33 (2000).

Fernando Marar, J., Carvalho Filho, E.C.B., Vasconcelos, G.C., "Function approximation by polynomial wavelets generated from powers of sigmoids," SPIE **2762**, 365-74. (1996).

Flehmig, F., v. Watzdorf, R., Marquardt, W., "Identification of trends in process measurements using the wavelet transform," Computers & Chemical Engineering 22, 491-6 (1998).

Foltyniewicz, R., Cichocki, A., "Higher order neural networks with wavelet preprocessing for face recognition," Proc. WCNN'96, 919-22 (1996).

Fonseca L.M.G., B.S. Manjunath, B.S., "Registration techniques for multisensor remotely sensed imagery," Journal of Photogrammetry Engineering & Remote Sensing **62**, 1049-56 (1996).

Friedman, J. H., Stuetzle, W., "Projection pursuit regression," J. Am. Stat. Assoc. **76**, 817-23 (1981).

Gabor, D., "Theory of communication," J. IEE 93, 429-57 (1946).

Gan, Q. and Harris, C.J., "Fuzzy local linearisation and local basis function expansion in nonlinear system modelling," IEEE Trans. On Systems, Man, and Cybernetics **B29**, 559-65 (1999).

Gantmacher, F.R., *The theory of matrices*, vol. 2, Chelsea, New York (1977).

Garay, M. R., Johnson, D.S., *Computers and intractability: a guide to the theory of NP-completeness*, W.H. Freeman and Co., New York (1979).

Goldberg, D.E., "Genetic Algorithms and Walsh functions: part I, a gentle introduction," Complex Systems **3**, 129-52 (1989).

Goldberg, D.E., *Genetic algorithms*, Addison-Wesley, USA (1991).

Grossmann, A., Morlet, J., "Decomposition of Hardy functions into square integrable wavelets of constant shape," SIAM J. Math. Anal. **15**, 723-36 (1984).

Haar, A, "Zur Theorie der orthogonalen Funktionensysteme," Math. Annual. **69**, 331-71 (1910).

Halgamuge, S.K., Herrmann, C.S., Jain, L., "Analysis of EEG signals with wavelets and knowledge engineering techniques," Proc. Int. Conf. on Neural Information Processing vol.2, Springer-Verlag, Singapore, 1381-86 (1996).

Harris, C.J., Wu Z.Q, Gan, Q., "Neurofuzzy state estimators and their applications," Annual Reviews in Control **23**, 149-58 (1999a).

Harris, C.J., Xia, H., Wilson, P.A., "A practical intelligent guidance and control system for ship obstacle avoidance," Journal Systems and Control Engineering **213**, 311-20 (1999b).

Hazarika, N., Chen, J.Z., Tsoi, A.C., Segejew, A., "Classification of EEG signals using the wavelet transform," Proc. 13th Int. Conf. On Digital Signal Processing, DSP'97, IEEE, New York, vol. 1, 89-92 (1997).

Heijmans, H.J.A.M, Goutsias, J., "Constructing morphological wavelets with the lifting scheme," Proc. PRIP'99, Minsk, May 18-20, 1999, 65-72 (1999).

Holland, J.H., *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor (1975).

Holmes, C.C., Mallick, B.K., "Bayesian wavelet networks for nonparametric regression," IEEE Trans. Neural Networks 11, 27-35 (2000).

Horn, J., Goldberg, D.E., "Genetic algorithm difficulty and the modality of fitness landscapes," Foundations of Genetic Algorithms 3, ed. L. D. Whitley and M. D. Rose, Morgan Kaufmann, San Francisco, CA, 243-69 (1995).

Hornik, K., "Multilayer feedforward networks are universal approximators," Neural Networks **2**, 359-66 (1989).

Huber, P.J., " Projection pursuit," Annals Stat. **13**, 435-75 (1985).

Intrator, N., "Feature extraction using an unsupervised neural network," in Touretzky D. S., Ellman J.L., Sejnowski T. J., and Hinton G.E., editors, Proc. 1990 Connectionist Models Summer School, 310-18. Morgan Kaufmann. (1992).

Jang, J.-S. R., Sun, C, Tmizutani, E., *Neuro-fuzzy and soft computing*, Prentice Hall, Upper Saddle River (1997).

Jun Li, Yueqin Zhou, Deren Li, "PCA and wavelet transform for fusing panchromatic and multi-spectral images," Proc. SPIE **3719**, 369-77 (1999).

Juuso, E, Leiviska, K., "Linguistic equations in system development for computational intelligence," Proc. EUFIT'96, Fourth European Congress on Intelligent Techniques and Soft Computing, Sept.2-5,1996, Aachen, Ed. H.-J. Zimmermann, Mainz Verlag, Vol. 2, 1127-31 (1996).

Juuso, E., Jarvensivu, M., "Lime kiln process modelling with neural networks and linguistic equations," Proc. EUFIT'98, Sixth European Congress on Intelligent Techniques and Soft Computing, Sept.7-10,1998, Aachen, Ed. H.-J. Zimmermann, Mainz Verlag, Vol. 3, 1601-05 (1998).

Kaiser, G., *A friendly guide to wavelets*, Birkhäuser, Boston (1994).

Kamarthi, S.V., Pittner, S., "Fourier and wavelet transform for flank wear estimation- a comparison," Mechanical Systems and Signal Processing **11**, 791-809 (1997).

Katic, D., Vukobratovic, M., "Wavelet neural network approach for control of non-contact and contact robotic tasks," Proc. IEEE Symposium on Intelligent Control, 16-18 July 1997, Istambul, IEEE, 245-50 (1997).

Kavli, T., "ASMOD-an Algorithm for adaptive spline modelling of observation data," in Advances in Intelligent Control, ed. by C.J. Harris, Taylor & Francis, Bristol, 141-62 (1994).

Kishikawa, Y., Tokinaga, S., "Prediction of stock trends by using the wavelet transform and the multi-stage fuzzy interference system optimized by the GA, "IEICE Trans. On Fundamentals of Electronics, Communications and Computer Sciences **E83-A**, 357-66 (2000).

Ko, H., Berko, F., Telfer, B., Garcia,J., "Image enhancements using wavelet preprocessing for printed circuit board classification," Proc. SPIE **2491**, 473-80 (1995).

Kosko, B., *Neural networks and fuzzy systems*, Prentice-Hall, Englewood Cliff (1992).

Kovacevic, J., W. Sweldens, "Wavelet families of increasing order in arbitrary dimensions," submitted IEEE Trans. On Image Processing (1997). also http://cm.bell-labs.com/who/wim/papers/papers.html#mdlift

Kreinovich, V., Sirisaengtaksin, O., Cabrera, S., "Wavelet neural networks are asymptotically optimal approximators for functions of one variable," IEEE, 299-304 (1994).

Kruger, V., Happe, A., Sommer, G., "Affine real-time face tracking using a wavelet network," Proc. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 26-27 Sept. 1999, Corfu, IEEE, 141-48 (1999).

Kugarajah, T. and Q. Zhang, "Multidimensional wavelet frames," IEEE Transactions on Neural Networks **6**, 1552- 56 (1995).

Kullback S., Leibler, R.A., "On information and sufficiency," Ann. Math. Statist. **22**, 79-86 (1951).

Kumar, P., Foufoula-Georgiou, E., "Wavelet analysis for geophysical applications," Reviews of Geophysics **35**, 385-412 (1997).

Kunt, T.A., McAvoy, T.J., Cavicchi, R.E., Semancik, S., "Optimization of temperature programmed sensing for gas identification using micro-hotplate sensors," Sensors and Actuators **B53**, 24-43 (1998).

Lankhorst, M.M., van der Laan, M.D., "Wavelet-based signal approximation with genetic algorithms," in Evolutionary Programming IV. Proceedings of the Fourth Annual Conference on Evolutionary Programming San Diego, USA, 1-3 March 1995, ed. McDonnell, J.R., Reynolds, R.G., Fogel, D.B., MIT Press, 237-55 (1995).

Lee, D., "An application of wavelet networks in condition monitoring," IEEE Power Engineering Review **19**, 69-70 (1999).

Leiviska, K., Juuso, E., "Modelling of industrial processes using linguistic equations: lime kiln as an example," Proc. EUFIT'96, Fourth European Congress on Intelligent Techniques and Soft Computing, Sept.2-5,1996 Aachen, Ed. H.-J. Zimmermann, Mainz Verlag, Vol. 3, 1919-23 (1996).

Lemarié, P.G., "Ondelettes à localisation exponentielle," J. Math. Pures et Appl. **67**, 227-36 (1988).

Leung, A.K., Foo-Tim Chau, Jun- Bin Gao, "A review on applications of wavelet transform techniques in chemical analysis," Chemometrics and Intelligent Laboratory Systems **43**, 165-84 (1998).

Li, H., Manjunath, B.S., Mitra, S.K., "A contour based approach to multisensor image registration," IEEE Trans. Image Processing, **4**, 320-34 (1995a).

Li, H., Manjunath, B.S., Mitra, S.K., "Multisensor image fusion using the wavelet transformation," Graphical Models and Image Processing **57**, 235-45 (1995b).

Li. L., Wang,W., Wang, D., "A new wavelet network architecture for echo cancellation," Proc. 1996 IEEE TENCON 2, IEEE, New York, 598-601 (1996).

Liao, B.Y., Pan, J.S., Wang, J.W., Hong, L., "2-D non-separable wavelet bases for texture classification with genetic feature selection," Proc. IAPR workshop on Machine Vision Applications, Chiba, Japan, 17-19 Nov. 1998, 258-61 (1998).

Linkens, D.A., Abbod, M.F., Backory, J.K., "Closed-loop control of depth of anaesthesia: a simulation study using auditory evoked responses," Control Engineering Practice **5**, 1717-26 (1997).

Linkens, D.A.; Abbod, M.F., "Intelligent control of anaesthesia," Proc. IEE Colloquium Intelligent Methods in Healthcare and Medical Applications (Digest 1998/514), Oct. 20, York, p.2/1-4 (1998).

Magli, E., Olmo, G., Lo Presti, L., "Pattern recognition by means of the Radon transform and the continuous wavelet transform," Signal Proc. **73**, 277-89 (1999).

Mallat, S., "A theory for multiresolution signal decomposition: the wavelet representation," IEEE Trans. Patt. Recog. And Mach. Intell. **11**, 674-93 (1989).

Mallat, S., *A wavelet tour of signal processing*, Academic Press, San Diego (1998).

Mallat, S., Zhang, Z., "Matching pursuits with time-frequency dictionaries," IEEE Trans. On Signal Proc. **41**, 3397-3415 (1993).

Meyer, Y., *Wavelets and operators*, Cambridge University Press (1992).

Micchelli, C. A., Rabut, C., Utreras, F. I., "Using the refinement equation for the construction of pre-wavelets III: elliptic splines," Num. Algorithms **1**, 331-52 (1991).

Miller, B.E., Colgate, J.E., "Using a wavelet network to characterize real environments for haptic display," Proc. ASME Dynamic Systems and Control Division, 15-20 Nov. 1998, Anaheim, ed. Furness, R.J., ASME, New York, 257-64 (1998).

Moreira-Tamayo, O., Pineda de Gyvez, J., "Preprocessing operators for image compression using cellular neural networks," Proc. Int. Conf. on Neural Networks, IEEE, New York, vol. 3, 1500-05 (1996).

Naghdy, G., Turgut,A., "Evolutionary procedure for the optimization of a generic texture classifier," 1997 IEEE Int. Conf. on Intelligent Processing Systems, IEEE, New York, vol. 1, 574-78 (1997).

Nauck, D., Klawonn, F., Kruse, R., *Foundations of neuro-fuzzy systems*, John Wiley & Sons, Chichester (1997).

Nie, Junhong, Linkens, D. A, *Fuzzy-neural control : principles, algorithms and applications*, Prentice Hall (1995).

Oja,E., "A simplified neuron model as principal component analyzer," J. Math. Biology **15**, 267-73 (1982).

Okabe, A., Boots, B., Sugihara, K., Chiu, S.N., *Spatial tessellations: concepts and applications of Voronoi diagrams* , John Wiley (1999).

Okimoto, G., Lemonds, D., "Principal component analysis in the wavelet domain: new features for underwater object recognition," Proc. SPIE **3710**, 697-708 (1999).

Padgett, M.L., Roppel, T.A., Johnson, J.L., "Pulse coupled neural networks (PCNN) wavelets and radial basis functions: olfactory sensor applications," Proc. 1998 IEEE Int. Joint Conf. on Neural Networks, IEEE world Congress on Computational Intelligence, vol. 3, IEEE, New York, 1784-89 (1998).

Park, Y, Chao,T.C. , "Automatic target recognition processor using an optical wavelet preprocessor and an electronic neural classifier," Proc. SPIE **3073**, 299-307 (1997).

Pati, Y.C., Krishnaprasad, P.S., "Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations," IEEE Trans. On Neural Networks **4**, 73-85 (1992).

Pati, Y.C., Krishnaprasad, P.S., "Discrete affine wavelet transforms for analysis and synthesis of feedforward neural networks," Advances in Neural Information Processing Systems **3**, 743-49 (1991).

Pham, V.L., Wong, K.P., "Wavelet-transform-based algorithm for harmonic analysis of power system waveforms," IEE Proc.-Generation, Transmission and Distribution **146**, 249-54 (1999).

Pittner, S., Kamarthi, S.V, Ginglan Gao, "Wavelet networks for sensor signal classification in flank wear assessment," J. Intelligent Manufacturing **9**, 315-22 (1998).

Pratt, L.Y., Misra M., Farris, C., Hansen, R.O., "Interpolation, wavelet compression, and neural network analysis for hazardous waste characterization," Proc. 1995 Int. Conf. on Systems, Man and Cybernetics. Intelligent Systems for the 21st century, IEEE New York, vol. 3, 2058-63 (1995).

Prochazka, A., Sys, V., "Times series prediction using genetically trained wavelet networks," Neural networks for signal processing IV : proceedings of the 1994 IEEE Workshop, IEEE, 195-202 (1994).

Przylucky, S.W., Surtel, W., "The wavelet transform preprocessing for data analysis in self-organizing artificial neural networks," Proc. Fourth Int. Symposium on Methods and Models in Automation and Robotics, Szczecin Poland, vol. 2, 751-56 (1997).

Qian, W., Clarke, L.P., Kallergi, M., Venugopal, P., Clark, R., Silbiger, M., Zheng, B., "Applications of wavelet transform for image enhancement in medical imaging," Proc Intelligent Enginnering Systems Through Artificial Neural Networks 4, ASME, New York, 651-60 (1994).

Queiroz, R.L, Florencio, D.A.F., Schafer, R.W., "Nonexpansive pyramid for image coding using a nonlinear filterbank," IEEE Trans. Image Processing **7**, 246-52 (1998).

Rao, S.S., and Kumthekar, B., "Recurrent wavelet networks," Neural networks for signal processing III : proceedings of the 1993 IEEE-SP Workshop IEEE, 3143-47 (1993).

Reeves, C.R., "Characterising and Searchng Fitness Landscapes," Seventh European Congress on Intelligent Techniques and Soft Computing, Sept.13-16,1999, Aachen, CD Proc. (1999).

Roya, S., Heckendorn, R.B., Whitley, D., "A tractable Walsh analysis of SAT and its implications for genetic algorithms," Proc. AAAI-IAAI, 392-97 (1998).

Said, A., Pearlman, W.A., "An image multiresolution representation for lossless and lossy compression," IEEE Trans. Signal Processing **5**, 1637-50 (1996).

Saito, N., "Local feature extraction and its applications using a library of bases," Ph.D thesis, Dept. Of Mathematics, Yale University (1994a).

Saito, N., Beylkin, "Multiresolution representations using the auto-correlation functions of compactly supported wavelets," Proc. ICASSP-92, vol. 4, IEEE, 381-84 (1992).

Saito, N., Coifman, R.R., "Local discriminant bases," in Mathematical Imaging: Wavelet Applications in Signal and Image Processing II, Eds. A.F. Laine and M.A. Unser, Proc. SPIE **2303** (1994b).

Samatsu, T., Uchino, E., Yamakawa, T., "Feature extraction of a vectorcardiogram by employing a wavelet network guaranteeing a global minimum," J. Intelligent & Fuzzy Systems **8**, 221-7 (2000).

Samuel, P., Pines, D., "Health monitoring/damage detection of a rotorcraft planetary geartrain system using piezoelectric sensors," Proc. SPIE **3041**, 44-53 (1997).

Sanchez-Redondo, J.L, Zufiria, P.J., "Function estimation and system estimation via wavenets. Applications to aircraft velocity estimation," Proc. EANN'98, 10-12 June 1998, Turku, ed. Bulsari, A.B., Fernandez de Canete, J., Kallio, S., 312-19 (1998).

Sarty, G.E., Kendall, E.J., "Improved T_2 and diffusion maps from wavelet de-noised magnetic resonance imaging data," Ed. H. Boom, Proc. 18th Annual Int. Conf. Of the IEEE Engineering in Medicine and Biology Society, IEEE, New York, vol. 3, 1113-14 (1997).

Schoenberg I.J., Whitney A., "On Pólya frequency functions III," Trans. Am. Math. **74**, 246-59 (1953).

Schoenberg, I.J., "Contribution to the problem of approximation of equidistant data by analytic functions," Quart. Appl. Math. **4**, 45-99, 112-41 (1946).

Schoonewelle, H., Van der Hagen, T.H.J.J., Hoogenboom, J.E., "Process monitoring by combining several signal-analysis results using fuzzy logic," Proc. 2nd FLINS Workshop, Mol 25-27 Mai 1996, World Scientific, Singapore, 316-22 (1996).

Serrano, I., Lazaro, A., Oria, J.P., "Ultrasonic inspection of foundry pieces applying wavelet transform analysis," Proc. 1999 IEEE Symposium on Intelligent Control, Intelligent Systems and Semiotics, Cambridge, 15-17 Sept 1999, IEEE, 375-80 (1999).

Shaikh, M.A., Tian, B., Azimi-Sadjadi, Eis, K.E., VonderHaar, T.H., "An automatic neural network-based cloud detection/classification scheme using multispectral and textural features," Proc. SPIE **2758**, 51-61 (1996).

Shark, L.-K., Yu, C., "Denoising by optimal fuzzy thresholding in wavelet domain," Electronics Lett. **36**, 581-2 (2000).

Shashidhara, H.L., Suneel, T.S., Gadre, V.M., Pande, S.S., Lohani, S., "Intelligent CNC turning using wavelet-neural networks," Proc. Second World Manufacturing Congress WMC'99, Durnham, UK, 27-30 Sept. 1999, ed. Nahavandi, S., Saadat, M., 525-31 (1999).

Shmilovici, A., Maimon, O., "Fuzzy systems approximation by frames-SISO case," Proc. 1995 IEEE International Conference on Fuzzy Systems, IEEE, New York, 2057-62 (1995).

Shmilovici, A., Maimon, O., "Best fuzzy rule selection with orthogonal matching pursuit," Proc. Fourth European Congress on Intelligent Techniques and Soft Computing EUFIT '96, 592-96 (1996).

Shmilovici, A., Maimon, O., "On the solution of differential equations with fuzzy spline wavelets ," Fuzzy Sets and Systems **96**, 77-99 (1998).

Shmilovici, A., Maimon, O., "Systems identification with fuzzy spline wavelets," Proc. Sixth IEEE Int. Conf. on Fuzzy Systems, vol.1, IEEE: New York, 299-304 (1997).

Staszewski, W.J., "Identification of damping in MDOF systems using time-scale decomposition," J. Sound and Vibration **203**, 283-305 (1997).

Sweldens, W., and Schröder, P., "Building your own wavelets at home," Tech. Report. IMI 1995:5, Dept. Of Mathematics, University of South Carolina (1995).

Swiercz, M., Mariak, Z., Lewko, J., Chojnacki, J.K., Kozlowski, A., Piekarski, P., "Neural network technique for detecting emergency states in neurosurgical patients," Med. & Biol. Eng. Comp. **36**, 717-22 (1998).

Szu, H., "Review of wavelet transforms for pattern recognitions," Proc. SPIE **2762**, 2-22 (1996a).

Szu, H., Hsu, C., Da-Hong Xie, "Continuous speech segmentation determined by blind source separation," Proc. SPIE **3391**, 396-408 (1998a).

Szu, H., Hsu, C., Yamakawa, T., "Image independent component analysis via wavelet subbands," Proc. 5th Conference on Soft Computing and Information/Intelligent Systems. Vol. 1, Fukuoka, 16-20 Oct 1998, 135-38 (1998b).

Szu, H., Telfer, B., Garcia, J., "Wavelet transforms and neural networks for compression and recognition," Neural Networks **9**, 695-708 (1996b).

Szu, H., Telfer, B., Kadambe, S., "Neural network adaptive wavelets for signal representation and classification," Opt. Engineering **31**, 1907-16 (1992).

Taeksoo Shin, Ingoo Han, "Optimal signal multi-resolution by genetic algorithms to support artificial neural networks for exchange-rate forecasting," Expert Systems with Applications **18**, 257-69 (2000).

Tagliarini, G.,Page, E.,Karlsen, R.,Gerhart,G., "Genetic algorithms for adaptive wavelet design," Proc. SPIE **2762**, 82-93 (1996).

Takagi, T. and Sugeno, M., "Fuzzy identification of systems and its applications to modeling and control," IEEE Trans. Syst. Man, Cybern. **15**,116-32 (1985).

Tang, Y.Y., Liu, J., Yang, L.H., Ma, H., *Wavelet Theory and its application to pattern recognition*, Series in Machine Perception and Artificial Intelligence, vol 36, World Scientific (2000).

Thomas, J.H., Dubuisson, B., Dillies-Peltier, M.A., "Engine knock detection from vibration signals using pattern recognition," Meccanica **32**, 431-39 (1996).

Thonet, G., Blanc, O., Vandergheynst, P., Pruvot, E., Vesin, J.M., Antoine, J.-P., "Wavelet-based detection of ventricular ectopic heart rate signals," Appl. Signal Proc. **5**, 170-81 (1998).

Thuillard, M., "A new flame detector using the latest research on flames and fuzzy-wavelet algorithms," Proc. AUBE'99, Duisburg, Ed. Luck, H., 170-79 (1999c).

Thuillard, M., "Adaptive fuzzy-wavelet modelling," Proc. European Symposium on Intelligent Techniques ESIT'97, Bari, 244-50 (1997).

Thuillard, M., "Applications of wavelets and wavenets in soft computing illustrated with the example of fire detectors," SPIE Wavelet Applications VII, April 24-28 2000, Orlando, Proc. SPIE **4056**, 351-61 (2000a).

Thuillard, M., "Combination of Fuzzy Logic and Wavelet Theory into a new Development Tool," L. Yliniemi, E. Juuso (eds.), Proceedings of TOOLMET'97—Tool Environments and Development Methods for Intelligent Systems, Oulu, April 17-18 1997, 179-83 (1997b).

Thuillard, M., "Fuzzy logic in the wavelet framework," Proc. Toolmet'2000 —Tool Environments and Development Methods for Intelligent Systems, April 13-14 2000 ", L. Yliniemi, E. Juuso (eds.), Oulu, 15-36 (2000b).

Thuillard, M., "Fuzzy wavenets: an adaptive, multiresolution, neurofuzzy learning scheme," Seventh European Congress on Intelligent Techniques and Soft Computing, Sept.13-16,1999, Aachen, Contrib. cc6-1, CD Proc. (1999a).

Thuillard, M., "Fuzzy-wavelets: theory and applications," Proc. EUFIT'98, Sixth European Congress on Intelligent Techniques and Soft Computing, Sept.8-10,1998, Aachen, Ed. H.-J. Zimmermann, Mainz Verlag, Vol. 2, 1149-59 (1998b).

Thuillard, M., "New methods for reducing the number of false alarms in fire detection systems," Fire Technology **30**, 250-68 (1994).

Thuillard, M., "New perspectives for the integration of wavelet theory to soft computing," European Symposium on Intelligent Techniques ESIT'99, June 3-4, 1999, Crete, CD Proc. (1999a).

Thuillard, M., "New Results on the flames'pulsation mechanisms permit to improve the quality of detection of pool fires," Proc. Fire Suppression and Detection Research Application Symposium, Feb. 24-26, 1999, Orlando, Ed. Fire Protection Research Foundation, 171-89 (1999b).

Thuillard, M., "The development of algorithms for a smoke detector with neuro-fuzzy logic," Fuzzy Sets and Systems 77, 117-24 (1996).

Thuillard, M., "Wavelets in preprocessing," Tutorial EUFIT'99 (1999c).

Thuillard, M., "Wavelets in soft computing: theory and real world applications," Tutorial EUFIT'98 (1998a).

Thuillard, M., "Multiresolution learning: from GA to fuzzy-wavenets," Proc. COIL 2000-Symposium on Computational Intelligence and Learning, 22-23 June 2000, Chios, Greece, 67-76 (2000c).

Tolias, Y., Panas, S., Tsoukalas, L.H., "FSMIQ: fuzzy similarity matching for image queries," Proc. Int. Conf. on Information Intelligence and Systems, Bethesda, USA, 31 Oct.-3 Nov. 1999, IEEE Comp. Soc., 249-54 (1999).

Toonstra, J., Kinsner,W., "A radio transmitter fingerprint System ODO-1," Proc. 1996 Canadian Conference on Electrical and Computer Engineering, Ed. T.J. Malkinson, vol.1, 60-63 (1996).

Torrance, C., Compo, G.P., "A practical guide to wavelet analysis," Bull. Amer. Meteor. Soc **79**, 61-78 (1998).

Unser, M., "Splines: a perfect fit for signal/image processing," IEEE Signal Proc. Mag. **16**, N°6, 22-38 (1999).

Unser, M., Aldroubi, A., "A review of wavelets in biomedical applications," Proc. IEEE **84**, 626-38 (1996).

Vetterli, M., "Multidimensional subband coding, some theory and algorithms," Signal Proc. **6**, 97-112 (1984).

Vetterli, M., "Wavelets, approximation and compression-a review," Proc. SPIE **3723**, 28-31 (1999).

Vetterli, M., Herley, C., "Wavelets and filter banks: theory and design," IEEE Trans. Signal Proc. **40**, 2207-32 (1992).

Vetterli, M., J. Kovacevic, *Wavelets and subband coding*, Prentice-Hall, Englewood Cliffs (1995).

Vose, M.D., *The simple genetic algorithm: foundations and theory*, MIT Press, Boston (1999).

Wang, J., Naghdy, G., Ogunbona, P., "A new wavelet based ART network for texture classification," 1996 Austrialian New Zealand Conf. on Intelligent Information Systems (ANZIIS'96), IEEE, New York, 250-53 (1996).

Wang, J., Naghdy, G., Ogunbona, P., "Wavelet-based feature —adaptive resonance theory neural network for texture identification," J. Electr. Imaging 6, 329-36 (1997a).

Wang, S., Chen, G., Sapounas, D., Hongch Shi, Peer, R., "Development of gazing algorithms for tracking oriented recognition," Proc. SPIE **3069**, 37-48 (1997b).

Watson, D.F., Phillip, G.M., "Neighbor-based interpolation," Geobyte **2**, 12-16 (1987).

Westra, R.L., "Adaptive control using CCD-images: towards a template-invariant approach," Proc. COIL 2000-Symposium on Computational Intelligence and Learning, 22-23 June 2000, Chios, Greece, 119-123 (2000c).

Wickerhauser, M.V., "Acoustic signal compression with wavelet packets," in Wavelets : a tutorial in theory and applications, ed. C.K. Chui, Academic Press, New York (1992).

Wickerhauser, M.V., *Adapted wavelet analysis from theory to software*, A.K. Peters (1994).

Wilson, T.A., Rogers, S.K., Broussard, R.P., Rathbun, T.F., "Fusion of focus of attention alternatives for FLIR imagery," Proc. SPIE **2756**, 76-86 (1996).

Yang, Z.J. Sagara, S., Tsuji, T., "System identification using a multiresolution neural network," Automatica **33**, 1345-50 (1997).

Yeung, L.F., Li, X.W., "Multi-input system identification and its applications using wavelet constructive method," Proc. 35th IEEE Conf. On Decison and Control, 11-13 Dec. 1996, Kobe, IEEE, vol. 3, 3230-35 (1996).

Yu, Y., Shaohua Tan, S., "Complementarity and equivalence relationships between convex fuzzy systems with symmetry restrictions and wavelets," Fuzzy Sets and Systems **101**, 423-38 (1999).

Yu, Y., Tan.S., Vanderwalle,J., Deprettere,E., "Near-optimal construction of wavelet networks for nonlinear system modelling," 1996 IEEE Int. Symposium on Circuits and Systems ISCAS'96, vol.3, Atlanta, 48-51 (1996a).

Yu, Y.; Tan, S.; Deprettere, E., "Stable construction of multi-scale fuzzy-wavelet system for image recovery and compression," Proc. SPIE **2846**, 354-65 (1996b).

Zhang Q. and Benveniste A., "Wavelet networks," IEEE Trans. On Neural Networks **3**, 889-898 (1992).

Zhang Xun, Shen Ronghui, Guo Guirong, "Automatic HRR target recognition based on Prony model wavelet and probability neural networks," Proc. CIE Int. Conf. on Radar, Publishing House of Electronics Industry, Beijing, 143-46 (1996).

Zhou, Q., Hong, G.S., Rahman, M., "A new tool life criterion for tool condition monitoring using a neural network," Engineering Applications of Artificial Intelligence **8**, 579-88 (1995).

This page is intentionally left blank

Index

This book presents the state of integration of wavelet theory and multiresolution analysis into soft computing. It is the first book on hybrid methods combining wavelet analysis with fuzzy logic, neural networks or genetic algorithms. Much attention is given to new approaches (fuzzy-wavelet) that permit one to develop, using wavelet techniques, linguistically interpretable fuzzy systems from data. The book also introduces the reader to wavelet-based genetic algorithms and multiresolution search. A special place is given to methods that have been implemented in real world applications, particularly the different techniques combining fuzzy logic or neural networks with wavelet theory.

ISBN 981-02-4609-9

www.worldscientific.com
4690 hc