

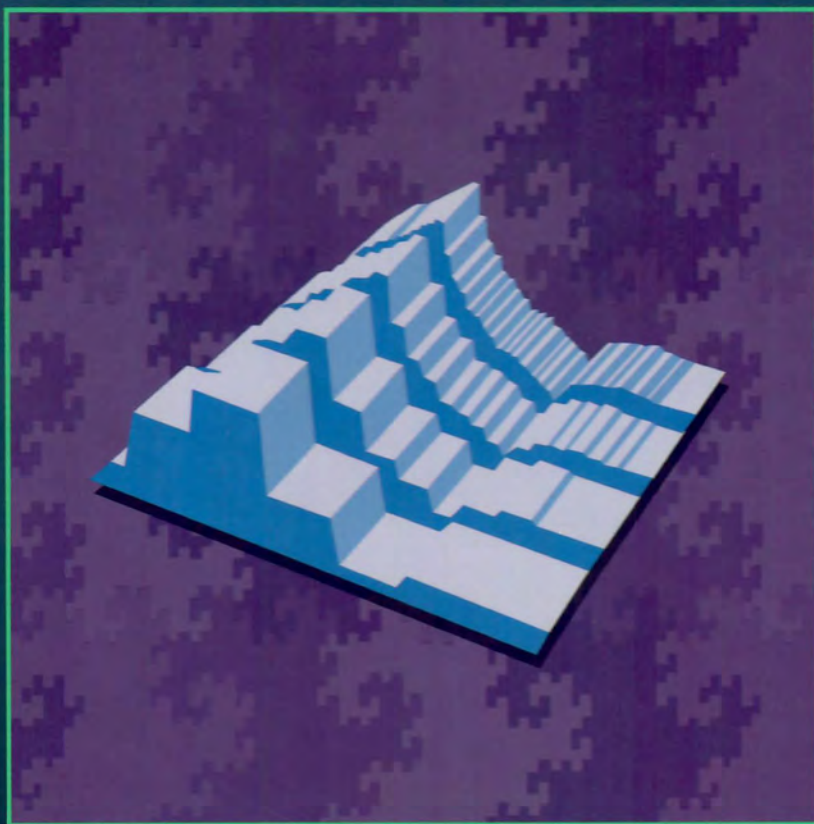
# WAVELETS

---

AND

---

# SUBBAND CODING



Martin Vetterli • Jelena Kovačević

---

Library of Congress Cataloging-in-Publication Data

Vetterli, Martin.

Wavelets and subband coding / Martin Vetterli, Jelena Kovačević.

p. cm.

ISBN 0-13-097080-8

1. Signal processing--Digital techniques. 2. Wavelets (Mathematics) I. Kovačević, Jelena. II. Title.

TK5102.9.V48 1995

621.382'2--dc20

95-5967

CIP

Editorial production: *bookworks*

Acquisitions editor: *Bernard Goodwin*

Manufacturing manager: *Alexis R. Heydt*

Cover designer: *Anthony Gemmellaro*

Cover illustration: *M. Vetterli*

About the cover: The three dimensional plot represents the first four iterations of a four-tap filter.

This filter converges to a scaling function used in the construction of a compactly supported Daubechies' wavelet. The background is a tiling of the plane with twin dragons. This is a two-dimensional generalization of the Haar scaling function.



© 1995 by Prentice Hall P T R

Prentice-Hall, Inc.

A Simon & Schuster Company

Englewood Cliffs, New Jersey 07632

The publisher offers discounts on this book when ordered in bulk quantities. For more information, contact: Corporate Sales Department, Prentice Hall P T R, 113 Sylvan Avenue, Englewood Cliffs, New Jersey 07632, Phone: (800) 382-3419, FAX: (201) 592-2249, E-mail: [dan\\_rush@prenhall.com](mailto:dan_rush@prenhall.com)

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3

ISBN 0-13-097080-8

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Simon & Schuster of Southeast Asia Pte. Ltd., *Singapore*

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

---

# Contents

|                                                                       |             |
|-----------------------------------------------------------------------|-------------|
| <b>Preface</b>                                                        | <b>xiii</b> |
| <b>1 Wavelets, Filter Banks and Multiresolution Signal Processing</b> | <b>1</b>    |
| 1.1 Series Expansions of Signals                                      | 3           |
| 1.2 Multiresolution Concept                                           | 9           |
| 1.3 Overview of the Book                                              | 10          |
| <b>2 Fundamentals of Signal Decompositions</b>                        | <b>14</b>   |
| 2.1 Notations                                                         | 15          |
| 2.2 Hilbert Spaces                                                    | 16          |
| 2.2.1 Vector Spaces and Inner Products                                | 17          |
| 2.2.2 Complete Inner Product Spaces                                   | 20          |
| 2.2.3 Orthonormal Bases                                               | 22          |
| 2.2.4 General Bases                                                   | 25          |
| 2.2.5 Overcomplete Expansions                                         | 26          |
| 2.3 Elements of Linear Algebra                                        | 28          |
| 2.3.1 Basic Definitions and Properties                                | 28          |
| 2.3.2 Linear Systems of Equations and Least Squares                   | 31          |
| 2.3.3 Eigenvectors and Eigenvalues                                    | 32          |
| 2.3.4 Unitary Matrices                                                | 33          |
| 2.3.5 Special Matrices                                                | 34          |
| 2.3.6 Polynomial Matrices                                             | 34          |

|          |                                                        |           |
|----------|--------------------------------------------------------|-----------|
| 2.4      | Fourier Theory and Sampling                            | 36        |
| 2.4.1    | Signal Expansions and Nomenclature                     | 36        |
| 2.4.2    | Fourier Transform                                      | 37        |
| 2.4.3    | Fourier Series                                         | 41        |
| 2.4.4    | Dirac Function, Impulse Trains and Poisson Sum Formula | 43        |
| 2.4.5    | Sampling                                               | 45        |
| 2.4.6    | Discrete-Time Fourier Transform                        | 48        |
| 2.4.7    | Discrete-Time Fourier Series                           | 50        |
| 2.4.8    | Discrete Fourier Transform                             | 51        |
| 2.4.9    | Summary of Various Flavors of Fourier Transforms       | 52        |
| 2.5      | Signal Processing                                      | 52        |
| 2.5.1    | Continuous-Time Signal Processing                      | 56        |
| 2.5.2    | Discrete-Time Signal Processing                        | 59        |
| 2.5.3    | Multirate Discrete-Time Signal Processing              | 65        |
| 2.6      | Time-Frequency Representations                         | 72        |
| 2.6.1    | Frequency, Scale and Resolution                        | 73        |
| 2.6.2    | Uncertainty Principle                                  | 75        |
| 2.6.3    | Short-Time Fourier Transform                           | 77        |
| 2.6.4    | Wavelet Transform                                      | 79        |
| 2.6.5    | Block Transforms                                       | 80        |
| 2.6.6    | Wigner-Ville Distribution                              | 80        |
| 2.A      | Bounded Linear Operators on Hilbert Spaces             | 81        |
| 2.B      | Parametrization of Unitary Matrices                    | 82        |
| 2.B.1    | Givens Rotations                                       | 83        |
| 2.B.2    | Householder Building Blocks                            | 84        |
| 2.C      | Convergence and Regularity of Functions                | 85        |
| 2.C.1    | Convergence                                            | 85        |
| 2.C.2    | Regularity                                             | 86        |
| <b>3</b> | <b>Discrete-Time Bases and Filter Banks</b>            | <b>92</b> |
| 3.1      | Series Expansions of Discrete-Time Signals             | 95        |
| 3.1.1    | Discrete-Time Fourier Series                           | 96        |
| 3.1.2    | Haar Expansion of Discrete-Time Signals                | 98        |
| 3.1.3    | Sinc Expansion of Discrete-Time Signals                | 103       |
| 3.1.4    | Discussion                                             | 106       |
| 3.2      | Two-Channel Filter Banks                               | 107       |
| 3.2.1    | Analysis of Filter Banks                               | 107       |
| 3.2.2    | Results on Filter Banks                                | 118       |
| 3.2.3    | Analysis and Design of Orthogonal FIR Filter Banks     | 122       |
| 3.2.4    | Linear Phase FIR Filter Banks                          | 133       |

|          |                                                              |            |
|----------|--------------------------------------------------------------|------------|
| 3.2.5    | Filter Banks with IIR Filters                                | 139        |
| 3.3      | Tree-Structured Filter Banks                                 | 142        |
| 3.3.1    | Octave-Band Filter Bank and Discrete-Time Wavelet Series     | 143        |
| 3.3.2    | Discrete-Time Wavelet Series and Its Properties              | 147        |
| 3.3.3    | Multiresolution Interpretation of Octave-Band Filter Banks   | 151        |
| 3.3.4    | General Tree-Structured Filter Banks and Wavelet Packets     | 153        |
| 3.4      | Multichannel Filter Banks                                    | 156        |
| 3.4.1    | Block and Lapped Orthogonal Transforms                       | 157        |
| 3.4.2    | Analysis of Multichannel Filter Banks                        | 159        |
| 3.4.3    | Modulated Filter Banks                                       | 165        |
| 3.5      | Pyramids and Overcomplete Expansions                         | 171        |
| 3.5.1    | Oversampled Filter Banks                                     | 171        |
| 3.5.2    | Pyramid Scheme                                               | 173        |
| 3.5.3    | Overlap-Save/Add Convolution and Filter Bank Implementations | 175        |
| 3.6      | Multidimensional Filter Banks                                | 176        |
| 3.6.1    | Analysis of Multidimensional Filter Banks                    | 177        |
| 3.6.2    | Synthesis of Multidimensional Filter Banks                   | 181        |
| 3.7      | Transmultiplexers and Adaptive Filtering in Subbands         | 185        |
| 3.7.1    | Synthesis of Signals and Transmultiplexers                   | 185        |
| 3.7.2    | Adaptive Filtering in Subbands                               | 187        |
| 3.A      | Lossless Systems                                             | 188        |
| 3.A.1    | Two-Channel Factorizations                                   | 189        |
| 3.A.2    | Multichannel Factorizations                                  | 190        |
| 3.B      | Sampling in Multiple Dimensions and Multirate Operations     | 194        |
| <b>4</b> | <b>Series Expansions Using Wavelets and Modulated Bases</b>  | <b>201</b> |
| 4.1      | Definition of the Problem                                    | 203        |
| 4.1.1    | Series Expansions of Continuous-Time Signals                 | 203        |
| 4.1.2    | Time and Frequency Resolution of Expansions                  | 206        |
| 4.1.3    | Haar Expansion                                               | 208        |
| 4.1.4    | Discussion                                                   | 213        |
| 4.2      | Multiresolution Concept and Analysis                         | 214        |
| 4.2.1    | Axiomatic Definition of Multiresolution Analysis             | 214        |
| 4.2.2    | Construction of the Wavelet                                  | 217        |
| 4.2.3    | Examples of Multiresolution Analyses                         | 220        |
| 4.3      | Construction of Wavelets Using Fourier Techniques            | 223        |
| 4.3.1    | Meyer's Wavelet                                              | 224        |
| 4.3.2    | Wavelet Bases for Piecewise Polynomial Spaces                | 228        |
| 4.4      | Wavelets Derived from Iterated Filter Banks and Regularity   | 237        |

|          |                                                                             |            |
|----------|-----------------------------------------------------------------------------|------------|
| 4.4.1    | Haar and Sinc Cases Revisited                                               | 237        |
| 4.4.2    | Iterated Filter Banks                                                       | 242        |
| 4.4.3    | Regularity                                                                  | 247        |
| 4.4.4    | Daubechies' Family of Regular Filters and Wavelets                          | 256        |
| 4.5      | Wavelet Series and Its Properties                                           | 259        |
| 4.5.1    | Definition and Properties                                                   | 260        |
| 4.5.2    | Properties of Basis Functions                                               | 265        |
| 4.5.3    | Computation of the Wavelet Series and Mallat's Algorithm                    | 269        |
| 4.6      | Generalizations in One Dimension                                            | 271        |
| 4.6.1    | Biorthogonal Wavelets                                                       | 271        |
| 4.6.2    | Recursive Filter Banks and Wavelets with Exponential Decay                  | 276        |
| 4.6.3    | Multichannel Filter Banks and Wavelet Packets                               | 278        |
| 4.7      | Multidimensional Wavelets                                                   | 281        |
| 4.7.1    | Multiresolution Analysis and Two-Scale Equation                             | 282        |
| 4.7.2    | Construction of Wavelets Using Iterated Filter Banks                        | 283        |
| 4.7.3    | Generalization of Haar Basis to Multiple Dimensions                         | 285        |
| 4.7.4    | Design of Multidimensional Wavelets                                         | 286        |
| 4.8      | Local Cosine Bases                                                          | 289        |
| 4.8.1    | Rectangular Window                                                          | 289        |
| 4.8.2    | Smooth Window                                                               | 291        |
| 4.8.3    | General Window                                                              | 292        |
| 4.A      | Proof of Theorem 4.5                                                        | 293        |
| <b>5</b> | <b>Continuous Wavelet and Short-Time Fourier Transforms and Frames</b>      | <b>299</b> |
| 5.1      | Continuous Wavelet Transform                                                | 301        |
| 5.1.1    | Analysis and Synthesis                                                      | 301        |
| 5.1.2    | Properties                                                                  | 304        |
| 5.1.3    | Morlet Wavelet                                                              | 311        |
| 5.2      | Continuous Short-Time Fourier Transform                                     | 312        |
| 5.2.1    | Properties                                                                  | 313        |
| 5.2.2    | Examples                                                                    | 314        |
| 5.3      | Frames of Wavelet and Short-Time Fourier Transforms                         | 315        |
| 5.3.1    | Discretization of Continuous-Time Wavelet and Short-Time Fourier Transforms | 316        |
| 5.3.2    | Reconstruction in Frames                                                    | 319        |
| 5.3.3    | Frames of Wavelets and STFT                                                 | 323        |
| 5.3.4    | Remarks                                                                     | 330        |

|                                                               |            |
|---------------------------------------------------------------|------------|
| <b>6 Algorithms and Complexity</b>                            | <b>333</b> |
| 6.1 Classic Results                                           | 334        |
| 6.1.1 Fast Convolution                                        | 334        |
| 6.1.2 Fast Fourier Transform Computation                      | 337        |
| 6.1.3 Complexity of Multirate Discrete-Time Signal Processing | 341        |
| 6.2 Complexity of Discrete Bases Computation                  | 345        |
| 6.2.1 Two-Channel Filter Banks                                | 345        |
| 6.2.2 Filter Bank Trees and Discrete-Time Wavelet Transforms  | 348        |
| 6.2.3 Parallel and Modulated Filter Banks                     | 352        |
| 6.2.4 Multidimensional Filter Banks                           | 353        |
| 6.3 Complexity of Wavelet Series Computation                  | 354        |
| 6.3.1 Expansion into Wavelet Bases                            | 354        |
| 6.3.2 Iterated Filters                                        | 355        |
| 6.4 Complexity of Overcomplete Expansions                     | 357        |
| 6.4.1 Short-Time Fourier Transform                            | 357        |
| 6.4.2 "Algorithme à Trous"                                    | 358        |
| 6.4.3 Multiple Voices Per Octave                              | 359        |
| 6.5 Special Topics                                            | 361        |
| 6.5.1 Computing Convolutions Using Multirate Filter Banks     | 361        |
| 6.5.2 Numerical Algorithms                                    | 365        |
| <b>7 Signal Compression and Subband Coding</b>                | <b>369</b> |
| 7.1 Compression Systems Based on Linear Transforms            | 371        |
| 7.1.1 Linear Transformations                                  | 372        |
| 7.1.2 Quantization                                            | 376        |
| 7.1.3 Entropy Coding                                          | 389        |
| 7.1.4 Discussion                                              | 392        |
| 7.2 Speech and Audio Compression                              | 393        |
| 7.2.1 Speech Compression                                      | 393        |
| 7.2.2 High-Quality Audio Compression                          | 394        |
| 7.2.3 Examples                                                | 397        |
| 7.3 Image Compression                                         | 399        |
| 7.3.1 Transform and Lapped Transform Coding of Images         | 400        |
| 7.3.2 Pyramid Coding of Images                                | 406        |
| 7.3.3 Subband and Wavelet Coding of Images                    | 410        |
| 7.3.4 Advanced Methods in Subband and Wavelet Compression     | 422        |
| 7.4 Video Compression                                         | 431        |
| 7.4.1 Key Problems in Video Compression                       | 431        |
| 7.4.2 Motion-Compensated Video Coding                         | 437        |
| 7.4.3 Pyramid Coding of Video                                 | 438        |

|                     |                                                                 |            |
|---------------------|-----------------------------------------------------------------|------------|
| 7.4.4               | Subband Decompositions for Video Representation and Compression | 440        |
| 7.4.5               | Example: MPEG Video Compression Standard                        | 447        |
| 7.5                 | Joint Source-Channel Coding                                     | 448        |
| 7.5.1               | Digital Broadcast                                               | 449        |
| 7.5.2               | Packet Video                                                    | 451        |
| 7.A                 | Statistical Signal Processing                                   | 451        |
| <b>Bibliography</b> |                                                                 | <b>461</b> |
| <b>Index</b>        |                                                                 | <b>481</b> |

---

## Preface

A central goal of signal processing is to describe real life signals, be it for computation, compression, or understanding. In that context, transforms or linear expansions have always played a key role. Linear expansions are present in Fourier's original work and in Haar's construction of the first wavelet, as well as in Gabor's work on time-frequency analysis. Today, transforms are central in fast algorithms such as the FFT as well as in applications such as image and video compression.

Over the years, depending on open problems or specific applications, theoreticians and practitioners have added more and more tools to the toolbox called signal processing. Two of the newest additions have been wavelets and their discrete-time cousins, filter banks or subband coding. From work in harmonic analysis and mathematical physics, and from applications such as speech/image compression and computer vision, various disciplines built up methods and tools with a similar flavor, which can now be cast into the common framework of wavelets.

This unified view, as well as the number of applications where this framework is useful, are motivations for writing this book. The unification has given a new understanding and a fresh view of some classic signal processing problems. Another motivation is that the subject is exciting and the results are cute!

The aim of the book is to present this unified view of wavelets and subband

coding. It will be done from a signal processing perspective, but with sufficient background material such that people without signal processing knowledge will find it useful as well. The level is that of a first year graduate engineering book (typically electrical engineering and computer sciences), but elementary Fourier analysis and some knowledge of linear systems in discrete time are enough to follow most of the book.

After the introduction (Chapter 1) and a review of the basics of vector spaces, linear algebra, Fourier theory and signal processing (Chapter 2), the book covers the five main topics in as many chapters. The discrete-time case, or filter banks, is thoroughly developed in Chapter 3. This is the basis for most applications, as well as for some of the wavelet constructions. The concept of wavelets is developed in Chapter 4, both with direct approaches and based on filter banks. This chapter describes wavelet series and their computation, as well as the construction of modified local Fourier transforms. Chapter 5 discusses continuous wavelet and local Fourier transforms, which are used in signal analysis, while Chapter 6 addresses efficient algorithms for filter banks and wavelet computations. Finally, Chapter 7 describes signal compression, where filter banks and wavelets play an important role. Speech/audio, image and video compression using transforms, quantization and entropy coding are discussed in detail. Throughout the book we give examples to illustrate the concepts, and more technical parts are left to appendices.

This book evolved from class notes used at Columbia University and the University of California at Berkeley. Parts of the manuscript have also been used at the University of Illinois at Urbana-Champaign and the University of Southern California. The material was covered in a semester, but it would also be easy to carve out a subset or skip some of the more mathematical subparts when developing a curriculum. For example, Chapters 3, 4 and 7 can form a good core for a course in Wavelets and Subband Coding. Homework problems are included in all chapters, complemented with project suggestions in Chapter 7. Since there is a detailed review chapter that makes the material as self-contained as possible, we think that the book is useful for self-study as well.

The subjects covered in this book have recently been the focus of books, special issues of journals, special conference proceedings, numerous articles and even new journals! To us, the book by I. Daubechies [73] has been invaluable, and Chapters 4 and 5 have been substantially influenced by it. Like the standard book by Meyer [194] and a recent book by Chui [49], it is a more mathematically oriented book than the present text. Another, more recent, tutorial book by Meyer gives an excellent overview of the history of the subject, its mathematical implications and current applications [195]. On the engineering side,

the book by Vaidyanathan [308] is an excellent reference on filter banks, as is Malvar's book [188] for lapped orthogonal transforms and compression. Several other texts, including edited books, have appeared on wavelets [27, 51, 251], as well as on subband coding [335] and multiresolution signal decompositions [3]. Recent tutorials on wavelets can be found in [128, 140, 247, 281], and on filter banks in [305, 307].

From the above, it is obvious that there is no lack of literature, yet we hope to provide a text with a broad coverage of theory and applications and a different perspective based on signal processing. We enjoyed preparing this material, and simply hope that the reader will find some pleasure in this exciting subject, and share some of our enthusiasm!

#### ACKNOWLEDGEMENTS

Some of the work described in this book resulted from research supported by the National Science Foundation, whose support is gratefully acknowledged. We would like also to thank Columbia University, in particular the Center for Telecommunications Research, the University of California at Berkeley and AT&T Bell Laboratories for providing support and a pleasant work environment. We take this opportunity to thank A. Oppenheim for his support and for including this book in his distinguished series. We thank K. Gettman and S. Papanikolaou of Prentice-Hall for their patience and help, and K. Fortgang of bookworks for her expert help in the production stage of the book.

To us, one of the attractions of the topic of *Wavelets and Subband Coding* is its interdisciplinary nature. This allowed us to interact with people from many different disciplines, and this was an enrichment in itself. The present book is the result of this interaction and the help of many people.

Our gratitude goes to I. Daubechies, whose work and help has been invaluable, to C. Herley, whose research, collaboration and help has directly influenced this book, and O. Rioul, who first taught us about wavelets and has always been helpful.

We would like to thank M.J.T. Smith and P.P. Vaidyanathan for a continuing and fruitful interaction on the topic of filter banks, and S. Mallat for his insights and interaction on the topic of wavelets.

Over the years, discussions and interactions with many experts have contributed to our understanding of the various fields relevant to this book, and we would like to acknowledge in particular the contributions of E. Adelson, T. Barnwell, P. Burt, A. Cohen, R. Coifman, R. Crochiere, P. Duhamel, C. Galland, W. Lawton, D. LeGall, Y. Meyer, T. Ramstad, G. Strang, M. Unser and V. Wickerhauser.

Many people have commented on several versions of the present text.

We thank I. Daubechies, P. Heller, M. Unser, P.P. Vaidyanathan, and G. Wornell for going through a complete draft and making many helpful suggestions. Comments on parts of the manuscript were provided by C. Chan, G. Chang, Z. Cvetković, V. Goyal, C. Herley, T. Kalker, M. Khansari, M. Kobayashi, H. Malvar, P. Moulin, A. Ortega, A. Park, J. Princen, K. Ramchandran, J. Shapiro and G. Strang, and are acknowledged with many thanks.

Coding experiments and associated figures were prepared by S. Levine (audio compression) and J. Smith (image compression), with guidance from A. Ortega and K. Ramchandran, and we thank them for their expert work. The images used in the experiments were made available by the Independent Broadcasting Association (UK).

The preparation of the manuscript relied on the help of many people. D. Heap is thanked for his invaluable contributions in the overall process, and in preparing the final version, and we thank C. Colbert, S. Elby, T. Judson, M. Karabatur, B. Lim, S. McCanne and T. Sharp for help at various stages of the manuscript.

The first author would like to acknowledge, with many thanks, the fruitful collaborations with current and former graduate students whose research has influenced this text, in particular Z. Cvetković, M. Garrett, C. Herley, J. Hong, G. Karlsson, E. Linzer, A. Ortega, H. Radha, K. Ramchandran, I. Shah, N.T. Thao and K.M. Uz. The early guidance by H.J. Nussbaumer, and the support of M. Kunt and G. Moschytz is gratefully acknowledged.

The second author would like to acknowledge friends and colleagues who contributed to the book, in particular C. Herley, G. Karlsson, A. Ortega and K. Ramchandran. Internal reviewers at Bell Labs are thanked for their efforts, in particular A. Reibman, G. Daryanani, P. Crouch, and T. Restaino.

## Wavelets, Filter Banks and Multiresolution Signal Processing

*"It is with logic that one proves;  
it is with intuition that one invents."*  
— Henri Poincaré

**T**he topic of this book is very old and very new. Fourier series, or expansion of periodic functions in terms of harmonic sines and cosines, date back to the early part of the 19th century when Fourier proposed harmonic trigonometric series [100]. The first wavelet (the only example for a long time!) was found by Haar early in this century [126]. But the construction of more general wavelets to form bases for square-integrable functions was investigated in the 1980's, along with efficient algorithms to compute the expansion. At the same time, applications of these techniques in signal processing have blossomed.

While linear expansions of functions are a classic subject, the recent constructions contain interesting new features. For example, wavelets allow good resolution in time and frequency, and should thus allow one to see "the forest and the trees." This feature is important for nonstationary signal analysis. While Fourier basis functions are given in closed form, many wavelets can only be obtained through a computational procedure (and even then, only at specific rational points). While this might seem to be a drawback, it turns out that if one is interested in implementing a signal expansion on real data, then a computational procedure is better than a closed-form expression!

The recent surge of interest in the types of expansions discussed here is due to the convergence of ideas from several different fields, and the recognition that techniques developed independently in these fields could be cast into a common framework.

The name “wavelet” had been used before in the literature,<sup>1</sup> but its current meaning is due to J. Goupillaud, J. Morlet and A. Grossman [119, 125]. In the context of geophysical signal processing they investigated an alternative to local Fourier analysis based on a single prototype function, and its scales and shifts. The modulation by complex exponentials in the Fourier transform is replaced by a scaling operation, and the notion of scale<sup>2</sup> replaces that of frequency. The simplicity and elegance of the wavelet scheme was appealing and mathematicians started studying wavelet analysis as an alternative to Fourier analysis. This led to the discovery of wavelets which form orthonormal bases for square-integrable and other function spaces by Meyer [194], Daubechies [71], Battle [21, 22], Lemarié [175], and others. A formalization of such constructions by Mallat [180] and Meyer [194] created a framework for wavelet expansions called multiresolution analysis, and established links with methods used in other fields. Also, the wavelet construction by Daubechies is closely connected to filter bank methods used in digital signal processing as we shall see.

Of course, these achievements were preceded by a long-term evolution from the 1910 Haar wavelet (which, of course, was not called a wavelet back then) to work using octave division of the Fourier spectrum (Littlewood-Paley) and results in harmonic analysis (Calderon-Zygmund operators). Other constructions were not recognized as leading to wavelets initially (for example, Stromberg’s work [283]).

Paralleling the advances in pure and applied mathematics were those in signal processing, but in the context of discrete-time signals. Driven by applications such as speech and image compression, a method called subband coding was proposed by Croisier, Esteban, and Galand [69] using a special class of filters called quadrature mirror filters (QMF) in the late 1970’s, and by Crochiere, Webber and Flanagan [68]. This led to the study of perfect reconstruction filter banks, a problem solved in the 1980’s by several people, including Smith and Barnwell [270, 271], Mintzer [196], Vetterli [315], and Vaidyanathan [306].

In a particular configuration, namely when the filter bank has octave bands, one obtains a discrete-time wavelet series. Such a configuration has been popular in signal processing less for its mathematical properties than because an octave band or logarithmic spectrum is more natural for certain appli-

<sup>1</sup>For example, for the impulse response of a layer in geophysical signal processing by Ricker [237] and for a causal finite-energy function by Robinson [248].

<sup>2</sup>For a beautiful illustration of the notion of scale, and an argument for geometric spacing of scale in natural imagery, see [197].

cations such as audio compression since it emulates the hearing process. Such an octave-band filter bank can be used, under certain conditions, to generate wavelet bases, as shown by Daubechies [71].

In computer vision, multiresolution techniques have been used for various problems, ranging from motion estimation to object recognition [249]. Images are successively approximated starting from a coarse version and going to a fine-resolution version. In particular, Burt and Adelson proposed such a scheme for image coding in the early 1980's [41], calling it pyramid coding.<sup>3</sup> This method turns out to be similar to subband coding. Moreover, the successive approximation view is similar to the multiresolution framework used in the analysis of wavelet schemes.

In computer graphics, a method called successive refinement iteratively interpolates curves or surfaces, and the study of such interpolators is related to wavelet constructions from filter banks [45, 92].

Finally, many computational procedures use the concept of successive approximation, sometimes alternating between fine and coarse resolutions. The multigrid methods used for the solution of partial differential equations [39] are an example.

While these interconnections are now clarified, this has not always been the case. In fact, maybe one of the biggest contributions of wavelets has been to bring people from different fields together, and from that cross fertilization and exchange of ideas and methods, progress has been achieved in various fields.

In what follows, we will take mostly a signal processing point of view of the subject. Also, most applications discussed later are from signal processing.

## 1.1 SERIES EXPANSIONS OF SIGNALS

We are considering linear expansions of signals or functions. That is, given any signal  $x$  from some space  $S$ , where  $S$  can be finite-dimensional (for example,  $\mathcal{R}^n$ ,  $\mathcal{C}^n$ ) or infinite-dimensional (for example,  $l_2(\mathcal{Z})$ ,  $L_2(\mathcal{R})$ ), we want to find a set of elementary signals  $\{\varphi_i\}_{i \in \mathcal{Z}}$  for that space so that we can write  $x$  as a linear combination

$$x = \sum_i \alpha_i \varphi_i. \quad (1.1.1)$$

The set  $\{\varphi_i\}$  is complete for the space  $S$ , if all signals  $x \in S$  can be expanded as in (1.1.1). In that case, there will also exist a dual set  $\{\tilde{\varphi}_i\}_{i \in \mathcal{Z}}$  such that the

<sup>3</sup>The importance of the pyramid algorithm was not immediately recognized. One of the reviewers of the original Burt and Adelson paper said, "I suspect that no one will ever use this algorithm again."

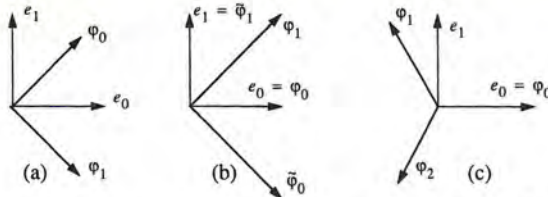


Figure 1.1 Examples of possible sets of vectors for the expansion of  $R^2$ . (a) Orthonormal case. (b) Biorthogonal case. (c) Overcomplete case.

expansion coefficients in (1.1.1) can be computed as

$$\alpha_i = \sum_n \tilde{\varphi}_i[n] x[n],$$

when  $x$  and  $\tilde{\varphi}_i$  are real discrete-time sequences, and

$$\alpha_i = \int \tilde{\varphi}_i(t) x(t) dt,$$

when they are real continuous-time functions. The above expressions are the *inner products* of the  $\tilde{\varphi}_i$ 's with the signal  $x$ , denoted by  $\langle \tilde{\varphi}_i, x \rangle$ . An important particular case is when the set  $\{\varphi_i\}$  is orthonormal and complete, since then we have an *orthonormal basis* for  $S$  and the basis and its dual are the same, that is,  $\varphi_i = \tilde{\varphi}_i$ . Then

$$\langle \varphi_i, \varphi_j \rangle = \delta[i - j],$$

where  $\delta[i]$  equals 1 if  $i = 0$ , and 0 otherwise. If the set is complete and the vectors  $\varphi_i$  are linearly independent but not orthonormal, then we have a *biorthogonal basis*, and the basis and its dual satisfy

$$\langle \varphi_i, \tilde{\varphi}_j \rangle = \delta[i - j].$$

If the set is complete but redundant (the  $\varphi_i$ 's are not linearly independent), then we do not have a basis but an overcomplete representation called a *frame*. To illustrate these concepts, consider the following example.

#### Example 1.1 Set of Vectors for the Plane

We show in Figure 1.1 some possible sets of vectors for the expansion of the plane, or  $\mathcal{R}^2$ . The standard Euclidean basis is given by  $e_0$  and  $e_1$ . In part (a), an orthonormal basis is given by  $\varphi_0 = [1, 1]^T / \sqrt{2}$  and  $\varphi_1 = [1, -1]^T / \sqrt{2}$ . The dual basis is identical, or  $\tilde{\varphi}_i = \varphi_i$ . In part (b), a biorthogonal basis is given, with  $\varphi_0 = e_0$  and  $\varphi_1 = [1, 1]^T$ . The dual basis is now  $\tilde{\varphi}_0 = [1, -1]^T$  and  $\tilde{\varphi}_1 = [0, 1]^T$ . Finally, in part (c), an overcomplete set is given, namely  $\varphi_0 = [1, 0]^T$ ,  $\varphi_1 = [-1/2, \sqrt{3}/2]^T$  and  $\varphi_2 = [-1/2, -\sqrt{3}/2]^T$ . Then, it can be verified that a possible reconstruction basis is identical (up to a scale factor), namely,  $\tilde{\varphi}_i = 2/3 \varphi_i$  (the reconstruction basis is not unique). This set behaves as an orthonormal basis, even though the vectors are linearly dependent.

The representation in (1.1.1) is a change of basis, or, conceptually, a change of point of view. The obvious question is, what is a good basis  $\{\varphi_i\}$  for  $S$ ? The answer depends on the class of signals we want to represent, and on the choice of a criterion for quality. However, in general, a good basis is one that allows compact representation or less complex processing. For example, the Karhunen-Loève transform concentrates as much energy in as few coefficients as possible, and is thus good for compression, while, for the implementation of convolution, the Fourier basis is computationally more efficient than the standard basis.

We will be interested mostly in expansions with some structure, that is, expansions where the various basis vectors are related to each other by some elementary operations such as shifting in time, scaling, and modulation (which is shifting in frequency). Because we are concerned with expansions for very high-dimensional spaces (possibly infinite), bases without such structure are useless for complexity reasons.

Historically, the Fourier series for periodic signals is the first example of a signal expansion. The basis functions are harmonic sines and cosines. Is this a good set of basis functions for signal processing? Besides its obvious limitation to periodic signals, it has very useful properties, such as the convolution property which comes from the fact that the basis functions are eigenfunctions of linear time-invariant systems. The extension of the scheme to nonperiodic signals,<sup>4</sup> by segmentation and piecewise Fourier series expansion of each segment, suffers from artificial boundary effects and poor convergence at these boundaries (due to the Gibbs phenomenon).

An attempt to create local Fourier bases is the Gabor transform or short-time Fourier transform (STFT). A smooth window is applied to the signal centered around  $t = nT_0$  (where  $T_0$  is some basic time step), and a Fourier expansion is applied to the windowed signal. This leads to a time-frequency representation since we get an approximate information about the frequency content of the signal around the location  $nT_0$ . Usually, frequency points spaced  $2\pi/T_0$  apart are used and we get a sampling of the time-frequency plane on a rectangular grid. The spectrogram is related to such a time-frequency analysis. Note that the functions used in the expansion are related to each other by shift in time and modulation, and that we obtain a linear frequency analysis. While the STFT has proven useful in signal analysis, there are no good orthonormal bases based on this construction. Also, a logarithmic frequency scale, or constant relative bandwidth, is often preferable to the linear frequency scale obtained with the

<sup>4</sup>The Fourier transform of nonperiodic signals is also possible. It is an integral transform rather than a series expansion and lacks any time locality.

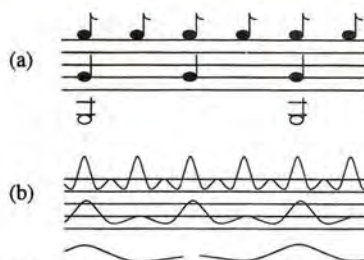


Figure 1.2 Musical notation and orthonormal wavelet bases. (a) The western musical notation uses a logarithmic frequency scale with twelve halftones per octave. In this example, notes are chosen as in an orthonormal wavelet basis, with long low-pitched notes, and short high-pitched ones. (b) Corresponding time-domain functions.

STFT. For example, the human auditory system uses constant relative bandwidth channels (critical bands), and therefore, audio compression systems use a similar decomposition.

A popular alternative to the STFT is the wavelet transform. Using scales and shifts of a prototype wavelet, a linear expansion of a signal is obtained. Because the scales used are powers of an elementary scale factor (typically 2), the analysis uses a constant relative bandwidth (or, the frequency axis is logarithmic). The sampling of the time-frequency plane is now very different from the rectangular grid used in the STFT. Lower frequencies, where the bandwidth is narrow (that is, the basis functions are stretched in time) are sampled with a large time step, while high frequencies (which correspond to short basis functions) are sampled more often. In Figure 1.2, we give an intuitive illustration of this time-frequency trade-off, and relate it to musical notation which also uses a logarithmic frequency scale.<sup>5</sup> What is particularly interesting is that such a wavelet scheme allows good orthonormal bases whereas the STFT does not.

In the discussions above, we implicitly assumed continuous-time signals. Of course there are discrete-time equivalents to all these results. A local analysis can be achieved using a block transform, where the sequence is segmented into adjacent blocks of  $N$  samples, and each block is individually transformed. As is to be expected, such a scheme is plagued by boundary effects, also called blocking effects. A more general expansion relies on filter banks, and can achieve both STFT-like analysis (rectangular sampling of the time-frequency plane) or wavelet-like analysis (constant relative bandwidth in frequency). Discrete-time expansions based on filter banks are not arbitrary, rather they

<sup>5</sup>This is the standard western musical notation based on J.S. Bach's "Well Tempered Piano". Thus one could argue that wavelets were actually invented by J.S. Bach!

are structured expansions. Again, for complexity reasons, it is useful to impose such a structure on the basis chosen for the expansion. For example, filter banks correspond to basis sequences which satisfy a block shift invariance property. Sometimes, a modulation constraint can also be added, in particular in STFT-like discrete-time bases. Because we are in discrete time, scaling cannot be done exactly (unlike in continuous time), but an approximate scaling property between basis functions holds for the discrete-time wavelet series.

Interestingly, the relationship between continuous- and discrete-time bases runs deeper than just these conceptual similarities. One of the most interesting constructions of wavelets is the one by Daubechies [71]. It relies on the iteration of a discrete-time filter bank so that, under certain conditions, it converges to a continuous-time wavelet basis. Furthermore, the multiresolution framework used in the analysis of wavelet decompositions automatically associates a discrete-time perfect reconstruction filter bank to any wavelet decomposition. Finally, the wavelet series decomposition can be computed with a filter bank algorithm. Therefore, especially in the wavelet type of a signal expansion, there is a very close interaction between discrete and continuous time.

It is to be noted that we have focused on STFT and wavelet type of expansions mainly because they are now quite standard. However, there are many alternatives, for example the wavelet packet expansion introduced by Coifman and coworkers [62, 64], and generalizations thereof. The main ingredients remain the same: they are structured bases in discrete or continuous time, and they permit different time versus frequency resolution trade-offs. An easy way to interpret such expansions is in terms of their time-frequency tiling: each basis function has a region in the time-frequency plane where most of its energy is concentrated. Then, given a basis and the expansion coefficients of a signal, one can draw a tiling where the shading corresponds to the value of the expansion coefficient.<sup>6</sup>

#### Example 1.2 *Different Time-Frequency Tilings*

Figure 1.3 shows schematically different possible expansions of a very simple discrete-time signal, namely a sine wave plus an impulse (see part (a)). It would be desirable to have an expansion that captures both the isolated impulse (or Dirac in time) and the isolated frequency component (or Dirac in frequency). The first two expansions, namely the identity transform in part (b) and the discrete-time Fourier series<sup>7</sup> in part (c), isolate the time and frequency impulse, respectively, but not both. The local discrete-time Fourier series in part (d) achieves a compromise, by locating both impulses to a certain degree. The discrete-time

<sup>6</sup>Such tiling diagrams were used by Gabor [102], and he called an elementary tile a “logon.”

<sup>7</sup>Discrete-time series expansions are often called discrete-time transforms, both in the Fourier and in the wavelet case.

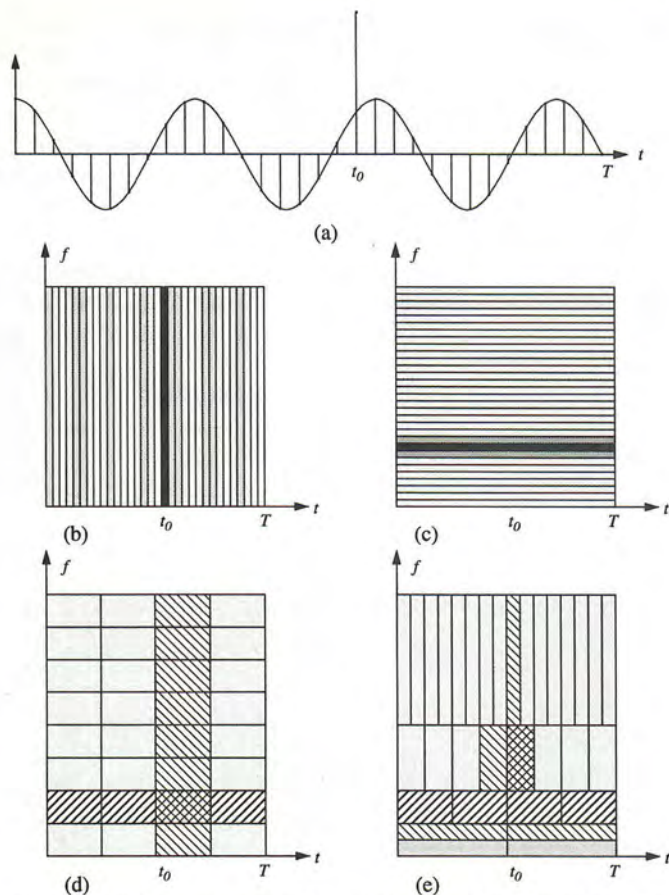


Figure 1.3 Time-frequency tilings for a simple discrete-time signal [130]. (a) Sine wave plus impulse. (b) Expansion onto the identity basis. (c) Discrete-time Fourier series. (d) Local discrete-time Fourier series. (e) Discrete-time wavelet series.

wavelet series in part (e) achieves better localization of the time-domain impulse, without sacrificing too much of the frequency localization. However, a high-frequency sinusoid would not be well localized. This simple example indicates some of the trade-offs involved.

Note that the local Fourier transform and the wavelet transform can be used for signal analysis purposes. In that case, the goal is not to obtain orthonormal bases, but rather to characterize the signal from the transform. The

local Fourier transform retains many of the characteristics of the usual Fourier transform with a localization given by the window function, which is thus constant at all frequencies (this phenomenon can be seen already in Figure 1.3(d)). The wavelet, on the other hand, acts as a microscope, focusing on smaller time phenomena as the scale becomes small (see Figure 1.3(e) to see how the impulse gets better localized at high frequencies). This behavior permits a local characterization of functions, which the Fourier transform does not.<sup>8</sup>

## 1.2 MULTIREOLUTION CONCEPT

A slightly different expansion is obtained with multiresolution pyramids since the expansion is actually redundant (the number of samples in the expansion is bigger than in the original signal). However, conceptually, it is intimately related to subband and wavelet decompositions. The basic idea is successive approximation. A signal is written as a coarse approximation (typically a lowpass, subsampled version) plus a prediction error which is the difference between the original signal and a prediction based on the coarse version. Reconstruction is immediate: simply add back the prediction to the prediction error. The scheme can be iterated on the coarse version. It can be shown that if the lowpass filter meets certain constraints of orthogonality, then this scheme is identical to an oversampled discrete-time wavelet series. Otherwise, the successive approximation approach is still at least conceptually identical to the wavelet decomposition since it performs a multiresolution analysis of the signal.

A schematic diagram of a pyramid decomposition, with attached resulting images, is shown in Figure 1.4. After the encoding, we have a coarse resolution image of half size, as well as an error image of full size (thus the redundancy). For applications, the decomposition into a coarse resolution which gives an approximate but adequate version of the full image, plus a difference or detail image, is conceptually very important.

### Example 1.3 *Multiresolution Image Database*

Let us consider the following practical problem: Users want to access and retrieve electronic images from an image database using a computer network with limited bandwidth. Because the users have an approximate idea of which image they want, they will first browse through some images before settling on a target image [214]. Given the limited bandwidth, browsing is best done on coarse versions of the images which can be transmitted faster. Once an image is chosen, the residual can be sent. Thus, the scheme shown in Figure 1.4 can be used, where the coarse and residual images are further compressed to diminish the transmission time.

<sup>8</sup>For example, in [137], this mathematical microscope is used to analyze some famous lacunary Fourier series that was proposed over a century ago.

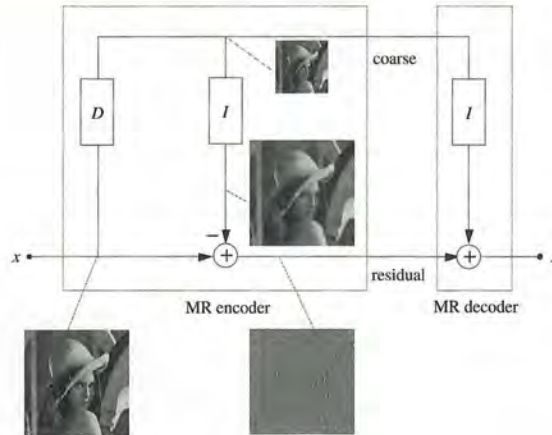


Figure 1.4 Pyramid decomposition of an image where encoding is shown on the left and decoding is shown on the right. The operators  $D$  and  $I$  correspond to decimation and interpolation operators, respectively. For example,  $D$  produces an  $N/2 \times N/2$  image from an  $N \times N$  original, while  $I$  interpolates an  $N \times N$  image based on an  $N/2 \times N/2$  original.

The above example is just one among many schemes where multiresolution decompositions are useful in communications problems. Others include transmission over error-prone channels, where the coarse resolution can be better protected to guarantee some minimum level of quality.

Multiresolution decompositions are also important for computer vision tasks such as image segmentation or object recognition: the task is performed in a successive approximation manner, starting on the coarse version and then using this result as an initial guess for the full task. However, this is a greedy approach which is sometimes suboptimal. Figure 1.5 shows a famous counterexample, where a multiresolution approach would be seriously misleading ...

Interestingly, the multiresolution concept, besides being intuitive and useful in practice, forms the basis of a mathematical framework for wavelets [181, 194]. As in the pyramid example shown in Figure 1.4, one can decompose a function into a coarse version plus a residual, and then iterate this to infinity. If properly done, this can be used to analyze wavelet schemes and derive wavelet bases.

### 1.3 OVERVIEW OF THE BOOK

We start with a review of fundamentals in Chapter 2. This chapter should make the book as self-contained as possible. It reviews Hilbert spaces at an el-



Figure 1.5 Counter-example to multiresolution technique. The coarse approximation is unrelated to the full-resolution image (Comet Photo AG).

elementary but sufficient level, linear algebra (including matrix polynomials) and Fourier theory, with material on sampling and discrete-time Fourier transforms in particular. The review of continuous-time and discrete-time signal processing is followed by a discussion of multirate signal processing, which is a topic central to later chapters. Finally, a short introduction to time-frequency distributions discusses the local Fourier transform and the wavelet transform, and shows the uncertainty principle. The appendix gives factorizations of unitary matrices, and reviews results on convergence and regularity of functions.

Chapter 3 focuses on discrete-time bases and filter banks. This topic is important for several later chapters as well as for applications. We start with two simple expansions which will reappear throughout the book as a recur-

ring theme: the Haar and the sinc bases. They are limit cases of orthonormal expansions with good time localization (Haar) and good frequency localization (sinc). This naturally leads to an in-depth study of two-channel filter banks, including analytical tools for their analysis as well as design methods. The construction of orthonormal and linear phase filter banks is described. Multichannel filter banks are developed next, first through tree structures and then in the general case. Modulated filter banks, corresponding conceptually to a discrete-time local Fourier analysis, are addressed as well. Next, pyramid schemes and overcomplete representations are explored. Such schemes, while not critically sampled, have some other attractive features, such as time invariance. Then, the multidimensional case is discussed both for simple separable systems, as well as for general nonseparable ones. The latter systems involve lattice sampling which is detailed in an appendix. Finally, filter banks for telecommunications, namely transmultiplexers and adaptive subband filtering, are presented briefly. The appendix details factorizations of orthonormal filter banks (corresponding to paraunitary matrices).

Chapter 4 is devoted to the construction of bases for continuous-time signals, in particular wavelets and local cosine bases. Again, the Haar and sinc cases play illustrative roles as extremes of wavelet constructions. After an introduction to series expansions, we develop multiresolution analysis as a framework for wavelet constructions. This naturally leads to the classic wavelets of Meyer and Battle-Lemarié or Stromberg. These are based on Fourier-domain analysis. This is followed by Daubechies' construction of wavelets from iterated filter banks. This is a time-domain construction based on the iteration of a multirate filter. Study of the iteration leads to the notion of regularity of the discrete-time filter. Then, the wavelet series expansion is considered both in terms of properties and computation of the expansion coefficients. Some generalizations of wavelet constructions are considered next, first in one dimension (including biorthogonal and multichannel wavelets) and then in multiple dimensions, where nonseparable wavelets are shown. Finally, local cosine bases are derived and they can be seen as a real-valued local Fourier transform.

Chapter 5 is concerned with continuous wavelet and Fourier transforms. Unlike the series expansions in Chapters 3 and 4, these are very redundant representations useful for signal analysis. Both transforms are analyzed, inverses are derived, and their main properties are given. These transforms can be sampled, that is, scale/frequency and time shift can be discretized. This leads to redundant series representations called frames. In particular, reconstruction or inversion is discussed, and the case of wavelet and local Fourier frames is considered in some detail.

Chapter 6 treats algorithmic and computational aspects of series expansion

sions. First, a review of classic fast algorithms for signal processing is given since they form the ingredients used in subsequent algorithms. The key role of the fast Fourier transform (FFT) is pointed out. The complexity of computing filter banks, that is, discrete-time expansions, is studied in detail. Important cases include the discrete-time wavelet series or transform and modulated filter banks. The latter corresponds to a local discrete-time Fourier series or transform, and uses FFT's for efficient computation. These filter bank algorithms have direct applications in the computation of wavelet series. Overcomplete expansions are considered next, in particular for the computation of a sampled continuous wavelet transform. The chapter concludes with a discussion of special topics related to efficient convolution algorithms and also application of wavelet ideas to numerical algorithms.

The last chapter is devoted to one of the main applications of wavelets and filter banks in signal processing, namely signal compression. The technique is often called subband coding because signals are considered in spectral bands for compression purposes. First comes a review of transform based compression, including quantization and entropy coding. Then follow specific discussions of one-, two- and three-dimensional signal compression methods based on transforms. Speech and audio compression, where subband coding was first invented, is discussed. The success of subband coding in current audio coding algorithms is shown on specific examples such as the MUSICAM standard. A thorough discussion of image compression follows. While current standards such as JPEG are block transform based, some innovative subband or wavelet schemes are very promising and are described in detail. Video compression is considered next. Besides expansions, motion estimation/compensation methods play a key role and are discussed. The multiresolution feature inherent in pyramid and subband coding is pointed out as an attractive feature for video compression, just as it is for image coding. The final section discusses the interaction of source coding, particularly the multiresolution type, and channel coding or transmission. This joint source-channel coding is key to new applications of image and video compression, as in transmission over packet networks. An appendix gives a brief review of statistical signal processing which underlies coding methods.

## Signal Compression and Subband Coding

*"That which shrinks must first expand."*  
— Lao-Tzu, *Tao Te Ching*

**T**he compression of signals, which is one of the main applications of digital signal processing, uses signal expansions as a major component. Some of these expansions were discussed in previous chapters, most notably discrete-time expansions via filter banks. When the channels of a filter bank are used for coding, the resulting scheme is known as *subband coding*. The reasons for expanding a signal and processing it in transform domain are numerous. While source coding can be performed on the original signal directly, it is usually more efficient to find an appropriate transform. By efficient we mean that for a given complexity of the encoder, better compression is achieved.

The first useful property of transforms, or "generalized" transforms such as subband coding, is their decorrelation property. That is, in the transform domain, the transform coefficients are not correlated, which is equivalent to diagonalizing the autocovariance matrix of the signal, as will be seen in Section 7.1. This diagonalization property is similar to the convolution property (or the diagonalization of circulant matrices) of the Fourier transform as we discussed in Section 2.4.8. However, the only transform that achieves exact diagonalization, the Karhunen-Loève transform, is usually impractical. Many other transforms come close to exact diagonalization and are therefore popular, such as the discrete cosine transform, or, appropriately designed subband or wavelet transforms. The second advantage of transforms is that the new domain is often more appropriate for quantization using perceptual criterions. That is, the transform domain can be used to distribute errors in a way that is less

objectionable for the human user. For example, in speech and audio coding, the frequency bands used in subband coding might mimic operations performed in the inner ear and thus one can exploit the reduced sensitivity or even masking between bands. The third advantage of transform coding is that the previous features come at a low computational price. The transform decomposition itself is computed using fast algorithms as discussed in Chapter 6, quantization in the transform domain is often simple scalar quantization, and entropy coding is done on a sample-by-sample basis.

Together, these advantages produced successful compression schemes for speech, audio, images and video, some of which are now industry standards (32 Kbits/sec subband coding for high-quality speech [192], AC [34, 290], PAC [147], and MUSICAM for audio [77, 279], JPEG for images [148, 327], MPEG for video [173, 201]).

It is important to note that the signal expansions on which we have focused so far are only one of the three major components of such compression schemes. The other two are quantization and entropy coding. This three part view of compression will be developed in detail in Section 7.1, together with the strong interaction that exists among them. That is, in a compression context, there is no need for designing the “ultimate” basis function system unless adequate quantization and entropy coding are matched to it. This interplay, while fairly obvious, is often insufficiently stressed in the literature. Note that this section is a review and can be skipped by readers familiar with basic signal compression.

Section 7.2 concentrates on one-dimensional signal compression, that is, speech and audio coding. Subband methods originated from speech compression research, and for good reasons: Dividing the signal in frequency bands imitates the human auditory system well enough to be the basis for a series of successful coders.

Section 7.3 discusses image compression, where transform and subband/wavelet methods hold a preeminent position. It turns out that representing images at multiple resolutions is a desirable feature in many systems using image compression such as image databases, and thus, subband or wavelet methods are a popular choice. We also discuss some new schemes which contain wavelet decompositions as a key ingredient.

Section 7.4 adds one more dimension and discusses video compression. While straight linear transforms have been used, they are outperformed by methods using a combination of motion based modeling and transforms. Again, a multiresolution feature is often desired and will be discussed.

Section 7.5 discusses joint source-channel coding using multiresolution source decompositions and matched channel coding. It turns out that several upcoming applications, such as digital broadcasting and transmission over

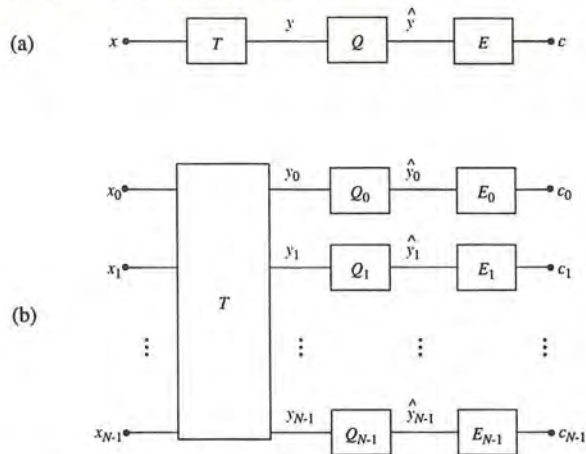


Figure 7.1 Compression system based on linear transformation. The linear transform  $T$  is followed by quantization ( $Q$ ) and entropy coding ( $E$ ). The reconstruction is simply  $\hat{x} = T^{-1}\hat{y}$ . (a) Global view. (b) Multichannel case with scalar quantization and entropy coding.

highly varying channels such as wireless channels or channels corresponding to packet-switched transmission, are improved by using multiresolution techniques.

## 7.1 COMPRESSION SYSTEMS BASED ON LINEAR TRANSFORMS

In this section, we will deal with compression systems, as given in Figure 7.1(a). The linear transformation ( $T$ ) is the first step in the process which includes quantization ( $Q$ ) and entropy coding ( $E$ ). Quantization introduces nonlinearities in the system and results in loss of information, while entropy coding is a reversible process. A system as given in Figure 7.1 is termed an open-loop system, since there is no feedback from the output to the input. On the other hand, a closed-loop system, such as the DPCM (see Figure 7.5), includes the quantization in the loop. We mostly concentrate on open-loop systems, because of their close connection to signal expansions. Following Figure 7.1, we start by discussing various linear transforms with an emphasis on the optimal Karhunen-Loève transform, followed by quantization, and end up briefly describing entropy coding methods. We try to emphasize the interplay among these three parts, as well as indicate the importance of perceptual criteria in designing the overall system. Our discussion is based on the excellent text by Gersho and Gray [109], to which we refer for more details. This chapter uses results from statistical signal processing, which are reviewed in Appendix 7.A.

Let us here define the measures of quality we will be using. First, the *mean square error* (MSE), or, *distortion*, equals

$$D = E\left(\sum_{i=0}^{N-1} |x_i - \hat{x}_i|^2\right), \quad (7.1.1)$$

where  $x_i$  are the input values and  $\hat{x}_i$  are the reconstructed values. For a zero-mean input, the signal-to-noise ratio (SNR) is given by

$$SNR = 10 \log_{10} \frac{\sigma^2}{D}, \quad (7.1.2)$$

where  $D$  is as given in (7.1.1) and  $\sigma^2$  is the input variance. The peak signal-to-noise ratio ( $SNR_p$ ) is defined as [138]

$$SNR_p = 10 \log_{10} \frac{M^2}{D}, \quad (7.1.3)$$

where  $M$  is the maximum peak-to-peak value in the signal (typically 256 for 8-bit images). Distortion measures based on squared error have shortcomings when assessing the quality of a coded signal such as an image. An improved distortion measure is a perceptually weighted mean square error. Even better are distortion models which include masking. These distortion metrics are signal specific, and some of them will be discussed in conjunction with practical compression schemes in later sections.

### 7.1.1 Linear Transformations

Assume a vector  $x[n] = (x[n], x[n+1], \dots, x[n+N-1])^T$  of  $N$  consecutive samples of a real wide-sense stationary random process (see Appendix 7.A). Typically, these samples are correlated and independent coding of the samples is inefficient. The idea is to apply a linear transform<sup>1</sup> so that the transform coefficients are decorrelated. While there is no general formal result that guarantees more efficient compression by decorrelation, it turns out in practice (and for certain cases in theory) that scalar quantization of decorrelated transform coefficients is more efficient than direct scalar quantization of the samples.

Since we assumed that the process is wide-sense stationary and we will be dealing only with the second-order statistics, we do not need to keep the index  $n$  for  $x[n]$  and can abbreviate it simply as  $x$ . From now on, we will assume that the process is zero-mean and thus its autocorrelation and autocovariance are

<sup>1</sup>This can also be seen as a discrete-time series expansion. However, since it is usually implemented as a matrix block transform we will adhere to the compression literature's convention and call it a transform.

the same, that is,  $K[n, m] = R[n, m]$ . The autocovariance matrix of the input vector  $\mathbf{x}$  is

$$\mathbf{K}_x = E(\mathbf{x} \cdot \mathbf{x}^T).$$

Again, since the process is wide-sense stationary and zero-mean,  $K[n, m] = K[n - m] = R[n - m]$  (see Appendix 7.A). Therefore, the matrix  $\mathbf{K}_x$  has the following form:

$$\mathbf{K}_x = \begin{pmatrix} R[0] & R[1] & \dots & R[N-1] \\ R[1] & R[0] & \dots & R[N-2] \\ \vdots & \vdots & \ddots & \vdots \\ R[N-1] & R[N-2] & \dots & R[0] \end{pmatrix}.$$

This matrix is Toeplitz, symmetric (see Section 2.3.5), and nonnegative definite since all of its eigenvalues are greater or equal to zero (this holds in general for autocorrelation matrices). Consider now the transformed vector  $\mathbf{y}$ ,

$$\mathbf{y} = \mathbf{T}\mathbf{x}, \quad (7.1.4)$$

where  $\mathbf{T}$  is an  $N \times N$  unitary matrix which thus satisfies  $\mathbf{T}^T \mathbf{T} = \mathbf{T} \mathbf{T}^T = \mathbf{I}$ . Then the autocovariance of  $\mathbf{y}$  is

$$\begin{aligned} \mathbf{K}_y &= E(\mathbf{y}\mathbf{y}^T) = E(\mathbf{T}\mathbf{x}\mathbf{x}^T\mathbf{T}^T) = \mathbf{T}E(\mathbf{x}\mathbf{x}^T)\mathbf{T}^T \\ &= \mathbf{T}\mathbf{K}_x\mathbf{T}^T. \end{aligned} \quad (7.1.5)$$

**Karhunen-Loève Transform** We would like to obtain uncorrelated transform coefficients. Recall that for each two coefficients to be uncorrelated, their covariance has to be zero (see Appendix 7.A). Thus, we are looking for a diagonal  $\mathbf{K}_y$ . For that to hold,  $\mathbf{T}$  has to be chosen with its rows equal to the eigenvectors of  $\mathbf{K}_x$ . Call  $\mathbf{v}_i$  the eigenvector (normalized to unit norm) of  $\mathbf{K}_x$  associated with the eigenvalue  $\lambda_i$ , that is,  $\mathbf{K}_x \mathbf{v}_i = \lambda_i \mathbf{v}_i$ , and choose the following ordering for the  $\lambda_i$ 's:

$$\lambda_0 \geq \lambda_1 \geq \dots \geq \lambda_{N-1} \geq 0, \quad (7.1.6)$$

where the last inequality holds because  $\mathbf{K}_x$  is nonnegative definite. Moreover, since  $\mathbf{K}_x$  is symmetric, there is a complete set of orthonormal eigenvectors (see Section 2.3.2). Take  $\mathbf{T}$  as

$$\mathbf{T} = [\mathbf{v}_0 \ \mathbf{v}_1 \ \dots \ \mathbf{v}_{N-1}]^T, \quad (7.1.7)$$

then, from (7.1.5),

$$\mathbf{K}_y = \mathbf{T} \cdot \mathbf{K}_x \cdot \mathbf{T}^T = \mathbf{T} \cdot \mathbf{T}^T \cdot \mathbf{\Lambda} = \mathbf{\Lambda}, \quad (7.1.8)$$

where  $\mathbf{\Lambda}$  is a diagonal matrix with  $\Lambda_{ii} = \lambda_i = \sigma_i^2 = y_i^2$ ,  $i = 0, \dots, N-1$ . The transform defined in (7.1.7) which achieves decorrelation as shown in (7.1.8) is the *discrete-time Karhunen-Loève (KLT)* or *Hotelling transform* [109, 138]. The following approximation result is intuitive:

## PROPOSITION 7.1

If only  $k$  out of the  $N$  transform coefficients are kept, then the coefficients  $y_0, \dots, y_{k-1}$  will minimize the MSE between  $x$  and its approximation  $\hat{x}$ .

Although the proof of this result follows from general orthonormal expansions results given in Chapter 2, we describe it here for completeness.

## PROOF

Following (7.1.1), the MSE is equal to

$$D = E \left( \sum_{i=0}^{N-1} (x_i - \hat{x}_i)^2 \right) = E((x - \hat{x})^T \cdot (x - \hat{x})) = E((y - \hat{y})^T \cdot (y - \hat{y})), \quad (7.1.9)$$

where the last equality follows from the fact that  $T$  is a unitary transform, that is, the MSE is conserved between transform and original domains. Keeping only the first  $k$  coefficients means that  $\hat{y}_i = y_i$  for  $i = 0, \dots, k-1$  and  $\hat{y}_i = 0$ , for  $i = k, \dots, N-1$ . Then the MSE equals

$$D_k = E \left( \sum_{i=0}^{N-1} (y_i - \hat{y}_i)^2 \right) = \frac{1}{N} \sum_{i=k}^{N-1} y_i^2 = \frac{1}{N} \sum_{i=k}^{N-1} \lambda_i,$$

and this is obviously smaller or equal to any other set of  $N - k$  coefficients because of the ordering in (7.1.6). Recall here that the assumption of zero mean still holds.

Another way to say this is that the first  $k$  coefficients contain most of the energy of the transformed signal. This is the “energy packing” property of the Karhunen-Loève transform. Actually, among all unitary transforms, the KLT is the one that packs most energy into the first  $k$  coefficients.

There are two major problems with the KLT, however. First, the KLT is signal dependent, since it depends on the autocovariance matrix. Second, it is computationally complex, since no structure can be assumed for  $T$ , and no fast algorithm can be used. This leads to an order  $N^2$  operations for applying the transform.

**Discrete Cosine Transform** Due to the discussed problems, various approximations to the KLT have been proposed. These approximations usually have fast algorithms for efficient implementation. The most successful is the *discrete cosine transform* (DCT), which calculates the vector  $y$  from  $x$  as

$$y_0 = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n, \quad (7.1.10)$$

$$y_k = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x_n \cos \left( \frac{2\pi(2n+1)k}{4N} \right), \quad k = 1, \dots, N-1. \quad (7.1.11)$$

The DCT was developed [2] as an approximation for the KLT of a first-order Gauss-Markov process with a large positive correlation coefficient  $\rho$  ( $\rho \rightarrow 1$ ). In this case,  $K_x$  is of the following form (assuming unit variance and zero mean)

$$K_x = \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 & \cdots \\ \rho & 1 & \rho & \rho^2 & \cdots \\ \rho^2 & \rho & 1 & \rho & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

For large  $\rho$ 's, the DCT approximately diagonalizes  $K_x$ . Actually, the DCT (as well as some other transforms) is asymptotically equivalent to the KLT of an arbitrary wide-sense stationary process when the block size  $N$  tends to infinity [294]. It should be noted that even if the assumptions do not hold exactly (images are not first-order Gauss-Markov), the DCT has proven to be a robust approximation to the KLT, and is used in several standards for speech, image and video compression as we shall see.

The DCT also has shortcomings. One must block the input stream in order to perform the transform and this blocking is quite arbitrary. The block boundaries often create not only loss of compression (correlation across the boundaries is not removed) but also annoying blocking effects. This is one of the reasons for using lapped transforms and subband or wavelet coding schemes. However, the goal of these generalized transforms is the same, namely, to create decorrelated outputs from a correlated input stream, and then to quantize the outputs separately.

**Discussion** We recall that decorrelation leads to independence only if the input is Gaussian (see Appendix 7.A). Also, even independent random variables are better quantized as a block (or as a vector) than as independent scalars, due to sphere packing gains (see discussion of vector quantization in Section 7.1.2). However, the complexity of doing so is high, and thus, scalar quantization is often preferred. It will be shown below, after a discussion of quantization and bit allocation, that the KLT is the optimal linear transformation (under certain assumptions) among block transforms. The performance of subband coding will also be analyzed.

The major point is that all these schemes are unitary transformations on the input and thus, if  $\hat{x}$  and  $\hat{y}$  are the approximate versions of  $x$  and  $y$ , respectively, we always have (similarly to (7.1.9))

$$\|x - \hat{x}\| = \|y - \hat{y}\|. \quad (7.1.12)$$

Note that nonorthogonal systems (such as linear phase biorthogonal filter banks) are usually designed to almost satisfy (7.1.12). If they do not, there

**Coding Gain** Now that we have discussed quantization and bit allocation, we can return to our study of transform coding and see what advantage is obtained by doing quantization in the transform domain (see Figure 7.1).

First, recall that the Karhunen-Loève transform leads to uncorrelated variables with variance  $\lambda_i$  (see (7.1.8)). Assume that the input to the transform is zero-mean Gaussian with variance  $\sigma_x^2$ , and that fine quantization is used. This leads us to Proposition 7.2.

**PROPOSITION 7.2** *Optimality of Karhunen-Loève Transform*

Among all block transforms and at a given rate, the Karhunen-Loève transform will minimize the expected distortion.

**PROOF**

After the KLT with optimal scalar quantization and bit allocation, the total distortion for all  $N$  channels is (following (7.1.20)),

$$D_{KLT} = N \cdot C \cdot 2^{-2\bar{R}} \cdot \rho^2 = N \cdot C \cdot 2^{-2\bar{R}} \left( \prod_{i=0}^{N-1} \lambda_i \right)^{1/N}, \quad (7.1.21)$$

where  $C = \sqrt{3\pi}/2$  (see (7.1.18)). Since the determinant of a matrix is equal to the product of its eigenvalues, the last term is equal to  $(\det(\mathbf{K}_x))^{1/N}$  where  $\mathbf{K}_x$  is the autocovariance matrix (assuming zero mean,  $\mathbf{K}_x = \mathbf{R}_x$ ). To prove the optimality of the KLT, we need the following inequality for the determinant of an autocorrelation matrix of  $N$  zero-mean variables with variances  $\sigma_i^2$  [109]:

$$\det(\mathbf{R}_x) \leq \prod_{i=0}^{N-1} \sigma_i^2, \quad (7.1.22)$$

with equality if and only if  $\mathbf{R}_x$  is diagonal. It turns out that the more correlated the variables are, the smaller the determinant.

Consider now an arbitrary orthogonal transform, with transform variables having variance  $\sigma_i^2$ . The distortion is

$$D_T = N \cdot C \cdot 2^{-2\bar{R}} \left( \prod_{i=0}^{N-1} \sigma_i \right)^{1/N}.$$

Because of (7.1.22) and the fact that the determinant is conserved by unitary transforms, this is greater or equal than

$$D_T \geq N \cdot C \cdot 2^{-2\bar{R}} \det(\mathbf{R}_x)^{1/N}.$$

Since the KLT achieves a diagonal  $\mathbf{R}_x$ , then the equality is reached by the KLT following (7.1.21). This proves that if the input to the transform is Gaussian and the quantization is fine, the KLT is optimal among all unitary transforms.

What is the gain we just obtained? If the samples are directly quantized, the distortion will be

$$D_{PCM} = N \cdot C \cdot 2^{-2\bar{R}} \cdot \sigma_x^2, \quad (7.1.23)$$

(where PCM stands for pulse code modulation, that is, sample-by-sample quantization) and the coding gain due to optimal transform coding is

$$\frac{D_{PCM}}{D_{KLT}} = \frac{\sigma_x^2}{\left(\prod_{i=0}^{N-1} \sigma_i^2\right)^{1/N}} = \frac{1/N \sum_{i=0}^{N-1} \sigma_i^2}{\left(\prod_{i=0}^{N-1} \sigma_i^2\right)^{1/N}}, \quad (7.1.24)$$

where we used the fact that  $N \cdot \sigma_x^2 = \sum \sigma_i^2$ . Recalling that the variances  $\sigma_i^2$  are the eigenvalues of  $R_x$ , it follows that the coding gain is the ratio of the arithmetic and geometric means of the eigenvalues of the autocorrelation matrix (under the zero-mean assumption). The lower bound on the gain is 1, which is attained only if all eigenvalues are identical.

Subband coding, being a generalization of transform coding, has a similar behavior. If the input is Gaussian, the channel signals are Gaussian as well. If the filters are ideal bandpass filters, the channels will be decorrelated. In any case, the distortion resulting from optimally allocating  $R = N \cdot \bar{R}$  bits across  $N$  channels with variances  $\sigma_i^2$  is, as in the usual transform case

$$D_{SBC} = N \cdot C \cdot 2^{-2\bar{R}} \cdot \rho^2,$$

where  $\rho$  is the geometric mean of the subband variances. Using (7.1.23) for direct quantization we get, similarly to (7.1.24), the subband coding gain as

$$\frac{D_{PCM}}{D_{SBC}} = \frac{1/N \sum_{i=0}^{N-1} \sigma_i^2}{\left(\prod_{i=0}^{N-1} \sigma_i^2\right)^{1/N}},$$

where the  $\sigma_i^2$ 's are the subband variances. That is, if the spectrum is far from being flat, there will be a large coding gain in subband methods. This is to be expected, since it becomes possible to match the spectral characteristics of the signal very closely, unlike in a sample-domain quantization. It is worthwhile to note that when the number of channels grows to infinity, both transform and subband coding achieve the theoretical performance of predictive coding with infinitely long predictor [143].

The obvious question is of course how do transform and subband coding compare? The ratio of  $D_{KLT}$  and  $D_{SBC}$  is:

$$\frac{D_{KLT}}{D_{SBC}} = \frac{\rho_{KLT}^2}{\rho_{SBC}^2},$$

that is, the one with the smaller geometric mean wins. Qualitatively, the one with the larger spread in variances will achieve better coding gain. The exact comparison thus requires measurements of variances in specific transforms (such as the DCT) versus filter banks (of finite length rather than ideal ones).

While the above considerations use some idealized assumptions, the concept holds true in general: The wider the variations between the component signals (transform coefficients or subbands), the higher the potential for coding gain. More about the above can be found in [5, 220, 273, 292, 295].

### 7.1.3 Entropy Coding

The last step in transform coding as shown in Figure 7.1 is entropy coding. Similarly to the first step, it is reversible and thus, there is no approximation problem as in quantization. After quantization, the variables take values drawn from a finite set  $\{a_i\}$ . The idea is to find a reversible mapping  $M$  to a new set  $\{b_i\}$  such that the average number of bits/symbol is minimized. A historical example is the Morse code which assigns short codes to the letters that appear frequently in the English language while reserving long codes to less frequent ones. The parameters in searching for the mapping  $M$  are the probabilities of occurrence of the symbols  $a_i$ ,  $p(a_i)$ . If the quantized variable is stationary, these probabilities are fixed, and a fixed mapping such as Huffman coding can be used. If the probabilities evolve over time, more sophisticated adaptive methods such as adaptive arithmetic coding can be used. Such mappings will transform fixed-length codewords into variable-length ones, creating a variable-length bit stream. If a constant bit rate channel is used, buffering has to smooth out variations so as to accommodate the fixed-rate channel.

**Huffman Coding** Given an alphabet  $\{a_i\}$  of size  $M$  and its associated probabilities of occurrence  $p(a_i)$ , the goal is to find a mapping  $b_i = F(a_i)$  such that the average length  $l(b_i)$  is minimized:

$$E(l(b_i)) = \sum_{i=0}^{M-1} p(a_i) l(b_i). \quad (7.1.25)$$

We also require that a sequence of  $b_i$ 's should be uniquely decodable (note that invertibility of  $F$  is not sufficient). This last requirement puts an extra constraint on the codewords  $b_i$ , namely, no codeword is allowed to be a prefix to another one. Then, the stream of  $b_i$ 's can be uniquely decoded by sequentially removing codewords  $b_i$ . The lower bound of the expected length (7.1.25) is given by the entropy of the set  $\{a_i\}$

$$H_a = - \sum_{i=0}^{M-1} p(a_i) \log_2(p(a_i)). \quad (7.1.26)$$

Huffman's construction elegantly meets the prefix condition while coming quite close to the entropy lower bound. The design is guided by the following property

The filter bank is based on the cosine modulated banks presented in Section 3.4.3, with window switching. The psychoacoustic analysis provides a noise threshold for  $L$  (Left),  $R$  (Right),  $S$  (Sum) and  $D$  (Difference) channels, where  $S = L + R$  and  $D = L - R$ . One feature of the PAC algorithm is that it is adaptive in time and frequency since, in each frequency band, it sends either the  $(L, R)$  or  $(S, D)$  signals, depending on which one is more efficient.

This coder provides transparent or near-transparent quality coding at 192 Kbits/sec/stereo pair, and high-quality coding at 128 Kbits/sec/stereo pair.

**AC System** Two well-known algorithms for high-quality audio compression are the AC-2 and AC-3 algorithms, coming from Dolby [34, 290]. They have both stereo and five-channel, surround system, versions.

The AC-2 version exploits both the time-domain and frequency-domain psychoacoustic models. It uses a time-frequency division scheme, achieving a trade-off between time and frequency resolutions, on a signal-dependent basis. This is achieved by selecting the optimal transform block length for each 10ms analysis interval. The filter bank is based again on the cosine-modulated filter bank [229, 228]. This coder operates at a variety of bit rates ranging from 64-192 Kbits/sec/channel. The 128 Kbits/sec/channel AC-2 version has been selected for use in a new multichannel NTSC compression system [34].

As can be seen from the above three examples, filter bank methods had a substantial impact on audio compression systems. Note that sophisticated time-frequency analysis is a key component.

### 7.3 IMAGE COMPRESSION

Multiresolution techniques are most naturally applied to images, where notions such as resolution and scale are very intuitive. Multiresolution techniques have been used in computer vision for tasks such as object recognition and motion estimation as well as in image compression, with pyramid [41] and subband coding [111, 314, 337]. An important feature of such image compression techniques is their successive approximation property: As higher frequencies are added (which is equivalent to more bands in subband coding or, difference signals in pyramids), higher-resolution images are obtained. Note that multiresolution successive approximation corresponds to the human visual system which helps the multiresolution techniques in terms of perceptual quality. Transform coding also has a successive approximation property (see the discussion on the Karhunen-Loève transform in Section 7.1.1) and is thus part of this broad class of techniques which are characterized by multiresolution approximations. In short, besides good compression capabilities, these schemes allow partial decoding of the coded version which lead to usable subresolution

approximations.

We start by discussing the standard image compression schemes, which are based on block transforms such as the discrete cosine transform (DCT) or overlapping block transforms such as the lapped orthogonal transform. This leads naturally to a description of the current image compression standard based on the DCT, called JPEG [148, 327], indicating some of the constraints of a “real-world” compression system.

We continue by discussing pyramid coding, which is a very simple but flexible image coding method. A detailed treatment of subband/wavelet image coding follows. Several important issues pertaining to the choice of the filters, the decomposition structure, quantization and compression are discussed and some examples are given.

Following these standard coding algorithms, we describe some more recent and sometimes exploratory compression schemes which use multiresolution as an ingredient. These include image compression methods based on wavelet maximums [184], and a method using adaptive wavelet packets [15, 233]. We also discuss some recent work on a successive approximation method for image coding using subband/wavelet trees [259], quantization error analysis in a subband system [331], joint design of quantization and filtering for subband coding [162], and nonorthogonal subband coding [200].

Note that in all experiments, we use the standard image *Barbara*, with  $512 \times 512$  pixels and 8-bit gray-scale values (see Figure 7.13). For comparison purposes, we will use the peak signal-to-noise ratio ( $SNR_p$ ) given by (7.1.3).

### 7.3.1 Transform and Lapped Transform Coding of Images

We have introduced block transforms in Section 3.4.1, and while they are a particular case of filter banks (with filter length  $L$  equal to the downsampling factor  $N$ ), they are usually considered separately. Their importance in practical image coding applications is such that a detailed treatment is justified. As we mentioned in audio coding examples, lapped orthogonal transforms are also filter bank expansions since they use modulated filter banks with filters of length typically twice the downsampling factor, or  $L = 2N$ . They have been introduced as an extension of block transforms in order to solve the problem of blocking in transform coding. Because of this close relationship between block transforms and lapped transforms, quantization and entropy coding for both schemes are usually very similar. A text on transform coding of images is [54], and lapped transform coding is treated in [188].

**Block Transforms** Recall that unitary block transforms of size  $N \times N$  are defined by  $N$  orthonormal basis vectors, that is, the transform matrix  $T$  has these basis



Figure 7.13 Standard image used for the image compression experiments, called *Barbara*. The size is  $512 \times 512$  pixels and 8 bits/pixel.

vectors as its rows (see Section 3.4.1 and (7.1.4)). For two-dimensional signals, one usually takes a separable transform which corresponds to the Kronecker product of  $T$  with itself,

$$T_{2D} = T \otimes T.$$

In other words, this separable transform can be evaluated by taking one-dimensional transforms along the rows and columns of a block  $B$  of an image. This can be written as:

$$B_T = TBT^T,$$

where the first product corresponds to transforming the columns, while the second product computes the transform on rows of the image block. Many transforms have been proposed for the coding of images. Besides the DCT given in (7.1.10–7.1.11), the sine, slant, Hadamard and Haar transform are common candidates, the last two mainly because of their low computational complexity (only additions and subtractions are involved). All of the transforms have fast,  $O(N \log N)$  algorithms, as opposed to the optimal KLT which has  $O(N^2)$  complexity and is signal dependent. The performance of the DCT in image compression is sufficiently close to that of the KLT as well as superior to other transforms so that it has become the standard transform. Figure 7.14 shows the  $8 \times 8$  DCT transform of the original image. Note the two representations shown. In part (a), we display the transform of each block of the image, while part (b) has gathered all coefficients of the same frequency into a block. This latter representation is simply a subband interpretation of the DCT; for example, the lowest left corner is the output of a filter which takes the average of  $8 \times 8$  blocks. The similarity of this representation with subband-decomposed images is obvious. Note that for quantization and entropy coding purposes, the representation (a) is preferred.

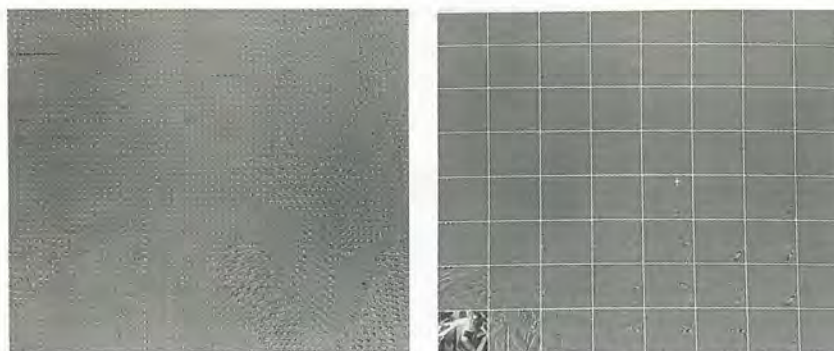


Figure 7.14  $8 \times 8$  DCT transform of the original image. On the left is the usual block-by-block representation and on the right is the reordering of the coefficients so that same frequencies appear together (subband interpretation of DCT). The lowest frequency is in the lower left corner.

The quantization in the DCT domain is usually scalar and uniform. The lowest two-dimensional frequency component, called the DC coefficient, is treated with particular care. According to (7.1.10), it corresponds to the local average of the block. Mismatches between blocks often lead to the feared blocking effect, that is, the boundaries between the blocks become visible, a visually annoying artifact. Because the DC coefficient has the highest energy, a fine scalar quantization leads to a large entropy. Also, as can be seen in Figure 7.14(b), there is still high correlation among DC coefficients (it resembles the original image). Therefore, predictive quantization, such as the DPCM, of the DC coefficients is often used to increase compression without increasing distortion.

The choice of the quantization steps for the various coefficients of the DCT is a classic bit-allocation problem, since distortion and rate are additive. However, perceptual factors are very important and careful experiments lead to quantization matrices which take into account the visibility of errors (besides the variance and entropy of the coefficients). While this has the flavor of a weighted MSE bit-allocation method, it relies heavily on experimental results. An example quantization matrix, showing the quantizer step sizes used for various DCT coefficients in JPEG, is given in Table 7.3 [148]. What is particularly important is the relative size of the steps, because within a certain range one can scale this quantization matrix, that is, multiply all step sizes by a scale factor greater or smaller than one in order to reduce or increase the bit rate, respectively. This scale factor is very useful for adaptive quantization, where the bit allocation is made between blocks which have various energy levels.

### 7.3.3 Subband and Wavelet Coding of Images

The generalization of subband decomposition to multiple dimensions is straightforward, especially in the separable case [314]. The application to compression of images has become popular [1, 111, 265, 330, 332, 335, 337]. The nonseparable multidimensional case, using quincunx [314] or hexagonal downsampling [264], as well as directional decompositions [19, 287], has also found applications in image compression. Recently, using filters specifically designed for regularity, methods closely related to subband coding have been proposed under the name of wavelet coding [14, 79, 81, 101, 176, 244, 260, 341]. The main difference with pyramid coding, discussed in Section 7.3.2, is that we have a critically sampled scheme and often an orthogonal decomposition. The price paid is more constrained filters in the decomposition, which leads to poorer coarse resolution pictures in general. In what follows, we discuss various forms of subband and wavelet compression schemes tailored to images.

**Separable Decompositions** We will call separable decompositions those which use separable downsampling. Usually, they also use separable filters (but this is not necessary). When both downsampling and filters are separable, the implementation is very efficient since it can be done on rows and columns separately, at least at each stage of the decomposition.

While being constrained, separable systems are often favored because of their computational efficiency with separable filters, since size- $N \times N$  filters lead to order  $N$  rather than  $N^2$  operations/input sample (see Section 6.2.4). Conceptually, separable systems are also much easier to implement since they are cascades of one-dimensional systems. However, from the fact that the two-dimensional filters are products of one-dimensional filters, it is clear that only rectangular pieces of the spectrum can be isolated.

**Nonseparable Decompositions** Recall that coding gain in subband coding was maximized when the variances in the channels were as different as possible (see Section 7.1.2). If one assumes that images have a power spectrum that is roughly rotationally-invariant and decreases with higher frequencies, then it is clear that separable systems are not best suited for isolating a lowpass channel containing most energy and having highpass channels with low energy. A better solution is found by opting for nonseparable systems. The two most important systems for image processing are based on the quincunx [314] and hexagonal downsamplings [264], for two- and four-channel subband coding systems, respectively. Quincunx and hexagonal sublattices of  $\mathbb{Z}^2$  are shown in Figure 7.20, together with the more conventional separable sublattice. They correspond to

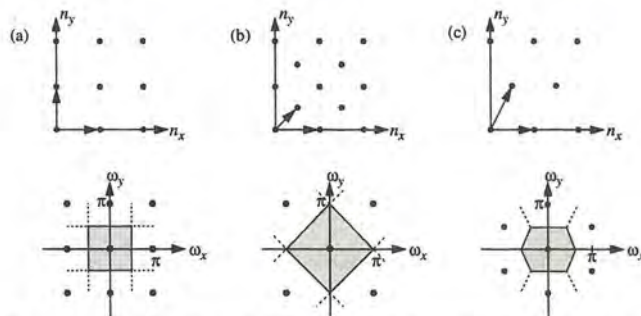


Figure 7.20 Sublattices of  $\mathbb{Z}^2$  and shapes of possible ideal lowpass filters (corresponding to the Voronoi cell of the dual lattice, which is indicated as well). (a) Separable sublattice  $D_S$ . (b) Quincunx  $D_Q$ . (c) Hexagonal  $D_H$ .

integer linear combinations of the columns of the following matrices<sup>6</sup>:

$$D_S = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \quad D_Q = \begin{pmatrix} 2 & 1 \\ 0 & 1 \end{pmatrix}, \quad D_H = \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix},$$

where the sampling density is reduced by a factor of four for the separable sampling, two for the quincunx sampling (see also Appendix 3.B) and by a factor of four for the hexagonal sampling. The repeated spectrums in Fourier domain due to downsampling appear on the dual lattice, which is given by the transposed inverse of the lattice matrix. Also shown in Figure 7.20 are possible ideal lowpass filters that will avoid aliasing when downsampling to these sublattices. If, as we said, images have circularly symmetric power spectrums that decrease with higher frequencies, then the quincunx lowpass filter will retain more of the original signal's energy than a separable lowpass filter (which would be one-dimensional since the downsampling is by two). Using the same argument, the hexagonal lowpass filter is then better than the corresponding lowpass filter in a separable system with downsampling by two in each dimension. Thus, these nonseparable systems, while being more difficult to design and more complex to implement, represent a better match to usual image spectrums.

Furthermore, the simple quincunx case has the following perceptual advantage: The human visual system is more accurate in horizontal and vertical high frequencies than along diagonals. The lowpass filter in Figure 7.20(b) conserves horizontal and vertical frequencies, while it cuts off diagonals to half of their original range. This is a good match to the human eye and often, the highpass channel (which is complementary to the lowpass channel) can

<sup>6</sup>Recall from Appendix 3.B, that a given sampling lattice may have infinitely many matrix representations.

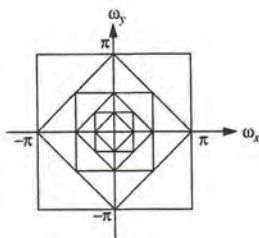


Figure 7.21 Frequency decomposition of iterated quincunx scheme.

be disregarded altogether. That is, a compression by a factor of two can be achieved with no visible degradation. Such preprocessing has been used in intraframe coding of HDTV [12]. The above quincunx scheme is often iterated on the lowpass channel, leading to a frequency decomposition as shown in Figure 7.21. This actually corresponds to a two-dimensional nonseparable wavelet decomposition [164] and has been used for image compression [14].

The hexagonal system, besides having a fairly good approximation to a circularly symmetric lowpass, has three directional channels which can be used to detect directional edges [264]. However, the goal of an isotropic analysis is only approximated, since the horizontal and vertical directions are not treated in the same manner (see Figure 7.20(c)). Therefore, it is not clear if the added complexity of a nonseparable four-channel system based on the hexagonal sublattice is justified for coding purposes.

**Choice of Filters** Unlike in audio compression, the filters for image subband coding do not need high out-of-band rejection. Instead, a number of other constraints have to be satisfied.

**Linear phase** In regular image filtering, the need for linear phase is well-known since without linear phase, the phase distortion around edges is very visible. Therefore, the use of linear phase filters in subband coding has been often advocated [14]. Recall from Section 3.2.4, that in two-band FIR systems, linear phase and orthogonality are mutually exclusive and this carries over to four-band separable systems which are most often used in practice.

However, the case for linear phase is not as obvious as it seems at first sight. For example, in the absence of quantization, the phase of the filters has no bearing since the system has perfect reconstruction. This argument carries over for fine quantization as well. In the case of coarse quantization, the situation is more complex. One scenario is to consider the highpass channel as being set to zero. Look at the two impulse responses of this system. Nonlinear phase systems lead to nonsymmetric responses, but so do some of the linear phase

systems. Only if the filters meet additional constraints do the two impulse responses remain symmetric. Note also, that for computational purposes, linear phase is more convenient because of the symmetry of the filters.

Note that orthogonal FIR filters of sufficient length can be made almost linear phase by appropriate factorization of their autocorrelation function. Also, there are nonseparable orthogonal filters with linear phase. Finally, by resorting to the IIR filters, one can have both linear phase and orthogonality, and such noncausal IIR filters can be used in image processing without problems since we are dealing with finite-length input signals.

*Orthogonality* Orthogonal filters implement a unitary transform between the input and the subbands. The usual features of unitary transforms hold, such as conservation of energy. In particular, the total distortion is the sum of the subband distortions, or:

$$D = \sum_i D_i, \quad (7.3.1)$$

and the total bit rate is the sum of all the subband's bit rates. Therefore, optimal bit-allocation algorithms which assume additivity of bit rate and distortion can be used (see Section 7.1.2). In the nonorthogonal case, (7.3.1) does not hold, and thus, these bit allocation algorithms cannot be used directly. It should be noted that well designed linear phase FIR filter banks (that is, with good out-of-band rejection) are often close to being orthogonal and thus satisfy (7.3.1) approximately.

*Filter size* Good out-of-band rejection or high regularity require long filters. Besides their computational complexity, long filters are usually avoided because they tend to spread coding errors. For example, sharp edges introduce distortions because high-frequency channels are coarsely quantized. If the filters are long (and usually their impulse response has several sign changes), this causes an annoying artifact known as ringing around edges. Therefore, filters used in audio subband compression, such as length-32 filters, are too long for image compression. Instead, shorter "smooth" filters are preferred. Sometimes both their impulse and their step response are considered from a perceptual point of view [167]. The step response is important since edges in images will generate step responses at least in some of the channels. Highly oscillating step responses will require more bits to code, and coarse quantization will produce oscillations which are related to the step response. As can already be seen from this short discussion, there is an intertwining between the choice of filters and the type of quantization that follows. However, it is clear that the frequency-domain criteria used in audio (sharp cut-off, strong

out-of-band rejection) have little meaning in the image compression context, where time-domain arguments such as ringing, are more important.

*Regularity* An orthogonal filter with a certain number of zeros at the aliasing frequency ( $\pi$  in the two-channel case) is called regular if its iteration tends to a continuous function (see Section 4.4). The importance of this property for coding is potentially twofold when the decomposition is iterated. First, the presence of many zeroes at the aliasing frequency can improve the coding gain and second, compression artifacts might be less objectionable. To investigate the first effect, Rioul [243] compared the compression gain for filters of varying regularity used in a wavelet coder, or octave-band subband coder, with four stages. The experiment included bit allocation, quantization, and entropy coding and is thus quite realistic. The results are quite interesting: Some regularity is desired (the performance with no regularity is poor) and higher regularity improves compression further (but not substantially).

As for the compression artifacts, the following argument shows that the filters should be regular when an octave-band decomposition is used: Assume a single quantization error in the lowpass channel. This will add an error to the reconstructed signal which depends only on the equivalent — iterated lowpass filter. If the iterated filter is smooth, this will be less noticeable than if it is a highly irregular function (even though both contribute the same MSE). Note also that the lowest band is upsampled  $2^{i-1}$  times (where  $i$  is the number of iterations) and thus, the iterated filter's impulse response is shifted by large steps, making irregular patterns in the impulse response more visible.

In the case of biorthogonal systems such as linear phase FIR filter banks, one is often faced with the case where either the analysis or the synthesis is regular, but not both. In that case, it is preferable to use the regular filter at the synthesis, by the same argument as above. Visually, an irregular analysis is less noticeable than an irregular synthesis, as can be verified experimentally.

When the decomposition is not iterated, regularity is of little concern. A typical example is the lapped orthogonal transform, that is, a multi-channel filter bank which is applied only once.

*Frequency selectivity* What is probably the major criterion in audio subband filter design is of much less concern in image compression. Aliasing, which is a major problem in audio, is much less disturbing in images [331]. The desire for short filters limits the frequency selectivity as well. One advantage of frequency selectivity is that perceptual weighting of errors is easier, since errors will be confined to the band where they occur.

In conclusion, subband image coding requires relatively short and smooth filters, with some regularity if the decomposition is iterated.

|        |        |        |        |
|--------|--------|--------|--------|
| LL, HL | LH, HL | HH, HL | HL, HL |
| LL, HH | LH, HH | HH, HH | HL, HH |
| LL, LH | LH, LH | HH, LH | HL, LH |
| LL, LL | LH, LL | HH, LL | HL, LL |

Figure 7.23 Uniform subband decomposition of an image into 16 subbands. The spectral decomposition and ordering of the channels is shown. The first two letters correspond to horizontal filtering and the last two to vertical filtering. LH, for example, means that a lowpass is used in the first stage and a highpass in the second. The ordering is such that frequencies increase from left to right and from bottom to top.

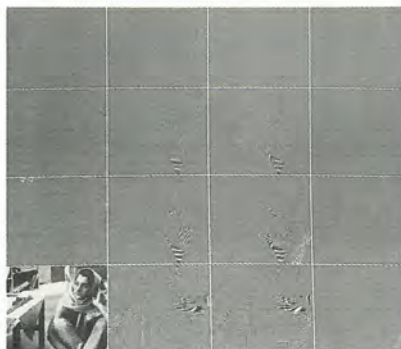


Figure 7.24 Uniform subband decomposition of the *Barbara* image. The ordering of the subbands is given in Figure 7.23.

Because higher bands tend to have large amounts of zeros (especially after deadzone quantizers), run-length coding and an end of block symbol can be used to increase compression substantially.

**Examples** Two typical coding examples will be described in some detail. The first is a uniform separable decomposition. The second is an octave-band or constant relative bandwidth decomposition (often called a wavelet decomposition).

*Uniform decomposition* By using a separable decomposition into four bands and iterating it once, we obtain 16 subbands as shown in Figure 7.23. The resulting subband images are shown in Figure 7.24. The filters used are linear phase length-12 QMF's [144] and the image was symmetrically extended before filtering. The variances of the samples in the bands are shown in Table 7.4. We

Table 7.4 Variances in the various bands of a uniform decomposition (defined as in Figure 7.23).

|    | LL      | LH      | HH      | HL      |
|----|---------|---------|---------|---------|
| HL | 0.58959 | 0.86237 | 1.77899 | 0.88081 |
| HH | 2.87483 | 6.71625 | 8.56729 | 3.25402 |
| LH | 23.5474 | 33.4055 | 60.9195 | 14.8490 |
| LL | 2711.45 | 56.0058 | 52.5202 | 13.9685 |

Table 7.5 Step sizes for the quantizers in the various bands (as defined in Figure 7.23), for a target rate of 0.5 bits/pixel. The lowest band was JPEG coded, and the step size corresponds to the quality factor (QF) used in JPEG.

|    | LL    | LH     | HH     | HL     |
|----|-------|--------|--------|--------|
| HL | 9.348 | 8.246  | 8.657  | 22.318 |
| HH | 8.400 | 10.161 | 8.887  | 13.243 |
| LH | 6.552 | 7.171  | 10.805 | 16.512 |
| LL | QF-89 | 8.673  | 11.209 | 15.846 |

code the lowest subband (LL,LL) with JPEG (see Section 7.3.1). For all other bands, we use uniform quantization with a dead zone of twice the step size used elsewhere. Using a set of step sizes, one can derive rate-distortion curves by measuring the entropy of the resulting quantized channels. A true operational rate-distortion curve would have to include run-length coding and actual entropy coding. Based on these rate-distortion curves, one can perform an optimal constant-slope bit allocation, that is, one can choose the optimal quantizer step sizes for the various bands. The step sizes for a budget of 0.5 bits/pixel are listed in Table 7.5. A set of Huffman codes and run-length codes are designed for each subband channel. Note that the special symbol “start of run” (SR) is entropy coded as any other nonzero pixel. Altogether, one obtains the final rate of 0.497 bits/pixel (the difference in rate comes from the fact that bit allocation was based on entropy measures). Then, the coded image has  $SNR_p$  of 30.38 dB. Figure 7.27 (top row) shows the compressed *Barbara* image and a detail at the same rate.

*Octave-band decomposition* Instead of uniformly decomposing the spectrum of the image, we iterate a separable four-band decomposition three times. The resulting split of the spectrum is shown in Figure 7.25, together with the subband images in Figure 7.26. Here, we used the Daubechies’ maximally flat

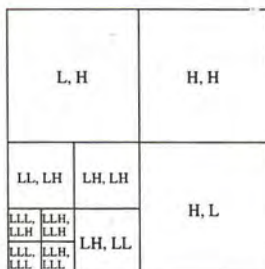


Figure 7.25 Octave-band or wavelet decomposition of an image into unequal subbands. The spectral decomposition and ordering of the channels is shown.



Figure 7.26 Subband images corresponding to the spectral decomposition shown in Figure 7.25.

orthogonal filters of length 8. At the boundaries, we used periodic extension. The variances in the bands are shown in Table 7.6. Histograms of pixel values of the bands are similar to the ones in a uniform decomposition. Because the lowest band (LLL, LLL) is small enough ( $64 \times 64$  pixels), we use scalar quantization on it as on all other bands. Again, uniform quantizers with double-sized dead zone are used and rate-distortion curves are derived for bit-allocation purposes. The resulting step sizes for the target bit rate of 0.5 bits/pixel are given in Table 7.7.

The development of entropy coding (including run-length coding for higher bands) is similar to the uniform-decomposition case discussed earlier. The final rate is 0.499 bits/pixel, with  $SNR_p$  of 29.21 dB. The coded image and a detail are shown in Figure 7.27 (bottom row). Note that there is little difference between the uniform and the octave-band decomposition results.

We would like to emphasize that the above examples are “textbook ex-



Figure 7.27 Compression results on *Barbara* image. Top left: Subband coding in 16 uniform bands at 0.4969 bits/pixel and  $SNR_p = 30.38$  dB. Top right: Detail of top left. Bottom left: Octave-band or wavelet compression at 0.4990 bits/pixel and  $SNR_p = 29.21$  dB. Bottom right: Detail of bottom left.

amples” for illustration purposes. For example, no statistics over large sets of images were taken and thus, the entropy coders might perform poorly for a substantially different image. The aim was more to demonstrate the ingredients used in a subband/wavelet image coder.

State of the art coders, which can be found in the current literature, improve substantially the results shown here. Major differences with respect to the simple coders we discussed so far are the following:

### 7.4.3 Pyramid Coding of Video

The difficulty of including motion in three-dimensional subband coding will be discussed shortly. It turns out that it is much easier to include motion in pyramid coding, due to the fact that the prediction or interpolation from low resolution to full resolution (see Figure 7.18) can be an arbitrary predictor [9], such as a motion based one. This is a general idea which can be used in various forms for video compression and we will describe a particular scheme as an example.

This video compression scheme was studied in [301, 302, 303]. Consider a progressive video sequence and its subresolutions, obtained by spatial filtering and downsampling as well as frame skipping over time. Note that filtering over time would create so-called “double images” when there is motion and thus straight downsampling in time is preferable. This is shown schematically in Figure 7.34(a), where the resolution is decreased by a factor of two in each dimension between one level of the pyramid and the next. Now we apply the classic pyramid coding scheme, which consists of the following:

- (a) Coding the low resolution..
- (b) Predicting the higher resolution based on the coded low resolution.
- (c) Taking the difference between the predicted and the true higher resolution, resulting in the prediction error.
- (d) Coding the prediction error.

While these steps could be done in the three dimensions at once, it is preferable to separate the spatial and temporal dimensions. First, the spatial dimension is interpolated using filtering and then the temporal dimension is interpolated using motion-based interpolation. This is shown in Figure 7.34(b). Following each interpolation step, the prediction error is computed and coded and this coded value is added to the prediction before going to the next step. Because at each step, we use coded versions for our prediction, we have a pyramid scheme with quantization noise feedback, as was described in Figure 7.19. Therefore, there is only one source of error, namely the compression of the last prediction error.

The oversampling inherent in pyramid coding is not a problem in the three-dimensional case, since, following (3.5.4), we have a total number of samples which increases only as

$$(1 + \frac{1}{8} + \frac{N}{8^2} + \cdots)N < \frac{8}{7}N,$$

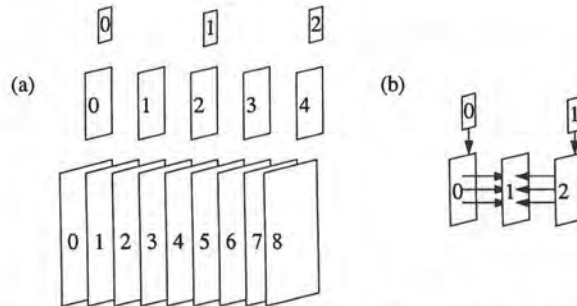


Figure 7.34 Spatio-temporal pyramid video coding. (a) Three layers of the pyramid, corresponding to three resolutions. (b) Prediction of the higher resolution. The spatial resolution is interpolated first (using linear filtering) and then the temporal resolution is increased using motion interpolation.

or at most 14%, since every coarser level has only 1/8th the number of samples of its predecessor.

The key technique in the spatio-temporal pyramid scheme is the motion interpolation step, which predicts a frame from its two neighbors based on motion vectors. Assume the standard rigid-object and pure translational motion model [207]. If we denote the intensity of a pixel at location  $\mathbf{r} = (x, y)$  and time  $t$  by  $I(\mathbf{r}, t)$ , we are looking for a mapping  $\mathbf{d}(\mathbf{r}, t)$  such that we can write:

$$I(\mathbf{r}, t) = I(\mathbf{r} - \mathbf{d}(\mathbf{r}, t), t - 1).$$

If motion is not changing over time, we also have:

$$I(\mathbf{r}, t) = I(\mathbf{r} + \mathbf{d}(\mathbf{r}, t), t + 1).$$

The goal is to find the function  $\mathbf{d}(\mathbf{r}, t)$ , that is, estimate the motion. This is a standard estimation procedure, where some simplifying assumptions are made (such as constant motion over a neighborhood). Typically, for a small block  $\mathbf{b}$  in the current frame, one searches over a set of possible motion vectors such that the sum of squared differences,

$$\sum_{\mathbf{r} \in \mathbf{b}} |I(\mathbf{r}, t) - \hat{I}(\mathbf{r}, t)|^2, \quad (7.4.1)$$

is minimized, where

$$\hat{I}(\mathbf{r}, t) = I(\mathbf{r} - \mathbf{d}_{\mathbf{b}}, t - 1), \quad (7.4.2)$$

corresponds to a block in the previous frame displaced by  $\mathbf{d}_{\mathbf{b}}$  (the motion for the block under consideration in the current frame). It is best to actually perform a symmetric search by considering the past (as in (7.4.2)), the future ((7.4.2)

with sign reversals for  $d_b$ ), and the average,

$$\hat{I}(r, t) = \frac{1}{2}[I(r - d_b, t - 1) + I(r + d_b, t + 1)],$$

and then to choose the best match. Choosing past or future for the interpolation is especially important for covering and uncovering of background due to moving objects, as well as in case of abrupt changes (scene changes).

Interestingly, a very successful technique to perform motion estimation (that is, finding the displacement  $d_b$  that minimizes (7.4.1)) is based on multiresolution or successive approximation. Instead of solving (7.4.1) directly, one solves a coarse version of the same problem, refines the solution (by interpolating the motion vector field), and uses this new field as a starting point for a new, finer search. This is not only computationally less complex, but also more robust in general [31, 302]. It is actually a regularization of the motion estimation problem.

As an illustration of this video coding scheme, a few representative pictures are shown. First, Figure 7.35 shows the successive refinement of the motion vector field, which starts with a sparse field on a coarse version and refines it to a fine field on the full-resolution image. In Figure 7.36, we show the resulting spatial and temporal prediction error signals. As can be seen, the spatial prediction error has higher energy than the temporal one, which shows that temporal interpolation based on motion is quite successful (actually, this sequence has high frequency spatial details, which cannot be well predicted from the coarse resolution).

A point to note is that the first subresolution sequence (which is downsampled by 2 in each dimension) is of good visual quality and could be used for a compatible coding scheme. This coding scheme was implemented for high quality coding of HDTV with a compatible subchannel and it performed well at medium compression (of the order of 10-15 to 1) with essentially no visible degradation [301, 303].

#### 7.4.4 Subband Decompositions for Video Representation and Compression

**Decompositions for Representation** We will discuss here two ways of sampling video by 2; the first, using quincunx sampling along (vertical, time)-dimensions and the second, true three-dimensional sampling by 2, using the FCO sampling lattice.

*Quincunx sampling for scanning format conversions* We have outlined previously the existence of different scanning standards (such as interlaced and progressive) as well as the desire for compatibility. A simple technique to deal with these problems is to use perfect reconstruction filter banks to go back and forth between

## Discrete-Time Bases and Filter Banks

*“What is more beautiful than the Quincunx,  
which, from whatever direction you look,  
is correct?”*  
— Quintilian

**O**ur focus in this chapter will be directed to series expansions of discrete-time sequences. The reasons for expanding signals, discussed in Chapter 1, are linked to signal analysis, approximation and compression, as well as algorithms and implementations. Thus, given an arbitrary sequence  $x[n]$ , we would like to write it as

$$x[n] = \sum_{k \in \mathcal{Z}} \langle \varphi_k, x \rangle \varphi_k[n], \quad n \in \mathcal{Z}.$$

Therefore, we would like to construct orthonormal sets of basis functions,  $\{\varphi_k[n]\}$ , which are complete in the space of square-summable sequences,  $l_2(\mathcal{Z})$ . More general, biorthogonal and overcomplete sets, will be considered as well.

The discrete-time Fourier series, seen in Chapter 2, is an example of such an orthogonal series expansion, but it has a number of shortcomings. Discrete-time bases better suited for signal processing tasks will try to satisfy two conflicting requirements, namely to achieve good frequency resolution while keeping good time locality as well. Additionally, for both practical and computational reasons, the set of basis functions has to be structured. Typically, the infinite set of basis functions  $\{\varphi_k\}$  is obtained from a finite number of prototype sequences and their shifted versions in time. This leads to discrete-time *filter banks* for the implementation of such structured expansions. This filter bank point of view has been central to the developments in the digital signal processing community, and to the design of good basis functions or filters in particular. While the expansion is not time-invariant, it will at least be *periodically*

*time-invariant*. Also, the expansions will often have a *successive approximation* property. This means that a reconstruction based on an appropriate subset of the basis functions leads to a good approximation of the signal, which is an important feature for applications such as signal compression.

Linear signal expansions have been used in digital signal processing since at least the 1960's, mainly as block transforms, such as piecewise Fourier series and Karhunen-Loève transforms [143]. They have also been used as overcomplete expansions, such as the short-time Fourier transform (STFT) for signal analysis and synthesis [8, 226] and in transmultiplexers [25]. Increased interest in the subject, especially in orthogonal and biorthogonal bases, arose with work on compression, where redundancy of the expansion such as in the STFT is avoided. In particular, subband coding of speech [68, 69] spurred a detailed study of critically sampled filter banks. The discovery of *quadrature mirror filters* (QMF) by Croisier, Esteban and Galand in 1976 [69], which allows a signal to be split into two downsampled subband signals and then reconstructed without aliasing (spectral foldbacks) even though nonideal filters are used, was a key step forward.

*Perfect reconstruction filter banks*, that is, subband decompositions, where the signal is a perfect replica of the input, followed soon. The first orthogonal solution was discovered by Smith and Barnwell [270, 271] and Mintzer [196] for the two-channel case. Fettweiss and coworkers [98] gave an orthogonal solution related to wave digital filters [97]. Vaidyanathan, who established the relation between these results and certain unitary operators (paraunitary matrices of polynomials) studied in circuit theory [23], gave more general orthogonal solutions [305, 306] as well as lattice factorizations for orthogonal filter banks [308, 310]. Biorthogonal solutions were given by Vetterli [315], as well as multidimensional quadrature mirror filters [314]. Biorthogonal filter banks, in particular with linear phase filters, were investigated in [208, 321] and multidimensional filter banks were further studied in [155, 164, 257, 264, 325]. Recent work includes filter banks with rational sampling factors [160, 206] and filter banks with block sampling [158]. Additional work on the design of filter banks has been done in [144, 205] among others.

In parallel to this work on filter banks, a generalization of block transforms called lapped orthogonal transforms (LOT's) was derived by Cassereau [43] and Malvar [186, 188, 189]. An attractive feature of a subclass of LOT's is the existence of fast algorithms for their implementation since they are modulated filter banks (similar to a "real" STFT). The connection of LOT's with filter banks was shown, in [321].

Another development, which happened independently of filter banks but turns out to be closely related, is the pyramid decomposition of Burt and Adel-

son [41]. While it is oversampled (overcomplete), it clearly uses multiresolution concepts, by decomposing a signal into a coarse approximation plus added details. This framework is central to wavelet decompositions and establishes conceptually the link between filter banks and wavelets, as shown by Mallat [179, 180, 181] and Daubechies [71, 73]. This connection has led to a renewed interest in filter banks, especially with the work of Daubechies who first constructed wavelets from filter banks [71] and Mallat who showed that a wavelet series expansion could be implemented with filter banks [181]. Recent work on this topic includes [117, 240, 319].

As can be seen from the above short historical discussion, there are two different points of view on the subject, namely, expansion of signals in terms of structured bases, and perfect reconstruction filter banks. While the two are equivalent, the former is more in tune with Fourier and wavelet theory, while the latter is central to the construction of implementable systems. In what follows, we use both points of view, using whichever is more appropriate to explain the material.

The outline of the chapter is as follows: First, we review discrete-time series expansions, and consider two cases in some detail, namely the Haar and the sinc bases. They are two extreme cases of two-channel filter banks. The general two-channel filter bank is studied in detail in Section 3.2, where both the expansion and the more traditional filter bank point of view are given. The orthogonal case with finite-length basis functions or finite impulse response (FIR) filters is thoroughly studied. The biorthogonal FIR case, in particular with linear phase filters (symmetric or antisymmetric basis functions), is considered, and the infinite impulse response (IIR) filter case (which corresponds to basis functions with exponential decay) is given as well.

In Section 3.3, the study of filter banks with more than two channels starts with tree-structured filter banks. In particular, a constant relative bandwidth (or constant-Q) tree is shown to compute a discrete-time wavelet series. Such a transform has a multiresolution property that provides an important framework for wavelet transforms. More general filter bank trees, also known as wavelet packets, are presented as well.

Filter banks with  $N$  channels are treated next. The two particular cases of block transforms and lapped orthogonal transforms are discussed first, leading to the analysis of general  $N$ -channel filter banks. An important case, namely modulated filter banks, is studied in detail, both because of its relation to short-time Fourier-like expansions, and because of its computational efficiency.

Overcomplete discrete-time expansions are discussed in Section 3.5. The pyramid decomposition is studied, as well as the classic overlap-add/save algorithm for convolution computation which is a filter bank algorithm.

Multidimensional expansions and filter banks are derived in Section 3.6. Both separable and nonseparable systems are considered. In the nonseparable case, the focus is mostly on two-channel decompositions, while more general cases are indicated as well.

Section 3.7 discusses a scheme that has received less attention in the filter bank literature, but is nonetheless very important in applications, and is called a transmultiplexer. It is dual to the analysis/synthesis scheme used in compression applications, and is used in telecommunications.

The two appendices contain more details on orthogonal solutions and their factorizations as well as on multidimensional sampling.

The material in this chapter covers filter banks at a level of detail which is adequate for the remainder of the book. For a more exhaustive treatment of filter banks, we refer the reader to the text by Vaidyanathan [308]. Discussions of filter banks and multiresolution signal processing are also contained in the book by Akansu and Haddad [3].

### 3.1 SERIES EXPANSIONS OF DISCRETE-TIME SIGNALS

We start by recalling some general properties of discrete-time expansions. Then, we discuss a very simple structured expansion called the Haar expansion, and give its filter bank implementation. The dual of the Haar expansion — the sinc expansion — is examined as well. These two examples are extreme cases of filter bank expansions and set the stage for solutions that lie in between.

Discrete-time series expansions come in various flavors, which we briefly review (see also Sections 2.2.3–2.2.5). As usual,  $x[n]$  is an arbitrary square-summable sequence, or  $x[n] \in l_2(\mathcal{Z})$ . First, *orthonormal expansions* of signals  $x[n]$  from  $l_2(\mathcal{Z})$  are of the form

$$x[n] = \sum_{k \in \mathcal{Z}} \langle \varphi_k[l], x[l] \rangle \varphi_k[n] = \sum_{k \in \mathcal{Z}} X[k] \varphi_k[n], \quad (3.1.1)$$

where

$$X[k] = \langle \varphi_k[l], x[l] \rangle = \sum_l \varphi_k^*[l] x[l], \quad (3.1.2)$$

is the *transform* of  $x[n]$ . The basis functions  $\varphi_k$  satisfy the orthonormality<sup>1</sup> constraint

$$\langle \varphi_k[n], \varphi_l[n] \rangle = \delta[k - l]$$

and the set of basis functions is complete, so that every signal from  $l_2(\mathcal{Z})$  can

<sup>1</sup>The first constraint is orthogonality between basis vectors. Then, normalization leads to orthonormality. The terms “orthogonal” and “orthonormal” will often be used interchangeably, unless we want to insist on the normalization and then use the latter.

be expressed using (3.1.1). An important property of orthonormal expansions is conservation of energy,

$$\|x\|^2 = \|X\|^2.$$

*Biorthogonal expansions*, on the other hand, are given as

$$\begin{aligned} x[n] &= \sum_{k \in \mathcal{Z}} \langle \varphi_k[l], x[l] \rangle \tilde{\varphi}_k[n] = \sum_{k \in \mathcal{Z}} \tilde{X}[k] \tilde{\varphi}_k[n], \\ &= \sum_{k \in \mathcal{Z}} \langle \tilde{\varphi}_k[l], x[l] \rangle \varphi_k[n] = \sum_{k \in \mathcal{Z}} X[k] \varphi_k[n], \end{aligned} \quad (3.1.3)$$

where

$$\tilde{X}[k] = \langle \varphi_k[l], x[l] \rangle \quad \text{and} \quad X[k] = \langle \tilde{\varphi}_k[l], x[l] \rangle$$

are the transform coefficients of  $x[n]$  with respect to  $\{\tilde{\varphi}_k\}$  and  $\{\varphi_k\}$ . The dual bases  $\{\varphi_k\}$  and  $\{\tilde{\varphi}_k\}$  satisfy the biorthogonality constraint

$$\langle \varphi_k[n], \tilde{\varphi}_l[n] \rangle = \delta[k - l].$$

Note that in this case, conservation of energy does not hold. For stability of the expansion, the transform coefficients have to satisfy

$$A \sum_k |X[k]|^2 \leq \|x\|^2 \leq B \sum_k |\tilde{X}[k]|^2$$

with a similar relation for the coefficients  $\tilde{X}[k]$ . In the biorthogonal case, conservation of energy can be expressed as

$$\|x\|^2 = \langle X[k], \tilde{X}[k] \rangle.$$

Finally, *overcomplete expansions* can be of the form (3.1.1) or (3.1.3), but with redundant sets of functions, that is, the functions  $\varphi_k[n]$  used in the expansions are not linearly independent.

### 3.1.1 Discrete-Time Fourier Series

The discrete-time Fourier transform (see also Section 2.4.6) is given by

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega \quad (3.1.4)$$

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}. \quad (3.1.5)$$

It is a series expansion of the  $2\pi$ -periodic function  $X(\omega)$  as given by (3.1.5), while  $x[n]$  is written in terms of an integral of the continuous-time function  $X(\omega)$ . While this is an important tool in the analysis of discrete-time signals and systems [211], the fact that the synthesis of  $x[n]$  given by (3.1.4) involves

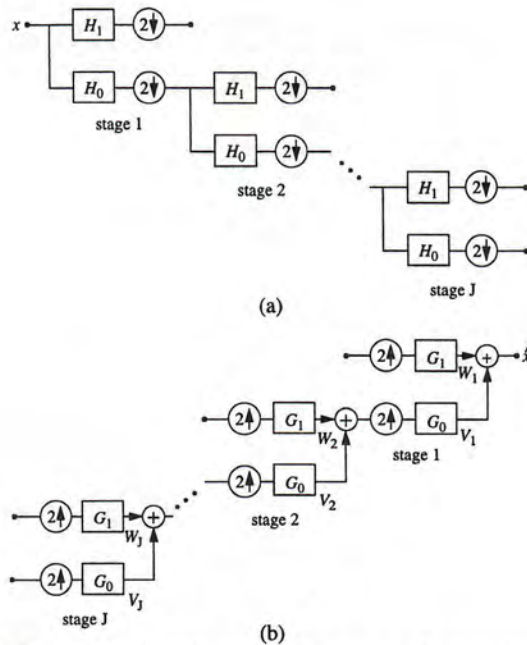


Figure 3.7 An octave-band filter bank with  $J$  stages. Decomposition spaces  $V_i$ ,  $W_i$  are indicated. If  $h_i[n]$  is an orthogonal filter, and  $g_i[n] = h_i[-n]$ , the structure implements an orthogonal discrete-time wavelet series expansion. (a) Analysis part. (b) Synthesis part.

### 3.3 TREE-STRUCTURED FILTER BANKS

An easy way to construct multichannel filter banks is to cascade two-channel banks appropriately. One case can be seen in Figure 3.7(a), where frequency analysis is obtained by simply iterating a two-channel division on the previous lowpass channel. This is often called a *constant- $Q$*  or constant relative bandwidth filter bank since the bandwidth at each channel, divided by its center frequency, is constant. It is also sometimes called a logarithmic filter bank since the channels are equal bandwidth on a logarithmic scale. We will call it an *octave-band filter bank* since each successive highpass output contains an octave of the input bandwidth. Another case appears when  $2^J$  equal bandwidth channels are desired. This can be obtained by a  $J$ -step subdivision into 2 channels, that is, the two-channel bank is now iterated on both the lowpass and highpass channels. This results in a tree with  $2^J$  leaves, each corresponding to  $(1/2^J)$ th of the original bandwidth, with a downsampling by  $2^J$ . Another

possibility is building an arbitrary tree-structured filter bank, giving rise to *wavelet packets*, discussed later in this section.

### 3.3.1 Octave-Band Filter Bank and Discrete-Time Wavelet Series

Consider the filter bank given in Figure 3.7. We see that the signal is split first via a two-channel filter bank, then the lowpass version is split again using the same filter bank, and so on. It will be shown later that this structure implements a discrete-time biorthogonal wavelet series (we assume here that the two-channel filter banks are perfect reconstruction). If the two-channel filter bank is orthonormal, then it implements an orthonormal discrete-time wavelet series.<sup>9</sup>

Recall that the basis functions of the discrete-time expansion are given by the impulse responses of the synthesis filters. Therefore, we will concentrate on the synthesis filter bank (even though, in the orthogonal case, simple time reversal relates analysis and synthesis filters). Let us start with a simple example which should highlight the main features of octave-band filter bank expansions.

#### Example 3.7

Consider what happens if the filters  $g_i[n]$  from Figure 3.7(a)-(b) are Haar filters defined in  $z$ -transform domain as

$$G_0(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}), \quad G_1(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}).$$

Take, for example,  $J = 3$ , that is, we will use three two-channel filter banks. Then, using the multirate identity which says that  $G(z)$  followed by upsampling by 2 is equivalent to upsampling by 2 followed by  $G(z^2)$  (see Section 2.5.3), we can transform this filter bank into a four-channel one as given in Figure 3.8. The equivalent filters are

$$\begin{aligned} G_1^{(1)}(z) &= G_1(z) = \frac{1}{\sqrt{2}}(1 - z^{-1}), \\ G_1^{(2)}(z) &= G_0(z) G_1(z^2) = \frac{1}{2}(1 + z^{-1} - z^{-2} - z^{-3}), \\ G_1^{(3)}(z) &= G_0(z) G_0(z^2) G_1(z^4) \\ &= \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^{-3} - z^{-4} - z^{-5} - z^{-6} - z^{-7}), \\ G_0^{(3)}(z) &= G_0(z) G_0(z^2) G_0(z^4) \\ &= \frac{1}{2\sqrt{2}}(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5} + z^{-6} + z^{-7}), \end{aligned}$$

preceded by upsampling by 2, 4, 8 and 8 respectively. The impulse responses follow by

<sup>9</sup>This is also sometimes called a discrete-time wavelet transform in the literature.

*Parseval's Equality* Orthogonality together with completeness (which follows from perfect reconstruction) leads to conservation of energy, also called Bessel's or Parseval's equality, that is

$$\|x[n]\|^2 = \sum_{k \in \mathbb{Z}} (|X^{(J)}[2k]|^2 + \sum_{j=1}^J |X^{(j)}[2k+1]|^2).$$

### 3.3.3 Multiresolution Interpretation of Octave-Band Filter Banks

The two-channel filter banks studied in Sections 3.1 and 3.2 have the property of splitting the signal into two lower-resolution versions. One was a lowpass or coarse resolution version, and the other was a highpass version of the input. Then, in this section, we have applied this decomposition recursively on the lowpass or coarse version. This leads to a hierarchy of resolutions, also called a *multiresolution decomposition*.

Actually, in computer vision as well as in image processing, looking at signals at various resolutions has been around for quite some time. In 1983, Burt and Adelson introduced the pyramid coding technique, that builds up a signal from its lower-resolution version plus a sequence of details (see also Section 3.5.2) [41]. In fact, one of the first links between wavelet theory and signal processing was Daubechies' [71] and Mallat's [180] recognition that the scheme of Burt and Adelson is closely related to wavelet theory and multiresolution analysis, and that filter banks or subband coding schemes can be used for the computation of wavelet decompositions. While these relations will be further explored in Chapter 4 for the continuous-time wavelet series, here we study the discrete-time wavelet series or its octave-band filter bank realization. This discrete-time multiresolution analysis was studied by Rioul [240].

Since this is a formalization of earlier concepts, we need some definitions. First we introduce the concept of embedded closed spaces. We will say that the space  $V_0$  is the space of all square-summable sequences, that is,

$$V_0 = l_2\{\mathcal{Z}\}. \quad (3.3.18)$$

Then, a multiresolution analysis consists of a sequence of embedded closed spaces

$$V_J \subset \cdots \subset V_2 \subset V_1 \subset V_0. \quad (3.3.19)$$

It is obvious that due to (3.3.18–3.3.19)

$$\bigcup_{j=0}^J V_j = V_0 = l_2\{\mathcal{Z}\}.$$

The orthogonal complement of  $V_{j+1}$  in  $V_j$  will be denoted by  $W_{j+1}$ , and thus

$$V_j = V_{j+1} \oplus W_{j+1}, \quad (3.3.20)$$