

Ondelettes régulières: application à la compression d'images fixes

Olivier Rioul

▶ **To cite this version:**

Olivier Rioul. Ondelettes régulières: application à la compression d'images fixes. domain_other. Télécom ParisTech, 1993. English. NNT : . pastel-00001307

HAL Id: pastel-00001307
https://pastel.hal.science/pastel-00001307

Submitted on 1 Jul 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

présentée pour obtenir le titre de

Docteur de l'École Nationale Supérieure des Télécommunications

Spécialité : Signal et Images

Olivier RIOUL

ONDELETTES RÉGULIÈRES : APPLICATION À LA COMPRESSION D'IMAGES FIXES

soutenue le 15 *Mars* 1993

devant le jury composé de :

Claude LABIT	*Président*
Yves MEYER	*Rapporteur*
Martin VETTERLI	*Rapporteur*
Michel BARLAUD	*Examinateur*
Yves GRENIER	*Examinateur*
Pierre DUHAMEL	*Directeur de thèse*

Table des matières

1 Introduction **1**
1.1 Les ondelettes en traitement du signal . . . 1
1.1.1 Une théorie à succès . . . 1
1.1.2 Différents types de transformées en ondelettes . . . 2
1.1.3 Bémols . . . 4
1.2 Cadre et organisation de l'étude . . . 5
1.2.1 Analyse du problème . . . 6
1.2.2 Outils . . . 6
1.2.3 Schéma de compression et résultats . . . 7
1.2.4 Restrictions . . . 7

I Ondelettes et bancs de filtres **11**

2 Ondelettes discrètes **13**
2.1 Fonctions de base discrètes . . . 15
2.1.1 Rappel sur les bancs de filtres . . . 15
2.1.2 Vocabulaire . . . 16
2.1.3 Bases discrètes bi-orthogonales . . . 17
2.1.4 Cas orthogonal . . . 18
2.2 Echelle et résolution . . . 19
2.2.1 Formalisme des opérateurs . . . 19
2.2.2 Echelle . . . 20
2.2.3 Résolution . . . 21
2.2.4 Encore la biorthogonalité . . . 22
2.3 Décompositions multi-résolution discrètes . . . 22
2.3.1 Résidus . . . 22
2.3.2 Résidus "critiques" . . . 23
2.3.3 Définition générale des systèmes multi-résolution . . . 23
2.3.4 Transformées pyramidales . . . 24
2.4 Transformées en ondelettes discrètes . . . 24
2.4.1 Cas bi-orthogonal . . . 26
2.4.2 Cas orthogonal . . . 27
2.5 Extensions possibles . . . 27
2.6 Conclusion . . . 27

3 Ondelettes continues **31**
3.1 Rappel sur les séries d'ondelettes . . . 31
3.2 L'algorithme de S. Mallat . . . 34
3.3 L'algorithme d'I. Daubechies . . . 35
3.4 Conclusion . . . 36

II Outils **41**

4 Régularité **43**
4.1 Préliminaires . . . 43
4.1.1 Cadre de l'étude . . . 43
4.1.2 Cas "pathologiques" instables . . . 44
4.1.3 Motivation . . . 45
4.1.4 Historique . . . 46
4.2 Analyse théorique . . . 47
4.2.1 Convergence uniforme et continuité . . . 47
4.2.2 Dérivées . . . 49
4.2.3 Zéros à la fréquence de Nyquist . . . 50
4.2.4 Moments nuls . . . 50
4.3 Estimations de régularité . . . 51
4.3.1 Régularité de Sobolev et de Hölder . . . 51
4.3.2 Caractérisation discrète de la régularité . . . 52
4.3.3 Vitesse de convergence . . . 53
4.3.4 Estimation optimale . . . 53
4.3.5 Une borne supérieure fine . . . 54
4.4 Cas particuliers et extensions . . . 56
4.4.1 Cas symétrique . . . 56
4.4.2 Régularité locale et "fractale" . . . 57
4.4.3 Autres extensions possibles . . . 60
4.5 Conclusion . . . 60

5 Calcul de bancs de filtres réguliers **63**
5.1 Filtres orthogonaux . . . 65
5.1.1 Méthode générale . . . 65
5.1.2 Approximation au sens de Tchebichev . . . 65
5.1.3 Les filtres de Daubechies . . . 66
5.1.4 Les filtres de Mintzer/Smith/Barnwell . . . 67
5.1.5 Résolution du programme de Tchebichev . . . 68
5.1.6 Résultats . . . 69
5.1.7 Cas de l'approximation quadratique . . . 72
5.2 Filtres à phase linéaire . . . 73
5.2.1 Généralités . . . 73
5.2.2 Détermination du complément bi-orthogonal . . . 74
5.2.3 Méthode choisie . . . 75
5.2.4 Résultats . . . 76
5.2.5 Extensions possibles . . . 76

5.3 Conclusion 76

6 Algorithmes rapides **81**
6.1 Généralités 81
6.1.1 Complexité arithmétique 81
6.1.2 La TOD standard est naturellement rapide 82
6.1.3 Calcul de la TOD inverse 83
6.2 Réorganisation des calculs 83
6.3 Algorithmes rapides basés sur la TFR 83
6.4 Algorithmes rapides pour les filtres courts 84
6.5 Cas particuliers et extensions 86
6.6 Conclusion 87

III Application à la compression d'images **89**

7 Schéma de compression d'images fixes **91**
7.1 Partie "transformée" 92
7.1.1 TOD bidimensionnelle séparable 92
7.1.2 Traitement aux bords 93
7.1.3 Rôle de la transformée 93
7.2 Quantification et codage 94
7.2.1 Quantification scalaire uniforme 95
7.2.2 Quantification vectorielle sur réseau 95
7.2.3 Justification du choix de la quantification "uniforme" 96
7.2.4 Codage entropique 97
7.3 Allocation optimale de bits 99
7.3.1 Préliminaires 99
7.3.2 Choix de l'algorithme 100
7.3.3 Description graphique 100
7.3.4 Nombre de décompositions optimal 102
7.4 Conclusion 103

8 Résultats **105**
8.1 Préliminaires 105
8.1.1 Filtres utilisés 105
8.1.2 Images testées 106
8.1.3 Visualisation des résultats 106
8.1.4 Mise en œuvre de l'allocation optimale de bits 106
8.1.5 Qualité visuelle et critère quadratique 107
8.2 Influence de la phase 107
8.3 Influence de la régularité/sélectivité en fréquence 110
8.3.1 Degré de platitude variable 110
8.3.2 Longueur variable 113
8.3.3 Bande de transition variable 115
8.4 Conclusion 115

9 Conclusion générale et perspectives **119**

IV Appendices **121**

A Articles **123**
A.1 Wavelets and Signal Processing 125
A.2 A Discrete-Time Multiresolution Theory 153
A.3 Simple Regularity Criteria for Subdivision Schemes 173
A.4 A Remez Exchange Algorithm for Orthonormal Wavelets 209
A.5 Fast Algorithms for Discrete and Continuous Wavelet Transforms 223
A.6 On the Choice of "Wavelet" Filters for Still Image Compression 243

B Bibliographie de l'auteur **249**
B.1 Articles "tutoriaux" 249
B.2 Articles publiés dans des revues à comité de lecture 249
B.3 Articles publiés dans des actes de congrès 250
B.4 Rapports techniques 251

C Calcul de filtres : boîtes à outils MATLAB **253**
C.1 DSP Toolbox 253
C.2 Wavelet Toolbox 256

Liste des Figures

2.1 Un banc de filtres itéré en octaves . . . 14
2.2 Décomposition fréquentielle en 5 octaves . . . 14
2.3 Un banc de filtres à deux sous-bandes et à reconstruction parfaite . . . 15
2.4 Un banc de filtres para-unitaire (cas orthogonal) . . . 18
2.5 Fonctions de base dans un banc de filtres itéré en octaves . . . 26

3.1 L'algorithme de S. Mallat . . . 34
3.2 Réponses impulsionnelles de filtres itérés . . . 37
3.3 Fonctions $\varphi(t)$ obtenues après convergence . . . 38

4.1 Exemple de schéma de subdivision divergent . . . 46
4.2 Divergence due à $G(-1) \neq 0$ . . . 48
4.3 Un exemple de fonction limite continue . . . 49
4.4 Implantation de l'algorithme 1 . . . 55
4.5 Bornes inférieures et supérieures de régularité . . . 58

5.1 Gabarit du filtre passe-bas $P(e^{j\omega})$ utilisé pour le programme de Tchebichev . . . 66
5.2 Caractéristiques d'amplitude des filtres de Daubechies et de Mintzer/Smith/Barnwell . . . 67
5.3 Un exemple de filtre régulier maximalement sélectif en fréquence . . . 70
5.4 Eventail de possibilités fourni par la méthode de calcul des filtres orthogonaux . . . 71
5.5 Réponse en fréquence d'un filtre régulier calculé par approximation "quadratique" 72
5.6 Exemple de solutions bi-orthogonales à phase linéaire . . . 77
5.7 Ondelettes (antisymétriques) correspondant à l'exemple de la figure 5.6 . . . 78

6.1 Cellule élémentaire de calcul dans une TOD . . . 82
6.2 Cellule modifiée . . . 84
6.3 Un exemple d'algorithme de filtrage RIF rapide . . . 85

7.1 Schéma général de compression d'images fixes . . . 92
7.2 Additivité de la distorsion . . . 99
7.3 Illustration graphique de l'algorithme d'allocation optimale de bits . . . 101

8.1 Influence de la phase (cas orthogonal) . . . 108
8.2 Influence de la phase (cas biorthogonal) . . . 109
8.3 Influence de la régularité (platitude) . . . 111
8.4 Influence du type de quantification . . . 112
8.5 Influence du critère d'optimisation des filtres . . . 113
8.6 Influence de la longueur . . . 114

8.7 Influence des largeurs de bandes de transition 116

Liste des Tables

3.1 Comparaison des ondelettes continues et discrètes 33

4.1 Estimations de régularité pour les ondelettes de Daubechies 56

5.1 Comparaison entre régularités obtenues par les deux méthodes de calcul de filtres 73

Whatever happens, looks as if it were intended.
— FIRST RULE OF ACTING (IN : THE COMPLETE MURPHY'S LAW, 1990)

Remerciements

If you do something which you are sure will meet everybody's approval, somebody won't like it.
— 2ND COROLLARY TO KRANSKE'S LAW
There are two types of people : those who divide people into two types, and those who don't.
— BARTH'S DISTINCTION
All the good ones are taken.
— HARRIS' LAMENT

Je tiens tout d'abord à remercier très vivement M. Claude Labit, directeur de recherche à l'IRISA de Rennes, et spécialiste du traitement d'images, de m'avoir fait l'honneur de bien vouloir présider le jury de cette thèse.

Je tiens également à exprimer ma profonde gratitude à M. Yves Meyer, professeur à l'Université Paris-Dauphine, dont les travaux sur les ondelettes ne sont plus à présenter. Il fut sans doute, par son enthousiasme, ses conseils et sa patience, le professeur qui m'a orienté vers le sujet des ondelettes, bien avant le commencement effectif de cette thèse. Il eût donc été anormal qu'il ne fît pas partie du jury. Je le remercie également d'avoir accepté d'être rapporteur, ce qui, grâce à ses qualités de mathématicien et son ouverture vers le traitement du signal, apporte une vision complémentaire du travail effectué.

Un grand merci à M. Martin Vetterli, professeur à l'Université de Berkeley, Californie, U.S.A. (*on leave from* Columbia University, New York), pour m'avoir fait l'amitié d'être rapporteur et de venir en France pour ma soutenance, ainsi que pour tous les bons moments passés ensemble, et la collaboration fructueuse avec lui-même et son équipe tout au long de ces trois dernières années. *Let me put it this way*, qu'il trouve ici l'expression de ma sincère gratitude.

Mes remerciements s'adressent également à M. Michel Barlaud, professeur à l'Université de Sophia-Antipolis, pour sa présence dans le jury, ainsi que pour de nombreux contacts échangés, notamment lors de son action entreprise au sein du GDR-134 "Traitement du Signal et des Images".

J'exprime ma reconnaissance à M. Yves Grenier, professeur à Télécom Paris, pour son travail en tant que correspondant scientifique de cette thèse, et pour l'apport d'un œil neuf vis-à-vis des travaux effectués lors de celle-ci.

Enfin, je ne saurais assez remercier M. Pierre Duhamel, pour l'estime et l'amitié qu'il a manifestées à mon égard, pour son soutien, son aide et sa participation directe tout au long de ce travail, mais aussi pour m'avoir appris les ficelles du métier et pour avoir rendu possible un premier développement de ma carrière scientifique. Qu'il soit ici assuré de mon entière reconnaissance.

Ce travail a été réalisé au Centre National d'Études des Télécommunications, au sein du Centre de Recherche en Physique de l'Environnement Terrestre et Planétaire, dans le

Département Electromagnétisme, Théorie des Communications et Propagation. Je tiens à remercier ici toutes les personnes que j'ai côtoyées quotidiennement au CNET, pour la confiance, la sympathie ou l'amitié qu'elles m'ont témoignée. Au nombre d'entre elles figurent notamment Jacob Bénesty, Thierry Blu, Katia Hilal, Lamia Karray, Sylvie Mayrargue, Mohsen Montazéri, Jean-Pierre Rossi, Bouchra SenHadji, Sara Tressens, et Gérard Yvon.

Mes pensées s'adressent également à mes parents, sans qui ni cette thèse ni l'auteur n'auraient vu le jour. Enfin, je voudrais exprimer, du fond du cœur, mon amour et ma reconnaissance à Nana, Lola, et Mouche, qui ont partagé avec moi les joies et les moments difficiles.

Résumé

Cette thèse, commencée fin 1989, est consacrée à l'étude de l'influence pratique de nouvelles propriétés mathématiques apportées par la théorie des *ondelettes* dans le domaine du codage en sous-bandes, en vue d'application à la compression d'images fixes.

On montre, par une analyse théorique de la décomposition multi-résolution de signaux à temps discret, que l'apport pratique essentiel de la théorie des ondelettes, dans ce cadre, est la propriété de *régularité* des bancs de filtres itérés en octaves utilisés pour le codage en sous-bandes. On développe ensuite un certain nombre d'outils, qui vont rendre possible une étude exhaustive du rôle de la régularité pour des applications en compression d'images :

Une étude mathématique détaillée de la notion de régularité est menée ; elle permet d'obtenir des algorithmes d'estimation optimale de régularité. Ensuite, on développe plusieurs méthodes de calcul de bancs de filtres, permettant de réaliser des bons compromis entre la régularité et les autres propriétés des filtres généralement considérées comme utiles pour le codage d'images (orthogonalité, phase linéaire, sélectivité en fréquence, etc.). On propose également des algorithmes rapides permettant, à peu de frais, de réduire notablement la charge de calcul nécessaire à la réalisation d'un banc de filtres itéré en octaves.

Finalement, on mène une étude expérimentale du rôle des propriétés des filtres calculés, pour un schéma simple de compression d'images fixes, où la transformée en ondelettes est séparable. Dans le cadre restreint choisi, on met en avant l'intérêt potentiel du critère de régularité par rapport à ceux de sélectivité en fréquence et de phase linéaire.

Summary

THIS thesis, started at the end of 1989, is devoted to the study of the effect of new mathematical properties brought by wavelet theory on subband coding, for still image compression problems.

Using a theoretical analysis of discrete-time multiresolution decomposition, it is shown that, within our framework, the practical novelty of wavelets reduces to the *regularity* property of octave-band filter banks used in subband coding schemes.

Then, several tools are derived, which make a systematic study of regularity possible, for applications such as image compression. A mathematical study of the regularity property is investigated, in which optimal regularity estimation algorithms are derived. Several filter bank design algorithms are then proposed. They allow one to balance regularity against other filter properties (orthogonality, linear phase response, frequency selectivity, etc.) which are generally believed to be relevant for image coding. Fast algorithms are also derived, for which substancial reduction of computational load required by octave-band filter banks is possible.

Finally, an experimental study of filter properties is carried out, for a simple still image compression scheme, in which the wavelet transform is separable. In this restricted framework, the potential interest of regularity is emphasized, in competition with other criteria such as frequency selectivity and linear phase.

Chapitre 1

Introduction

If all you have is a hammer, everything looks like a nail.
— BARUCH'S OBSERVATION
Don't force it ; get a larger hammer.
— ANTHONY'S LAW OF FORCE
Create a need and fill it.
— ELY'S KEY TO SUCCESS
Enough research will tend to support your theory.
— MURPHY'S LAW OF RESEARCH

DANS l'introduction d'une thèse consacrée à l'application des ondelettes en codage d'images, il me semble indispensable de présenter tout d'abord quelques réflexions sur l'intrégration et l'apport des ondelettes en traitement du signal, et de souligner la problématique que cette intrégration a posée. Cette problématique apparaît en particulier pour l'application de la théorie des ondelettes au codage d'images, où un lien très fort existe avec la technique classique de codage en sous-bandes utilisant des bancs de filtres. Il est donc important d'identifier précisément ce qu'ont apporté les ondelettes dans le codage en sous-bandes, tant au niveau théorique que pratique. C'est cette identification qui a conduit aux principaux résultats de cette thèse : il s'agira de pouvoir mesurer l'effet pratique de nouvelles propriétés mathématiques qu'apportent les ondelettes dans des schémas de codage.

Pour la plus grande part de l'introduction, on pourra se référer au travail de synthèse commun de l'auteur et de Martin Vetterli [1], exposé dans un article reproduit à l'appendice A 1. Ce «tutorial» fait le point sur l'utilisation des ondelettes en traitement du signal.

1.1 Les ondelettes en traitement du signal

1.1.1 Une théorie à succès

Les ondelettes, en tant que sujet de recherche en traitement du signal, ont connu un développement impressionnant ces trois dernières années, particulièrement outre-Atlantique et en France. Ces années correspondent, incidemment, à la période que j'ai consacrée à cette thèse. Au moment où celle-ci commença, les ondelettes constituaient encore un sujet marginal en traitement du signal, pour la communauté scientifique internationale : les «pionniers» (en traitement du signal) étaient relativement peu nombreux. Aujourd'hui, on ne compte plus le nombre

de chercheurs ayant publié dans ce domaine (une centaine d'articles publiés dans des revues françaises ou internationales, et plus encore d'articles de conférence), de thèses qui y sont consacrées, ou de livres parus ou en cours de parution (une bonne dizaine). Les revues internationales les plus importantes, comme les IEEE TRANSACTIONS, ont consacré plusieurs numéros spéciaux sur le sujet, et les conférences les plus suivies en traitement du signal (comme l'ICASSP) sont truffées de sessions «ondelettes». Incontestablement donc, les ondelettes constituent aujourd'hui un domaine à part entière du traitement du signal, débordant toujours d'activité.

Il est relativement facile d'analyser les raisons pour lesquelles les ondelettes ont connu un tel succès en traitement du signal : tout d'abord, il faut reconnaitre le rôle important qu'a joué l'action, l'enthousiasme, et l'ouverture pluri-disciplinaire des premiers chercheurs en France, au nombre desquels il faut citer le triumvirat «MGM» (Jean Morlet, Alex Grossmann, et Yves Meyer). Cette action s'est concrétisée aux U.S.A. par les succès théoriques majeurs d'Ingrid Daubechies et de Stéphane Mallat.

Ensuite, les ondelettes apportent un certain nombre d'outils très généraux qui trouvent un cadre parfaitement naturel en traitement du signal, et ce pour de nombreuses raisons : les techniques développées par la théorie des ondelettes sont essentiellement des *transformées*, qui, appliquées au signal, produisent une représentation à deux paramètres (temps et échelle) de celui-ci. La transformée en ondelettes trouve donc un cadre naturel dans l'analyse de signaux, où l'échelle permet de définir une nouvelle notion de caractéristique «fréquentielle» dépendante du temps. De plus, les transformées en ondelettes sont *inversibles* : la théorie des ondelettes s'applique donc naturellement à toutes les domaines du traitement du signal basés sur l'*analyse* et la *synthèse*, où un traitement est effectué dans l'espace transformé : la liste est longue. Citons l'identification, la détection, la synthèse directe, les techniques adaptatives, et, bien sûr, le codage. Enfin, la théorie des ondelettes prévoit plusieurs cadres d'utilisation, où le signal et les paramètres de l'analyse sont soit continus (analogiques), soit discrets (échantillonnés). Ceci permet de couvrir une grande partie des domaines étudiés en traitement du signal. L'idée qui fut longtemps pressentie ou admise est que, pour une application donnée, les ondelettes apportent systématiquement une *alternative* à l'ensemble des techniques précédemment développées en traitement du signal.

1.1.2 Différents types de transformées en ondelettes

Afin de comprendre ces derniers points, il est peut-être utile de rappeler brièvement les définitions, devenues classiques, des différents types de transformées en ondelettes. C'est également l'occasion de s'étendre sur le vocabulaire spécifique aux ondelettes, qui est un peu particulier pour le traitement du signal.

Le point commun entre ces transformées est l'utilisation de «fonctions de base» sur lesquelles on décompose le signal. Le terme générique «ondelettes» est d'ailleurs là pour nous rappeler leur existence : les ondelettes sont les formes d'onde (allures temporelles) des fonctions de base ; elles oscillent à l'intérieur une étendue temporelle limitée, ce sont des fonctions «passe-bande». Ce qui fait la spécificité des ondelettes, c'est la façon particulière dont ces fonctions de base sont construites.

Donnons tout de suite un exemple : dans sa version continue, la transformée en ondelettes a été introduite, sous sa forme actuelle, par Goupillaud, Grossmann et Morlet [2]. La *Transformée en Ondelettes Continue* (TOC) d'un signal analogique $x(t)$ est définie par :

$$\mathrm{TOC}(a,b) = |a|^{-1/2} \int_{\mathbf{R}} x(t)\psi^* \left(\frac{t-b}{a}\right) dt \tag{1.1}$$

où ψ est fonction «ondelette» donnée. Cette formule amène plusieurs remarques importantes :

- Le domaine transformé est un plan : il est décrit par deux paramètres variables, a et b. Le deuxième, b, fait intervenir une translation en temps, alors que a est associé à une dilatation ou à une contraction temporelle. Pour cette raison, a est appelé *paramètre d'échelle*. Cette transformée *temps-échelle* offre, pour l'analyse de signaux, une alternative à l'analyse temps-fréquence traditionnelle [1]. Il est à noter que la formule (1.1) était déjà utilisée, sous le nom de *fonction d'inter-ambiguïté large bande* [3], dans des problèmes tels que l'estimation de l'effet Doppler d'une cible dans des applications Radar (où typiquement le paramètre a est très proche de 1).

- Les fonctions de base «ondelettes» elles-même, dépendent des paramètres temps-échelle (a, b). Ce sont :
$$\psi_{a,b}(t) = |a|^{-1/2}\psi\left(\frac{t-b}{a}\right).$$
La caractéristique des ondelettes est qu'elles sont définies à partir d'une seule fonction «prototype» $\psi(t)$ localisée en temps et en fréquence, par translation temporelle et dilatation/contraction (changement d'échelle).

  En interprétant les ondelettes $\psi_{a,b}(t)$ comme des fenêtres (oscillantes) délimitant la portion du signal analysé autour de l'instant $t = b$ et à l'échelle a, on observe que la TOC utilise des fenêtres courtes et très oscillantes – correspondant à des fréquences élevées – et des fenêtres larges et peu oscillantes – correspondant à des fréquences basses. Ceci est exactement dans l'esprit de l'analyse fréquentielle à largeurs de bande relatives constantes, appelée aussi *analyse à Q-constant* [4], par opposition à la transformée de Fourier court terme [5], qui, elle, utilise une seule largeur de fenêtre.

  On peut donc ici interpréter la TOC comme un cas particulier de l'analyse à Q-constant où les fonctions de base se déduisent les unes des autres par simple dilatation/contraction.

- En termes des fonctions de base $\psi_{a,b}(t)$, la formule (1.1) est un *produit scalaire* : on intègre le signal contre le complexe conjugué des $\psi_{a,b}(t)$. Ceci est le produit scalaire naturel de l'espace $\mathbf{L}^2(\mathbf{R})$ des signaux d'énergie finie. Bien que ce ne soit pas à proprement parler l'apanage des ondelettes, on insiste ici sur le fait que les coefficients (1.1) de l'analyse sont obtenus par produit scalaire sur des fonctions de base.

Nous venons de parler de produits scalaires. L'analogie avec ceux utilisés en géométrie pour projeter orthogonalement un vecteur dans un sous-espace peut alors venir à l'esprit ; elle est ici bien justifiée. En effet, une constante dans la théorie des ondelettes est que le signal $x(t)$ apparaît comme une décomposition (sommation) de «projections» sur les ondelettes $\psi_{a,b}(t)$. Ces projections sont les ondelettes elles-même, pondérées par les produits scalaires correspondants (1.1). Pour la TOC, la sommation est continue :

$$x(t) = \iint \mathrm{TOC}(a,b)\,\psi_{a,b}(t)\frac{da\,db}{a^2} \tag{1.2}$$

C'est la formule de *synthèse*, qui permet de reconstruire le signal à partir des coefficients (1.1) fournis par l'analyse. (La validité de cette formule requiert quelques conditions sur $\psi(t)$ [2].)

Les autres types de transformées en ondelettes suivent exactement le même schéma général, et il est donc inutile de faire trop de remarques sur celles-ci. Les seules différences importantes concernent la nature, continue ou discrète, du signal ou des paramètres temps-échelle.

Les Séries d'Ondelettes (SO) considèrent des paramètres temps-échelle discrets, tout en conservant la nature analogique du signal. Plusieurs discrétisations sont possibles, mais celle qui est, de loin, la plus étudiée fournit une décomposition en *octaves*, ce qui correspond à $a = 2^j$, où j est entier. Le choix $b = k2^j$ (k entier) est alors naturel [1]. Les formules d'analyse et de synthèse deviennent respectivement

$$\mathrm{SO}(j,k) = \int x(t)\tilde{\psi}_k^{j\,*}(t)\,dt \tag{1.3}$$

$$x(t) = \sum_j \sum_k \mathrm{SO}(j,k)\psi_k^j(t) \tag{1.4}$$

où l'on a fait intervenir deux familles d'ondelettes : une pour l'analyse, $\tilde{\psi}_k^j(t) = \tilde{\psi}(2^{-j}t - k)$, et une autre pour la synthèse, $\psi_k^j(t) = \psi(2^{-j}t - k)$. Bien entendu, la validité de ces formules dépend de conditions particulières sur $\psi(t)$ et $\tilde{\psi}(t)$. J'ai choisi ici d'introduire deux ondelettes prototypes pour mentionner le cas général (cas «bi-orthogonal», traité dans [6]). Nous y reviendrons tout au long de ce manuscrit.

Le cas *orthogonal* correspond au choix $\psi(t) = \tilde{\psi}(t)$: on utilise les mêmes ondelettes à l'analyse et à la synthèse. Le terme «orthogonal» vient des conditions que doivent alors satisfaire les ondelettes : le produit scalaire d'une ondelette quelconque avec une autre est nul :

$$\int \psi(2^{-j}t - k)\psi^*(2^{-i}t - l)\,dt = \begin{cases} 1 & \text{lorsque } i = j \text{ et } k = l, \\ 0 & \text{sinon.} \end{cases} \tag{1.5}$$

Autrement dit, les ondelettes sont «orthogonales» entre elles, à la fois pour différents pas de translation et au travers de différentes échelles.

La théorie des ondelettes orthogonales est très complète. Elle a été développée principalement sous l'impulsion d'Yves Meyer [7], sous la forme de décomposition sur des espaces «multi-résolution». (Incidemment, un schéma similaire à (1.3), (1.4) a été proposé par Frazier et Jawerth [8].) Il est à noter que le lien entre ondelettes et *bancs de filtres numériques* a été montré pour la première fois dans ce cadre par Stéphane Mallat (*cf.* chapitre 3).

Enfin, la Transformée en Ondelettes Discrètes (TOD) concerne les signaux purement discrets, et c'est dans ce cadre que le lien avec les bancs de filtres est le plus fort : cette transformée fait l'objet des chapitres suivants[1].

1.1.3 Bémols

Dans la description sommaire ci-dessus, on a énuméré un certain nombre de techniques très similaires, qui, avant l'apparition des ondelettes, étaient connues indépendamment en traitement du signal, dans différents domaines. Ceci peut être vu à la fois comme un avantage et un inconvénient des ondelettes.

Un avantage, parce que les ondelettes apportent une vision synthétique en unifiant, par reformulation, de nombreuses techniques qui étaient jusqu'alors totalement indépendantes. De plus, dans ce cadre unifié, on utilise des concepts théoriques importants (fonctions de base, orthogonalité), qui, bien qu'étant utilisés épisodiquement auparavant, sont soulignés de façon systématique par la vision «ondelettes».

[1] Les notations adoptées ici (TOC,SO,TOD) sont utilisées par de nombreux auteurs. Elles sont d'ailleurs bien justifiées si on les met en parallèle avec les dénominations classiques des différents types de transformées de Fourier (Transformée de Fourier Continue, Séries de Fourier, et Transformée de Fourier Discrète). Néanmoins, certains auteurs préfèrent un vocabulaire légèrement différent...

Un inconvénient, parce que l'apport *pratique* des ondelettes dans les applications, par rapport à des techniques connues auparavant en traitement du signal, est loin d'être clair. Les liens, souvent très forts, entre ondelettes et techniques apparentées du traitement du signal furent même, dans un premier temps, assez obscurs dans l'esprit des chercheurs. Pour des raisons de langage essentiellement, il a parfois été difficile de reconnaître, derrière la belle théorie mathématique des ondelettes, l'application de principes déjà utilisés en traitement du signal, et d'identifier les principes nouveaux.

Au nombre des raisons du succès des ondelettes en traitement du signal, il serait donc juste d'ajouter un effet de mode certain, qui a parfois conduit les chercheurs, travaillant dans des domaines d'application potentielle des ondelettes, à considérer celles-ci comme un passage obligé, sans nécessairement tirer profit de leurs spécificités. Une autre tendance de l'effet de mode «ondelettes» a été d'utiliser ce nouveau jouet à tout ce qui tombe sous la main, y compris en vue d'applications pour lesquelles le bon sens nous dicterait plutôt qu'elles sont plus adaptées à l'utilisation de techniques traditionnelles (comme l'analyse de Fourier, par exemple). L'enthousiasme et l'excitation produits par ce sujet provoque même des affirmations excessives : il y a quelques mois, un article du *New York Times* [9] annonçait que l'utilisation de la transformée en ondelettes pour la compression d'images permettrait une réduction du taux de compression d'un facteur 40 par rapport à l'utilisation de transformées classiques basées sur une analyse de Fourier, comme la TCD (Transformée en Cosinus Discrète). Tous ces revers de médailles ont suscité des réactions négatives compréhensibles de la part de certains spécialistes, d'autant plus que, comme mentionné plus haut, l'apport pratique des ondelettes sur certaines techniques connues reste souvent à confirmer.

La vision synthétique des ondelettes a néanmoins permis un certain nombres de généralisations ou d'idées nouvelles et prometteuses. Citons, pour l'analyse de signaux, le travail d'unification des représentations bi-linéaires («énergétiques») temps-échelle et temps-fréquence (voir par exemple [10]). Pour le codage, les «paquets d'ondelettes», où la décomposition spectrale du signal est adaptative [11], et la vision «ondelettes» adaptée aux bancs de filtres à facteurs rationnels [12, 13], permettent une plus grande flexibilité dans la décomposition.

Mais ces extensions dépassent parfois largement le cadre strict des ondelettes ; et, puisque celui-ci a suscité un énorme intérêt, il serait souhaitable de pouvoir réellement confirmer ou infirmer l'apport théorique et pratique des ondelettes pour telle ou telle application.

1.2 Cadre et organisation de l'étude

Mesurer l'apport réel des ondelettes par rapport aux techniques traditionnelles apparentées, pour une application donnée, requiert une étude suffisemment exhaustive. L'objet de cette thèse est précisément d'effectuer cette étude, pour des applications de compression d'images fixes : dans ce cadre, le problème revient à déterminer si, oui ou non, les propriétés mathématiques supplémentaires qu'apportent les ondelettes permettent, au bout du compte, d'améliorer le taux de compression obtenu par les techniques similaires de *codage en sous-bandes* (utilisant des bancs de filtres). Ces dernières ont déjà été utilisées avec succès en compression de parole ou d'images [14]–[16].

Un certain nombre de schémas de compression d'images, basés sur une transformée en ondelettes, ont été proposés [17]–[21]. Dans ces travaux, la transformée est, la plupart du temps, présentée sous la forme de séries d'ondelettes, telles qu'elles sont décrites ci-dessus. D'autre part, il est connu, depuis les travaux de Stéphane Mallat [22], que l'implantation

de ces séries d'ondelettes requiert l'utilisation de bancs de filtres numériques, à deux sous-bandes et à reconstrution parfaite, itérés sur les branches passe-bas[2]. De tels bancs de filtres sont classiquement utilisés en codage en sous-bandes. Puisqu'ils génèrent une décomposition fréquentielle en *octaves*, je prendrai, par souci de simplification, l'habitude de les qualifier de *bancs de filtres itérés en octaves*. Le problème posé est donc clair : il s'agit d'identifier les aspects pratiques nouveaux des séries d'ondelettes par rapport aux bancs de filtres itérés en octaves, et de mesurer leurs effets dans des schémas pratiques de compression d'images.

1.2.1 Analyse du problème

Dans un premier temps, une analyse théorique s'impose. Il s'agit d'établir, avec précision, les liens existant entre bancs de filtres et ondelettes, et d'insister sur leurs différences[3]. C'est l'objet de la première partie, divisée en deux chapitres.

Le chapitre 2 commence par effectuer une démarche à contre-pied : il décrit une vision «ondelettes» directement sur les bancs de filtres à deux sous-bandes, puis généralise cette vision en développant une approche «multi-résolution» basée uniquement sur des notions d'échelle et de résolution en temps discret. Cette approche permet de définir et d'étudier les propriétés des *ondelettes discrètes* et de la TOD, sans référence à leurs homologues en temps continu. Dans cette approche, toutes les notions mentionnées au § 1.1.2 sont développées pour les bancs de filtres en octaves.

Au chapitre 3, on met en parallèle les résultats obtenus au chapitre 2 avec les propriétés classiques des séries d'ondelettes à temps continu. On met également en avant les liens très forts existant entre TOD et séries d'ondelettes, en précisant ce qui permet le passage de l'un à l'autre. Ceci nous permet d'identifier le seul apport pratique et nouveau des ondelettes dans ce cadre, la propriété de *régularité*.

1.2.2 Outils

Cette notion de régularité pose problème, car elle est associée indirectement au banc de filtres, par le biais d'une définition mathématique portant sur les ondelettes continues. Il est donc indispensable de développer un certain nombre d'*outils*, qui vont nous permettre notamment de mesurer, dans les schémas de codage, l'influence de la régularité. Ces outils font l'objet de trois chapitres regroupés dans la Partie II.

Tout d'abord, il est nécessaire de pouvoir quantifier la régularité de façon précise. Il est en effet illusoire de prétendre étudier le rôle de la régularité si on ne dispose pas d'estimations correctes. Le chapitre 4 résout ce problème par une approche originale, qui consiste à traduire la notion de régularité sur les réponses impulsionnelles des filtres à temps discret. Elle donne, pour un filtre donné, une estimation de régularité optimale (au sens des définitions mathématiques), qui est, de plus, utilisable en pratique.

Ensuite, se pose le problème du choix des filtres, pour intégrer non seulement la propriété de régularité, mais aussi d'autres propriétés jugées utiles, voire indispensables pour le codage (sélectivité en fréquence, phase linéaire...). Les exemples de filtres réguliers disponibles dans la littérature sont très académiques et d'un intérêt limité à cet égard. Le chapitre 5 propose donc de nouveaux algorithmes de calcul de filtres (réguliers ou non), basées sur des techniques

[2] Je suppose ici que le lecteur est familiarisé avec le vocabulaire des bancs de filtres. Celui-ci sera néanmoins rappelé ultérieurement, au chapitre 2.

[3] Aujourd'hui, ces liens sont probablement bien établis dans la littérature et dans l'esprit des chercheurs. Cependant, les travaux présentés ici datent de deux ou trois ans, époque à laquelle la situation était différente.

d'optimisation. Ces algorithmes permettent effectivement de disposer d'une batterie de filtres avec de nombreux compromis possibles, indispensables pour une étude précise du rôle de chaque propriété.

Enfin, le chapitre 6 propose un certain nombre d'algorithmes rapides pour l'implantation de bancs de filtres, supposément responsables de la majeure partie de la charge de calcul nécessaire à la réalisation d'un schéma de codage. Ce chapitre donne, dans un cas assez général, les outils permettant d'utiliser facilement des techniques traditionnelles d'accélération de calcul de filtrage, ce qui permet une réduction notable de la complexité arithmétique, surtout pour des grandes longueurs de filtres.

1.2.3 Schéma de compression et résultats

La troisième et dernière partie de cette thèse est consacré à l'étude expérimentale proprement dite. On commence, au chapitre 7, par fixer le cadre de cette étude, ce qui suppose le choix de techniques de quantification et de codage, éléments indispensables pour mesurer les effets du choix des filtres de la partie transformée. Optimiser particulièrement ces éléments tomberait en dehors du sujet de cette thèse. Cependant, ils ne doivent pas être conçus grossièrement, pour éviter des comparaisons trop naïves. La partie quantification choisie est soit scalaire, soit vectorielle sur réseau (*lattice VQ*) – cette dernière technique fut introduite en compression d'images, avec succès, par l'équipe de Michel Barlaud [17]. Les taux de compression sont évalués grâce à de mesures d'entropie, ou de taux obtenus après codage de Huffmann. Finalement, afin de produire des comparaisons équitables entre les différents filtres, on a mis en œuvre un algorithme d'allocation de bits optimale (au sens de la distortion quadratique et du taux de compression), basé sur la description faite par Martin Vetterli et K. Ramchandran [23] pour les paquets d'ondelettes. Les résultats, décris au chapitre 8, ont été obtenus pour ce schéma de compression, sur des images standard J-PEG, ainsi que sur l'inévitable «Lena».

1.2.4 Restrictions

Un certain nombre de restrictions se sont imposées d'elles-mêmes lors de cette thèse. Par manque de temps, bien sûr, mais aussi parce qu'il est préférable, pour que cette étude soit précise, que les outils portant sur l'estimation de régularité et sur le calcul de filtres, mentionnés ci-dessus, aient un caractère optimal :

D'une part, pour permettre de mesurer l'influence de la régularité sur les résultats de compression, on a besoin d'estimations de régularité optimale. Le cadre naturel de l'approche effectuée au chapitre 4 est celui des filtres à Réponse Impulsionnelle Finie (RIF) en une dimension. On se restreint donc dans la suite aux filtres RIF. Il est probablement très difficile d'étendre cette approche au cas des filtres à réponse impulsionnelle infinie. Par contre, celle-ci se généralise très bien à deux dimensions (filtres non-séparables) ou aux bancs de filtres à changements d'échantillonnage rationnels.

D'autre part, pour permettre une comparaison précise de différents filtres, on a besoin de disposer de familles de filtres pour lesquelles certains compromis (régularité/sélectivité, par exemple), sont optimaux. On s'est donc restreint au cas monodimensionel, où les méthodes de calcul de bancs de filtres à deux sous-bandes, tenant compte de contraintes de régularité, sont bien maitrisées (ce qui n'est pas le cas en non-séparable à deux dimensions [24]). On se placera donc dans le cas séparable (traitement ligne/colonne) pour effectuer la transformée en ondelettes d'une image.

Ceci justifie *a posteriori* que l'étude théorique menée dans cette thèse porte essentiellement sur les bancs de filtres RIF (associées à des ondelettes à support compact) en une dimension. Cependant, certains résultats présentés ici s'étendent aisément à des cas plus généraux, et il est donc possible que la démarche entreprise dans cette thèse, qui donne les outils de base, puisse être développée dans d'autres contextes.

Bibliographie

[1] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, vol. 8, no. 4, pp. 14–38, Oct. 1991.

[2] P. Goupillaud, A. Grossmann, and J. Morlet, "Cycle-octave and related transforms in seismic signal analysis," *Geoexploration*, vol. 23, pp. 85–102, 1984/85.

[3] L. Auslander and I. Gertner, "Wide-band ambiguity function and the $ax + b$ group," in *Signal Processing, Part I : Signal Processing Theory*, L. Auslander, T. Kailath, and S. Mitter, Eds., New York : Springer, 1990, pp. 1–12.

[4] J. E. Younberg and S. F. Boll, "Constant-Q signal analysis and synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Tulsa, OK, USA), 1978, pp. 375–378.

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

[6] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Comm. Pure Applied Math.* To appear.

[7] Y. Meyer, *Ondelettes et Opérateurs, tome I*, Paris : Herrmann, 1990.

[8] M. Frazier and B. Jawerth, "The φ-transform and decomposition of distributions," in *Proc. Conf. Fucntion Spaces and Appl.*, 1986.

[9] G. Kolata, "New technique stores images more efficiently," in *Science Times (The New York Times)*, 12 Novembre 1991, pp. C1, C12.

[10] O. Rioul and P. Flandrin, "Time-scale energy distributions—A new class extending wavelet transforms," *IEEE Trans. Signal Processing*, vol. 40, no. 7, pp. 1746–1757, July 1992.

[11] M. V. Wickerhauser, "Acoustic signal compression with wavelet packets," in *Wavelets : A Tutorial in Theory and Applications*, C. K. Chui, Ed., Academic Press, 1992, pp. 679–700.

[12] J. Kovačević and M. Vetterli, "Perfect reconstruction filter banks with rational sampling factors," *IEEE Trans. Signal Processing.* To appear.

[13] T. Blu, "Iterated filter banks with rational factors : Links with discrete wavelet transforms," *IEEE Trans. Signal Processing.* Special issue on wavelets, to appear.

[14] R. E. Crochiere, S. A. Weber, and J. L. Flanagan, "Digital coding of speech in subbands," *Bell Syst. Tech. J.*, vol. 55, pp. 1069–1085, Oct. 1976.

[15] D. Esteban and C. Galand, "Application of quadrature mirror filters to split-band voice coding schemes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Hartford, CT, USA), May 1977, pp. 191–195.

[16] M. Vetterli, J. Kovačević, and D. L. Gall, "Perfect reconstruction filter banks for HDTV representation and coding," *Image Communication*, vol. 2, no. 3, pp. 349–364, Oct. 1990.

[17] M. Antonini, M. Barlaud, and P. Mathieu, "Image coding using lattice vector quantization of wavelet coefficients," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Toronto, Canada), 1991, pp. 2273–2276.

[18] C. K. Cheong, K. Aizawa, T. Saito, and M. Hatori, "Subband image coding with biorthogonal wavelets," *IEICE Trans. Fondamentals*, vol. E75-A, no. 7, pp. 871–881, July 1992.

[19] P. L. Borgne, F. Sellan, and C. Dorize, "Applications of wavelet transform to image compression and texture analysis," in *Proc. Int. Colloqueum "Wavelets and Applications"*, (Toulouse, France), 1993. To appear.

[20] S. Mallat and S. Zhong, "Compact image coding from edges with wavelets," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Toronto, Canada), 1991, pp. 2745–2748.

[21] W. R. Zettler, J. Huffamnn, and D. C. P. Linden, "Application of compactly supported wavelets to image compression," in *SPIE Vol. 1244 Image Processing Algorithms and Techniques*, 1990, pp. 150–160.

[22] S. Mallat, "A theory for multiresolution signal decomposition : The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674–693, July 1989.

[23] K. Ramachandran and M. Vetterli, "Best wavelet packet bases in the rate-distortion sense," *IEEE Trans. Image Processing*, Apr. 1993.

[24] J. Kovačević and M. Vetterli, "Design of multidimensional non-separable regular filter banks and wavelets," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (San Francisco, CA), 1992, pp. IV–389–392.

Partie I

Ondelettes et bancs de filtres

Chapitre 2

Ondelettes discrètes

The means justify the means. The approach to a problem is more important than its solution.
— HALL'S LAW
There's always an easier way to do it.
— ILES' LAW
When looking directly at the easier way, especially for long periods, you will not see it.
— 1ST COROLLARY
Neither will Iles.
— 2ND COROLLARY

Ce chapitre identifie les caractéristiques communes des ondelettes et des bancs de filtres itérés en octaves. Un tel banc de filtres est décrit à la figure 2.1. L'approche choisie ici est de développer une vision «ondelettes» directement sur les bancs de filtres.

Il est à noter que l'approche traditionnelle [2] pour les bancs de filtres est fréquentielle : l'idée est que, si les filtres[1] $G(z)$ et $H(z)$ étaient des filtres demi-bande idéaux, on aurait une décomposition en deux sous-bandes sans recouvrement. Les opérations de sous-échantillonnage par deux dans la figure 2.1 sont alors justifiées par la condition de Nyquist, et permettent d'obtenir globalement le même taux d'échantillonnage dans les sous-bandes que pour le signal original («échantillonnage critique»). Par itérations successives sur la demi-bande passe-bas, effectuées par le banc de filtres de la figure 2.1, on obtient une décomposition en octaves représentée à la figure 2.2.

Par opposition à cette approche fréquentielle, on adopte dans ce chapitre une approche temporelle (en temps discret) : tout d'abord, en partant d'un banc de filtres à deux sous-bandes classiques, on définit les «fonctions de base» et les propriétés de bi-orthogonalité et d'orthogonalité (brièvement présentées au chapitre d'introduction). Ensuite, grâce à un formalisme d'opérateurs, on développe les notions d'analyse multi-échelle et multi-résolution en temps discret. La Transformée en Ondelettes Discrètes (TOD), ainsi que les décompositions pyramidales, se définissent alors aisément dans cette approche, sans aucune référence au temps continu. On débouche enfin naturellement sur une relecture «temporelle» des bancs de filtres en octaves.

Une grande partie du contenu de ce chapitre a été exposé dans l'article [1], reproduit à l'appendice A 2, auquel on se réfèrera.

[1] Dans toute la suite, on utilise systématiquement les lettres H pour désigner le passe-Haut, et G pour le passe-bas.

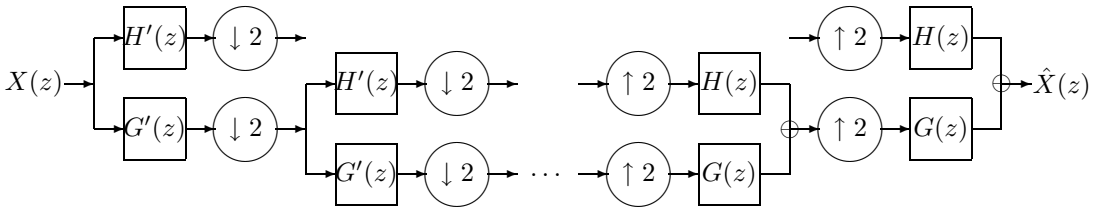


Figure 2.1: Un banc de filtres itéré en octaves. La partie «analyse» comporte les filtres passe-bas $G'(z)$ et passe-haut $H'(z)$. La partie «synthèse» (reconstruction) comporte les filtres passe-bas $G(z)$ et passe-haut $H(z)$. L'itération a donc lieu sur les branches passe-bas, où les mêmes filtres sont utilisés à chaque étape (on a représenté ici deux itérations). La condition de reconstruction parfaite impose que le signal de sortie $\hat{X}(z)$ soit identique à celui d'entrée $X(z)$, à un retard près.

Figure 2.2: Décomposition fréquentielle en 5 octaves. La sous-bande passe-bas est divisée en deux à chaque itération.

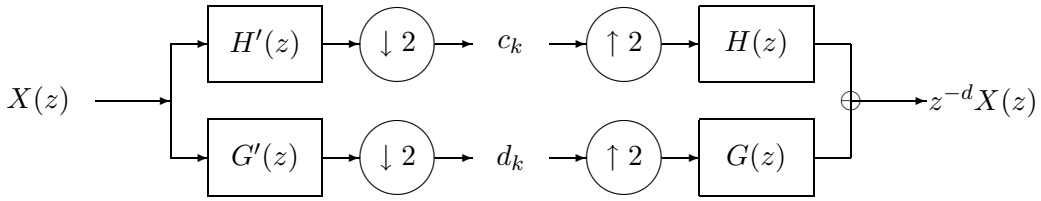


Figure 2.3: Un banc de filtres à deux sous-bandes et à reconstruction parfaite. On a supposé ici que les filtres sont causaux (*cf.* notations de la figure 2.1). Le retard après reconstruction est de d échantillons. Une fois qu'un tel banc de filtres a été construit, il suffit de l'itérer sur les sorties branches passe-bas (en tenant compte des retards accumulés) pour obtenir la figure 2.1, où la reconstruction parfaite est alors automatiquement garantie.

2.1 Fonctions de base discrètes

Commençons par une seule étape de décomposition du banc de filtres RIF, décrite à la figure 2.3. Contrairement à la figure 2.2, l'utilisation de filtres RIF, non-idéaux, induit un recouvrement spectral entre bandes, et la reconstruction parfaite requerra donc des conditions particulières sur les filtres pour que les recouvrements entre bandes s'annihilent durant la synthèse. Afin de fixer les idées, il est peut-être utile de rappeler tout d'abord la forme des contraintes classiques [3] que la condition de reconstruction parfaite impose sur les filtres.

2.1.1 Rappel sur les bancs de filtres

La figure 2.3 fait intervenir des successions d'opérations de sous-échantillonnage (ou *décimation*) et de sur-échantillonnage (aussi appelées *interpolation*). Clairement, la succession «décimation-interpolation» revient à prendre la partie paire de la transformée en z. La condition de reconstruction parfaite s'écrit donc :

$$\begin{aligned} z^{-d}X(z) &= \frac{1}{2}\left(X(z)G'(z)+X(-z)G'(-z)\right)G(z) \\ &\quad +\frac{1}{2}\left(X(z)H'(z)+X(-z)H'(-z)\right)H(z). \end{aligned}$$

En identifiant les termes en $X(z)$ (gain) et $X(-z)$ (recouvrement final), on obtient :

$$G(z)G'(z)+H(z)H'(z) = 2z^{-d} \quad (2.1)$$

$$G(z)G'(-z)+H(z)H'(-z) = 0. \quad (2.2)$$

Ces fonctions de transfert correspondent à des filtres RIF causaux, et peuvent donc être vues comme des polynômes en z^{-1}, décomposés en facteurs irréductibles. L'équation (2.1) nous permet de dire que, mis à part les termes monômes (retards) de la forme $c\,z^{-k}$, $G(z)$ et $H(z)$ n'ont aucun facteur commun, et de même pour $G'(z)$ et $H'(z)$. Considérant alors (2.2), on voit facilement, toujours en ignorant les termes retards, qu'il est nécessaire que tous les facteurs de $H(z)$ se retrouvent dans $G'(-z)$, et inversement. La même remarque est vraie pour $G(z)$ et $H'(-z)$. On peut donc écrire $H(z)=\alpha G'(-z)$ et $G(z)=\alpha H'(-z)$, où α et β sont de la forme $c\,z^{-k}$. En reportant dans (2.2), il vient $\alpha\beta=-1$, et sans nuire à la généralité du problème, on peut supposer $\alpha=1$ et $\beta=-1$ pour des filtres causaux (ce qui est l'approche traditionnelle) :

$$\begin{aligned} H(z) &= G'(-z) \\ H'(z) &= -G(-z). \end{aligned} \quad (2.3)$$

On a ainsi ramené le problème du calcul du banc de filtres à la détermination de deux filtres sur les quatre. Les relations (2.3) sont ici croisées : le filtre passe-haut d'analyse se déduit par modulation simple à partir du filtre passe-bas de synthèse, et le filtre passe-haut de synthèse se déduit par modulation simple à partir du filtre passe-bas d'analyse.

Il reste à déterminer les deux filtres passe-bas $G(z)$ et $G'(z)$. En reportant (2.3) dans (2.1), la condition de reconstruction parfaite se réduit à une condition simple sur la partie impaire de $P(z)$:

$$P(z) - P(-z) = 2z^{-d}, \tag{2.4}$$

où $P(z)$ est le filtre «produit» des passe bas :

$$P(z) = G(z)G'(z). \tag{2.5}$$

2.1.2 Vocabulaire

Afin d'identifier les «fonctions de base» intervenant dans le banc de filtres de la figure 2.3, et d'interpréter la reconstruction parfaite (2.4) en termes de ces fonctions de base, il est utile d'introduire quelques notations [1].

Ces notations sont «géométriques» : on considère le signal original $\{x_n\}$ (de transformée en z : $X(z)$) comme un «vecteur» x de coordonées x_n. Ce que nous allons appeler «fonctions de base» sont d'autres «vecteurs» f^k, indicés par un entier k : ces fonctions de base vont être définies à partir des réponses impulsionnelles des filtres. Elles peuvent être vues comme des signaux particuliers $\{f_n^k\}$. Dans la suite, nous indiquerons parfois n, comme un indice muet (représentant le temps). Ainsi, on notera indifféremment x, f^k, ou $\{x_n\}$, $\{f_n^k\}$.

En géométrie, on définit le produit scalaire $\langle u, v\rangle$ de deux vecteurs, comme la somme des produits des coordonnées. Ici, le produit scalaire du signal $\{x_n\}$ et d'*un* des vecteurs de base $\{f_n^k\}$ s'écrit

$$\langle x_n, f_n^k\rangle = \sum_n x_n (f_n^k)^*. \tag{2.6}$$

(On a introduit un complexe conjugué, pour le cas où le signal et/ou les filtres seraient complexes.) Par exemple, l'énergie du signal $\{x_n\}$ est $\langle x_n, x_n\rangle = \sum_n |x_n|^2$. L'orthogonalité est alors très simple à définir : des signaux sont *orthogonaux* si leurs produits scalaires, effectués deux à deux, s'annulent. Ils sont de plus *orthonormaux* si leurs énergies (ou leurs normes) valent 1.

Pour que cette définition de produit scalaire ait un sens, on demande que tous les signaux considérés soient d'énergie finie. La cadre mathématique naturel est l'espace $\ell^2(\mathbf{Z})$ des signaux d'énergie finie. Tous les «vecteurs» que nous considérons sont donc des éléments de $\ell^2(\mathbf{Z})$.

Le terme «fonction de base» pour les $\{f_n^k\}$ est justifié si l'on peut décomposer tout signal $\{x_n\}$ comme combinaison linéaire (infinie) des $\{f_n^k\}$:

$$x_n = \sum_k c_k f_n^k. \tag{2.7}$$

Les c_k sont les coefficients de la décomposition de $\{x_n\}$ sur la «base» des $\{f_n^k\}$. Nous ne donnerons pas plus de précisions pour l'instant : les propriétés mathématiques des fonctions de base seront claires sur les exemples ci-dessous.

2.1.3 Bases discrètes bi-orthogonales

Nous allons maintenant opérer une relecture du banc de filtres de la figure 2.3 à l'aide des notations que nous venons d'introduire.

Considérons tout d'abord les sorties du banc de filtre d'analyse. Elles sont calculées à partir du signal original $\{x_n\}$ par filtrage (convolution) et décimation. Ainsi, pour la partie passe-bas par exemple, où les coefficients du filtre sont donnés par $G'(z) = \sum_n g'_n z^{-n}$, on obtient les coefficients $c_k = \sum_n x_n g'_{2k-n}$ (indicés par k), ce que l'on peut réécrire facilement sous la forme d'un produit scalaire [1] :

$$c_k = \langle x_n, \tilde{g}'_{n-2k} \rangle \tag{2.8}$$

Il a fallu néanmoins opérer un «retournement» de g', indiqué par le symbole «˜». Ce retournement (avec passage au complexe conjugué) est connu dans la théorie des bancs de filtres comme une *para-conjugaison* [4] :

$$\tilde{g}'_n = (g'_{-n})^* \tag{2.9}$$

est le para-conjugué de $\{g'_n\}$. De même, les coefficients passe-haut dont donnés par $d_k = \langle x_n, \tilde{h}'_{n-2k} \rangle$ (*cf.* figure 2.3).

Avant d'aborder la partie reconstruction, il convient de faire une remarque importante concernant la *causalité* des filtres de la figure 2.3. En effet, celui-ci induit un retard total d. Mais, dans notre relecture, il est plus agréable d'imposer que le signal soit reconstruit sans terme de retard, ce qui impose de travailler avec des filtres non-causaux. Par rapport à la présentation donnée en § 2.1.1, il suffit, par exemple, d'opérer une translation de $(-d)$ coefficients sur les réponses impulsionnelles des filtres de synthèse. Nous supposerons que ce changement a été fait dans la suite de ce chapitre, lorsque l'on considère les fonctions de base. Ceci rendra les formules concernant ces fonctions de base plus agréables à manipuler. Bien sûr, ces formules restent valables pour le cas causal, pourvu que la translation adéquate sur les réponses impulsionnelles des filtres soit rajoutée.

Du côté de la synthèse, le signal $\{x_n\}$ est reconstruit comme somme de deux signaux calculés à partir des coefficients c_k et d_k fournis par l'analyse, par sur-échantillonnage et filtrage. On obtient [1] :

$$x_n = \sum_k c_k g_{n-2k} + d_k h_{n-2k} \tag{2.10}$$

Le signal est décomposé sur la base de fonctions $\{g_{n-2k}, h_{n-2k}\}$.

On utilise donc, en général, deux familles de fonctions de base : l'une pour la synthèse, qui est l'ensemble des signaux translatés $\{g_{n-2k}, h_{n-2k}\}$, et l'autre pour l'analyse, qui est l'ensemble $\{\tilde{g}'_{n-2k}, \tilde{h}'_{n-2k}\}$.

Nous arrivons maintenant à la partie intéressante : la condition de reconstruction parfaite (2.4), (2.5) peut se réécrire en fonction des $\{g_{n-2k}\}$ et des $\{\tilde{g}'_{n-2k}\}$ [1][2] :

$$\langle g_{n-2k}, \tilde{g}'_{n-2l} \rangle = \delta_{k,l} \tag{2.11}$$

C'est la propriété de *bi-orthogonalité*, qui a été introduite par Cohen, Daubechies et Feauveau [5] dans le cadre des séries d'ondelettes. Elle indique que les fonctions de base des deux familles

[2] Dans toute la suite, on utilisera le symbole de Kronecker

$$\delta_{k,l} = \begin{cases} 1 & \text{si } k = l, \\ 0 & \text{sinon.} \end{cases}$$

On utilisera également le signal «impulsion» δ_n, dont les échantillons sont $\delta_0 = 1$ et $\delta_n = 0$ pour $n \neq 0$.

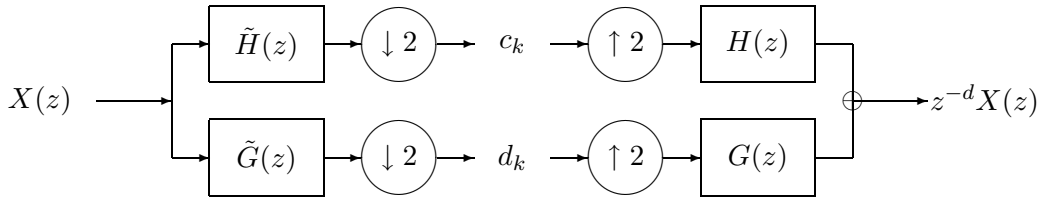


Figure 2.4: Un banc de filtres para-unitaire (cas orthogonal). Les filtres d'analyse sont les para-conjugués (2.14) de ceux de synthèse.

d'analyse, $\{\tilde{g}'_{n-2k}\}$, et de synthèse, $\{g_{n-2l}\}$, sont mutuellement orthogonales, sauf lorsque leurs indices coïncident.

Il est facile de vérifier [1], grâce aux relations (2.3), que la bi-orthogonalité est vérifiée, plus généralement, lorsque l'on met ensemble des fonctions de base passe-bas et passe-haut, c'est à dire pour les deux familles $\{g_{n-2k}, h_{n-2k}\}$ et $\{\tilde{g}'_{n-2k}, \tilde{h}'_{n-2k}\}$.

Nous avons donc, jusqu'ici, réécrit une étape de décomposition en bancs de filtres à deux sous-bandes et à reconstruction parfaite (figure 2.3) comme une décomposition du signal sur les fonctions de base bi-orthogonales $\{\tilde{g}'_{n-2k}, \tilde{h}'_{n-2k}\}$ (à l'analyse) et $\{g_{n-2k}, h_{n-2k}\}$ (à la synthèse). De plus, il est facile de montrer [1] que, réciproquement, cette bi-orthogonalité implique la reconstruction parfaite. C'est pourquoi on considérera les termes «reconstruction parfaite» et «bi-orthogonalité» comme synonymes dans la suite : le cas bi-orthogonal est finalement le cas le plus général des bancs de filtres à reconstruction parfaite.

2.1.4 Cas orthogonal

Le cas orthogonal est un cas particulier du cas bi-orthogonal, où l'on impose la contrainte supplémentaire que les filtres d'analyse sont les mêmes (à un retournement – para-conjugaison – près) que ceux de synthèse :

$$g'_n = \tilde{g}_n \qquad \text{et} \qquad h'_n = \tilde{h}_n \tag{2.12}$$

Autrement dit, les fonctions de base d'analyse et de synthèse coïncident.

Commençons tout d'abord par poursuivre la présentation du § 2.1.1 dans ce cas, en revenant donc (temporairement) à des filtres causaux : en termes de transformées en z, (2.12) s'écrit :

$$G'(z) = \tilde{G}(z) \text{ et } H'(z) = \tilde{H}(z). \tag{2.13}$$

(*cf.* figure 2.4). Ici, le «retournement» est modifié pour conserver la causalité, c'est à dire qu'on définira, pour tout $A(z)$:

$$\tilde{A}(z) = z^{-(L_A-1)} A_*(z^{-1}), \tag{2.14}$$

où L_A est la longueur du filtre $A(z)$ et où $A_*(z)$ est la transformée en z associée aux complexes conjugués des coefficients du filtre.

Il est facile de voir, grâce aux relations (2.3), que tous les filtres doivent avoir la même longueur L. Puisque le retard z^{-d} dans (2.4) est le terme milieu d'un polynôme impair en z^{-1} de degré $2L-2$, le retard total impair d est égal à $L-1$. Ainsi, la longueur des filtres est *paire*.

Grâce à l'identification (2.13), il ne reste plus qu'un seul filtre, disons $G(z)$, à déterminer. La relation entre le filtre passe-haut et passe-bas se réduit à $H(z) = \tilde{G}(-z)$, c'est à dire (pour des filtres causaux) :

$$h_n = (-1)^n g_{L-1-n}$$

Quant à la condition de reconstruction parfaite (2.4), elle se réduit dans ce cas à

$$P(z) = z^{-(L-1)} + Q(z^2) \tag{2.15}$$

où $Q(z)$ est un filtre de longueur L dont les coefficients sont symétriques. Historiquement, ce cas a été le premier pour lequel on a trouvé des solutions permettant la reconstruction parfaite dans un banc de filtres à deux sous-bandes, grâce aux travaux de Mintzer [6] et Smith et Barnwell [7]. La contrainte (2.15), exprimée sur $G(z)$ et $H(z)$, portait alors le nom de «CQF» (*Conjugate Quadrature mirror Filters*). Les travaux de Vaidyanathan [4] ont par la suite identifié, dans ce cas, la propriété de *para-unitarité* sur des matrices de transfert des bancs de filtres, aussi ce vocabulaire est-il devenu classique. Il provient essentiellement de la relation de para-conjugaison (2.12), est a également été perçu par plusieurs auteurs (notament Ingrid Daubechies [8]) travaillant sur la construction de séries d'ondelettes orthogonales à partir des bancs de filtres (dont nous parlerons au chapitre 3).

Passons maintenant au point de vue des fonctions de base. La condition (2.12), reportée dans (2.11), donne immédiatement la condition d'*orthonormalité* des fonctions de base $\{g_{n-2k}\}$:

$$\langle g_{n-2k}, g_{n-2l} \rangle = \delta_{k,l} \tag{2.16}$$

et, plus généralement, l'orthogonalité des fonctions de base $\{g_{n-2k}, h_{n-2k}\}$, c'est à dire (2.16) et

$$\begin{aligned} \langle h_{n-2k}, h_{n-2l} \rangle &= \delta_{k,l} \\ \langle g_{n-2k}, h_{n-2l} \rangle &= 0 \end{aligned}$$

Nous avons donc affaire ici à une *base orthonormale* pour les signaux d'énergie finie : les coefficients de l'analyse sont les produits scalaires du signal avec les fonctions de base,

$$\begin{aligned} c_k &= \langle x_n, g_{n-2k} \rangle \\ d_k &= \langle x_n, h_{n-2k} \rangle \end{aligned} \tag{2.17}$$

et la reconstruction est faite en sommant les projections sur ces fonctions de base :

$$x_n = \sum_k c_k g_{n-2k} + d_k h_{n-2k}. \tag{2.18}$$

En conclusion, le cas «CQF» des bancs de filtres, où l'on impose que les filtres d'analyse et de synthèse soient les mêmes (à un retournement près), s'identifie donc parfaitement au cas orthogonal.

2.2 Echelle et résolution

2.2.1 Formalisme des opérateurs

Jusqu'ici, nous n'avons considéré qu'une étape de décomposition, sans itérer le banc de filtres comme dans la figure 2.1. Afin de passer à plusieurs étapes de décomposition, et de définir les notions d'échelle et de résolution, il est commode d'introduire une nouvelle notation, celle d'opérateurs.

Ces notations suivent en fait l'écriture des graphes de fluence des figures 2.1, 2.3 et 2.4 : l'opérateur de filtrage RIF par $G(z)$ est noté $\mathbf{G}$. On peut le voir comme associée à une matrice

de filtrage infinie. On définit de même $\mathbf{H}$, $\mathbf{G}'$ et $\mathbf{H}'$ pour $H(z)$, $G'(z)$ et $H'(z)$. Les opérateurs de décimation et de sur-échantillonnage d'un facteur deux intervenant dans les figures sont notées $\downarrow$ et $\uparrow$, respectivement. Tous ces opérateurs sont linéaires et bornés dans $\ell^2(\mathbf{Z})$.

Ainsi, la branche passe-bas du banc de filtres de la figure 2.3 correspond à l'opérateur $\mathbf{G}\uparrow\downarrow\mathbf{G}'x$ appliqué au signal x. Et la reconstruction parfaite peut s'écrire $\mathbf{G}\uparrow\downarrow\mathbf{G}' + \mathbf{H}\uparrow\downarrow\mathbf{H}' = \mathbf{Id}$ (opérateur identité). Les opérateurs se calquent tout simplement sur les graphes de fluence, il suffit d'inverser l'ordre de lecture : l'ordre successif des opérations dans un graphe de fluence se lit de gauche à droite, alors que celui les opérateurs se lit conventionnellement de droite à gauche. Ces opérateurs seront donc très commodes (voire indispensables) pour décrire des bancs de filtres itérés sans refaire constamment des figures. C'est par ailleurs une notation classique qui a été utilisée par plusieurs auteurs, notament Ingrid Daubechies [8] et Mark Shensa [9].

2.2.2 Echelle

A l'aide des notations que nous venons d'introduire, nous allons maintenant définir des notions (différentes) d'*échelle* et de *résolution* en temps-discret.

La notion d'échelle s'inspire ici de celle des cartes routières [1]. C'est une notion relative, pour laquelle nous dirons, par convention, que le signal original x_n est à l'échelle 1. L'échelle se définit donc par la donnée d'opérateurs de changement d'échelle : une version d'un signal donné, «augmentée d'échelle», est un signal similaire, mais échantillonné à un taux plus élevé. De même, diminuer l'échelle revient à diminuer le taux d'échantillonnage en conservant une forme d'onde similaire.

On peut utiliser comme facteur d'échelle un entier quelconque (et même un rationnel [10]), mais dans le cadre qui nous intéresse ici, n'intervient que le facteur 2. On définit donc un opérateur de «doublement d'échelle» et un opérateur de «réduction d'échelle par deux» sous les hypothèses naturelles suivantes :

- Ce sont des opérateurs linéaires.
- Ils sont «cohérents» avec les translations en temps [1] : le pas d'une translation appliquée au signal est doublé par un doublement d'échelle, et, s'il est pair, divisé par deux par une réduction d'échelle par deux.

Ces conditions suffisent [1] pour identifier la forme générale de ces opérateurs :

$$\text{Doublement d'échelle} \;:\; \text{Opérateur } \mathbf{G}\uparrow \tag{2.19}$$

$$\text{Réduction d'échelle par 2} \;:\; \text{Opérateur } \downarrow\mathbf{G}' \tag{2.20}$$

où $\mathbf{G}$ et $\mathbf{G}'$ correspondent à des filtres quelconques. Ces opérateurs sont précisément les «briques de base» dont sont composés les bancs de filtres (*cf.* figure 2.3). On prendra toujours les mêmes filtres pour augmenter ou diminuer l'échelle d'un cran : cela signifie qu'aucune échelle n'est privilégiée et qu'elle est bien une notion relative.

Afin que les versions mises à l'échelle d'un signal aient des formes d'onde similaires, il est nécessaire d'imposer des conditions sur les filtres représentés par $\mathbf{G}$ et $\mathbf{G}'$. Intuitivement, il est clair que ces filtres seront plutôt passe-bas que passe-haut [1]. Obtenir des conditions précises est plus difficile : une façon de le faire est de considérer la propriété de régularité, traitée au chapitre 4.

Nous noterons $s(y)$ la valeur de l'échelle[3] d'un signal y. En partant du signal original x, d'échelle $s(x) = 1$, on peut appliquer successivement les opérateurs (2.19), (2.20) pour obtenir une version de x à une échelle quelconque (dont la valeur est une puissance entière de 2). Par exemple,

$$y = (\mathbf{G}\uparrow)^5(\downarrow\mathbf{G}')^3(\mathbf{G}\uparrow)^7(\downarrow\mathbf{G}')^{12}x$$

est une version de x à l'échelle $2^{-12+7-3+5} = \frac{1}{8}$. Il existe, bien sûr, de nombreuses façon d'aboutir à l'échelle $1/8^{\text{ième}}$, sans pour autant obtenir le même signal y [1].

On a donc besoin d'un autre paramètre pour caractériser de façon unique deux versions différentes du même signal x à la même échelle. On va maintenant l'introduire : il s'agit de la *résolution.*

2.2.3 Résolution

Intuitivement, la résolution dépend de la quantité d'information présente dans le signal : plus celle-ci est importante, plus le signal est à haute résolution. Nous dirons par convention que le signal original x est à résolution 1. C'est la résolution maximale : il est impossible, à partir de x, d'obtenir une plus grande résolution sans rajouter d'information.

La valeur $r(y)$ de la résolution du signal y est définie par son comportement vis-à-vis de l'action des opérateurs de changement d'échelle [1] : une réduction d'échelle par deux à partir du signal original x, fait intervenir une décimation d'un facteur 2, où la moitié des échantillons sont perdus. On dira alors que la résolution a été divisée par deux : $r(\downarrow\mathbf{G}'x) = r(x)/2$.

Par contre, si on augmente l'échelle, la résolution n'est pas affectée : ceci se justifie par le fait que le doublement d'échelle est injectif, c'est à dire que si deux versions doublées d'échelle sont égales, alors ils proviennent du même signal. Autrement dit, deux signaux distincts, une fois doublés d'échelle, donnent deux signaux nécessairement distincts. Ceci est toujours vérifié, sauf cas pathologique [1].

Maintenant, supposons qu'on réduise de moitié l'échelle d'un signal, qui lui-même provient déjà d'un doublement d'échelle. Ce signal était déjà à une échelle plus élevée que sa résolution. Dans ce cas, réduire l'échelle de moitié ne doit donc pas affecter cette résolution.

En résumé, la résolution d'une version y du signal original x à différentes échelles est $r(y) = 2^{-j}$, s'il est caractérisé par un échantillon sur 2^j périodes d'échantillonnage de x. D'après ce qui précède, les règles de calcul de la résolution d'un signal sont :

$$r(\mathbf{G}\uparrow y) = r(y) \tag{2.21}$$

$$r(\downarrow\mathbf{G}'y) = \begin{cases} r(y)/2 & \text{si } r(y) = s(y), \\ r(y) & \text{si } r(y) < s(y). \end{cases} \tag{2.22}$$

et on a toujours :

$$r(y) \leq s(y), \tag{2.23}$$

sinon, y serait caractérisé par plus d'échantillons que ceux véritablement présents dans le signal.

[3] Il serait juste de noter qu'incidemment, la convention adoptée ici pour définir l'échelle sur le signal est exactement inverse de celle utilisée habituellement par les théoriciens des ondelettes. Cette dernière définit l'échelle par rapport aux fonctions de base analysant le signal. La convention adoptée ici semble plus naturelle pour les décompositions multi-résolution d'images (voir plus loin).

2.2.4 Encore la biorthogonalité

Certes, ces définitions d'échelle et de résolution, bien que naturelles, sont un peu rébarbatives. Mais voici maintenant la récompense :

Nous avons dit que la notion de résolution allait permettre de caractériser de façon unique les versions du signal original à différentes échelles. Cela est vrai, sous une condition supplémentaire : il est montré dans [1] qu'étant données une échelle s et une résolution $r \leq s$, *il existe une et une seule version y du signal x à cette échelle s et cette résolution r*, si et seulement si

$$\downarrow \mathbf{G}'\mathbf{G}\uparrow = \mathbf{Id}. \tag{2.24}$$

Ceci semble un peu abstrait, mais, s'il l'on y regarde d'un peu plus près, on s'aperçoit [1] que c'est exactement la condition de *bi-orthogonalité* (2.11) qui nous revient ici[4]. La forme générale du signal x mis à l'échelle 2^{-i} et à la résolution 2^{-j} $(j \geq i)$, est alors $(\mathbf{G}\uparrow)^{j-i}(\downarrow\mathbf{G}')^j x$.

Alors que la bi-orthogonalité découlait en § 2.1.3 de la condition de reconstruction parfaite de bancs de filtres, elle apparaît ici comme une condition indispensable pour avoir les systèmes multi-résolution «cohérents», pour lesquels les approximations multi-résolution sont uniques.

2.3 Décompositions multi-résolution discrètes

En se fondant sur les notions d'échelle et de résolution que nous venons de décrire, il est possible de définir un système général de décomposition multi-résolution pour les signaux à temps-discret, dont les deux seuls cas intéressants correspondent, d'une part, aux décompositions pyramidales [12], et d'autre part, aux décompositions en ondelettes.

2.3.1 Résidus

D'après les règles de changement de résolution (2.21) et (2.22), on s'aperçoit [1] qu'une version du signal original x, à une résolution donnée $r \leq 1$, contient toute l'information nécessaire pour obtenir n'importe quelle autre version de x à une résolution inférieure $r' < r$, et ceci par seule application des opérateurs de changement d'échelle (2.19) et (2.20). Bien qu'il soit donc toujours possible de parcourir tout l'éventail multi-résolution désiré 2^{-j}, $j \in \mathbf{Z}$, il est important d'éliminer la redondance d'information sous-jacente.

La solution à ce problème est simple, mais fondamentale : elle est l'origine de tout système de décomposition multi-résolution. Elle se retrouve dans les techniques ondelettes, bien sûr – notamment dans la théorie des espaces multirésolution d'Yves Meyer [11] –, mais aussi dans les décompositions pyramides introduites par Burt et Adelson [12].

Il suffit de décomposer le signal en des signaux *résidus*, qui contiennent l'information nécessaire pour passer d'une résolution donnée à une autre plus grande. On définit donc le signal résidu du signal original x, à l'échelle s et à la résolution r, comme le signal qui, ajouté à la version de x à la même échelle et résolution, *augmente* la résolution d'un facteur 2. En notant $x_{s,r}$ la version de x à l'échelle s et à la résolution r, le résidu $d_{s,r}$ est ainsi défini par la différence :

$$d_{s,r} = x_{s,2r} - x_{s,r}. \tag{2.25}$$

[4]Pour être complet, signalons qu'une autre condition équivalente [1] est que l'opérateur correpondant à la branche passe bas de la figure 2.3, $\mathbf{A} = \mathbf{G}\uparrow\downarrow\mathbf{G}'$, qui approxime le signal original à la moitié de sa résolution sans changer l'échelle, est une *projection* ($\mathbf{A}^2 = \mathbf{A}$) : ré-approximer par $\mathbf{A}$ ne change rien.

Il est important de noter que cette définition n'est valide que si $s \geq 2r$, pour que $x_{s,2r}$ soit bien défini (*cf.* (2.23)). Par exemple, le résidu à l'échelle 1/2 et à la résolution 1/4 est égal à $\downarrow\mathbf{G}'x - \mathbf{G}\uparrow(\downarrow\mathbf{G}')^2x$.

Il est facile de voir que les résidus $d_{s,r}$ subissent les mêmes règles de changement d'échelle et de résolution par action des opérateurs (2.19) et (2.20) que les versions $x_{s,r}$ du signal x. La seule exception est que si $s = 2r$, alors $\downarrow\mathbf{G}'d_{s,r} = 0$. La dernière remarque montre que l'on ne peut pas, à l'aide des seuls opérateurs de changement d'échelle (2.19) et (2.20), définir des résidus à une même valeur d'échelle et de résolution. Ceci serait pourtant naturel : on conçoit difficilement qu'un signal à une résolution r ne puisse pas posséder un «représentant critique» à l'échelle $s = r$.

2.3.2 Résidus "critiques"

Comment peut-on faire pour définir quand même un résidu à la résolution r et à l'échelle $s = r$? C'est ici que les bancs de filtres à reconstruction parfaite, décrits à la figure 2.3, viennent à notre secours[5]. Identifions les opérateurs $\mathbf{G}$ et $\mathbf{G}'$ aux filtres passe-bas $G(z)$ et $G'(z)$, et faisons intervenir de nouveaux opérateurs, $\mathbf{H}$ et $\mathbf{H}'$, correspondant cette fois-ci aux filtres passe-haut $H(z)$ et $H'(z)$. Tout ceci est cohérent avec l'approche effectuée pour définir les notions d'échelle et de résolution, puisque la condition de base (2.24) s'identifie à une condition de bi-orthogonalité qui est bien, comme on l'a vu, imposée par la reconstruction parfaite. Il est facile de voir que celle-ci s'écrit, en termes d'opérateurs, $\mathbf{G}\uparrow\downarrow\mathbf{G}' + \mathbf{H}\uparrow\downarrow\mathbf{H}' = \mathbf{Id}$. D'après cette condition, il vient $d_{2s,r=s} = \mathbf{H}\uparrow\downarrow\mathbf{H}'x_{2s,r=2s}$.

On peut donc définir [1] le résidu à l'échelle s et à la résolution $r = s$ comme $d_{s,r=s} = \downarrow\mathbf{H}'x_{2s,r=2s}$: c'est effectivement un signal à l'échelle s, et il contient toute l'information nécessaire pour obtenir un résidu à la résolution $r = s$. Il est ramené à une échelle double de la résolution par action de l'opérateur $\mathbf{H}\uparrow$.

Par exemple, le résidu à l'échelle 1/2 et résolution 1/2 est tout simplement $\downarrow\mathbf{H}'x$. En appliquant $\mathbf{H}\uparrow$, on obtient le résidu à l'échelle 1 et résolution 1/2, donné par $x - \mathbf{G}\uparrow\downarrow\mathbf{G}'x$.

En conclusion, la définition des résidus «critiques», à échelle et résolution égales, se base essentiellement sur l'existence de bancs de filtres à reconstruction parfaite, et requiert donc des contraintes additionnelles sur les filtres utilisés, ce qui n'est pas le cas des résidus non «critiques».

2.3.3 Définition générale des systèmes multi-résolution

Armé des notions précédentes, il est maintenant très facile de donner une définition générale de décomposition multi-résolution en temps-discret. Cette décomposition consiste à itérer (2.25), de façon à obtenir le signal original comme une somme où les résidus $d_{s,r}$ à tous les niveaux de résolution $r = 2^{-j}$, $j = 1, \ldots, J$ apparaissent.

$$x = \sum_{j=1}^{J} d_{s=1,r=2^{-j}} + x_{s=1,r=2^{-J}} \tag{2.26}$$

Dans cette formule de décomposition multi-résolution, $x_{s=1,r=2^{-J}}$ est une version de x à très basse résolution, auquel on ajoute successivement des résidus, pour augmenter cette résolution. On s'arrête dès que la résolution maximale $r = 1$ est atteinte, ce qui correspond au signal x reconstruit. On vient de décrire la partie reconstruction du système multirésolution.

[5] On pourrait ne pas faire explicitement appel à eux, mais cela reviendrait conceptuellement au même.

La partie analyse multirésolution consiste à calculer, à partir du signal initial x, tous les résidus aux différentes résolutions, ainsi qu'une version correspondant à la résolution la plus basse.

Dans la formule (2.26), j'ai supposé tous les signaux à l'échelle $s = 1$ afin qu'ils puissent être directement sommés. Mais, bien entendu, pour être économique du point de vue du volume de données calculé, la partie analyse devrait uniquement calculer les résidus à l'échelle la plus petite possible par rapport à leurs niveaux de résolution. Ainsi, plutôt que de calculer $d_{s=1,r=1/8}$ à l'analyse, on préfèrera calculer $d_{s=1/4,r=1/8}$, ou mieux, le résidu critique $d_{s=1/8,r=1/8}$. Ce résidu sera ensuite remis à l'échelle 1 pendant la synthèse, suivant les règles décrites plus haut. (Cette remarque est valable, bien sûr, pour le calcul de la version très basse résolution de x.)

En conclusion, ce qui distingue les systèmes de décomposition multi-résolution entre eux, c'est l'échelle à laquelle les résidus et la version basse-résolution du signal sont calculés à l'analyse. La partie synthèse remet tous ces signaux à l'échelle 1, en même temps qu'elle somme les résultats pour reconstruire le signal initial x (2.26). Dans le langage des transformées, la transformée elle-même est la partie «analyse», alors que la transformée inverse effectue la synthèse (reconstruction).

2.3.4 Transformées pyramidales

La manière la plus économique de calculer les résidus, sans faire appel aux résidus «critiques», est de définir la transformée comme l'opération calculant les résidus à l'échelle 2^{-j+1} et à la résolution 2^{-j}, pour $j = 1, \ldots, J$, et le signal basse-résolution à l'échelle 2^{-J} et à la résolution 2^{-J}. L'article [1] montre que cette simple description suffit pour décrire entièrement les *transformées pyramidales*, telles qu'elles furent proposées par Burt et Adelson [12]. Puisque les résidus sont calculés à une échelle double de leurs résolutions, ces transformées calculent plus d'échantillons que ceux présents dans le signal initial [1] (échantillonnage sur-critique).

2.4 Transformées en ondelettes discrètes

Pour obtenir le même taux d'échantillonnage après transformée qu'avant («échantillonnage critique»), il est nécessaire de faire appel aux résidus «critiques», qui ont été défini ci-dessus grâce à une contrainte supplémentaire de reconstruction parfaite d'un banc de filtres. On obtient alors la Transformée en Ondelettes Discrètes (TOD), qui, à partir du signal original x, calcule les résidus aux échelles 2^{-j} et résolutions 2^{-j}, $j = 1, \ldots, J$, ainsi que la version basse-résolution à l'échelle et résolution 2^{-J}. Cette description suffit à définir entièrement la TOD.

Les signaux (coefficients) calculés par la TOD, avec J niveaux de décomposition, sont donc, d'après les règles de calcul décrites plus haut :

$$d^j = \downarrow \mathbf{H}'(\downarrow \mathbf{G}')^{j-1} x, \qquad \text{pour } j = 1, \ldots, J \tag{2.27}$$

$$c^J = (\downarrow \mathbf{G}')^J x. \tag{2.28}$$

La TOD inverse ramène tous les coefficients à l'échelle 1 et effectue la somme pour reconstruire x :

$$x = (\sum_{j=1}^{J} (\mathbf{G}\uparrow)^{j-1} \mathbf{H}\uparrow d^j) + (\mathbf{G}\uparrow)^J c^J. \tag{2.29}$$

En regardant ces formules d'assez près, on s'aperçoit [1] qu'elles constituent une simple écriture, en termes d'opérateurs, du banc de filtres itéré sur J octaves de la figure 2.1. Par

conséquent (du point de vue structurel), *la TOD n'est ni plus ni moins qu'un banc de filtres itéré en octaves*[6].

Il est temps, maintenant, d'exhiber les fonctions de base «ondelettes discrètes» et les notions de bi-orthogonalité/orthogonalité pour la TOD (ce qui a déjà été fait pour $J = 1$ en § 2.1.3). On a besoin pour cela d'une dernière propriété des opérateurs :

Comme il est mentionné dans [1], la para-conjugaison (2.9) s'interprète comme une transposition hermitienne d'opérateurs (matrice associée transposée et conjuguée). Ainsi, $\downarrow\tilde{\mathbf{G}}$, correspondant au filtre para-conjugué $\tilde{G}(z)$, a pour transposé hermitien $\mathbf{G}\uparrow$. De même, $\tilde{\mathbf{G}}'\uparrow$ est le transposé hermitien de $\downarrow\mathbf{G}'$. De façon générale, le transposé hermitien se calcule en changeant l'ordre des opérateurs, le sens des flèches, et en para-conjuguant[7]. Le transposition hermitienne d'un opérateur $\mathbf{O}$ se traduit également en terme de produits scalaires, par la relation $\langle x, \mathbf{O}y\rangle = \langle \mathbf{O}^\dagger x, y\rangle$, ou $\langle x, \mathbf{O}^\dagger y\rangle = \langle \mathbf{O}x, y\rangle$. Ces relations sont très utiles pour dériver les résultats qui suivent de manière concise [1]. Pour les utiliser, il suffit de se rappeler que le passage d'un opérateur d'un coté à l'autre dans un produit scalaire fait passer au hermitien conjugué.

A l'aide de cette dernière propriété, on peut ré-écrire (2.27), (2.28) en fonction d'un produit scalaire :

$$\begin{aligned} d_k^j &= \langle \downarrow\mathbf{H}'(\downarrow\mathbf{G}')^{j-1} x_n, \delta_{n-k}\rangle \\ &= \langle x_n, (\tilde{\mathbf{G}}'\uparrow)^{j-1}\tilde{\mathbf{H}}'\delta_{n-k}\rangle \\ &= \langle x_n, \left((\tilde{\mathbf{G}}'\uparrow)^{j-1}\tilde{h}'\right)_{n-2^j k}\rangle \end{aligned}$$

La fonction de base contre laquelle le produit scalaire est effectué est donc $(\tilde{\mathbf{G}}'\uparrow)^{j-1}\tilde{h}'$. Le cas de c^J est similaire. Passons maintenant à la formule de reconstruction (2.29). Un de termes de la somme est

$$\begin{aligned} (\mathbf{G}\uparrow)^{j-1}\mathbf{H}\uparrow d_n^j &= \sum_k d_k^j (\mathbf{G}\uparrow)^{j-1}\mathbf{H}\uparrow\delta_{n-k} \\ &= \sum_k d_k^j \left((\mathbf{G}\uparrow)^{j-1} h\right)_{n-2^j k} \end{aligned}$$

La fonction de base est ici $(\mathbf{G}\uparrow)^{j-1}h$. Le terme basse-résolution conduit à un résultat similaire.

En résumé, nous avons montré que les coefficients de la TOD sont des produits scalaires de x par des fonctions de base d'analyse $\tilde{h'}^j_{n-2^j k}$, appelées *ondelettes discrètes d'analyse*, et $\tilde{g'}^J_{n-2^J k}$:

$$\begin{aligned} d_k^j &= \langle x_n, \tilde{h'}^j_{n-2^j k}\rangle, \qquad \text{pour } j = 1, \ldots, J \\ c_k^J &= \langle x_n, \tilde{g'}^J_{n-2^J k}\rangle \end{aligned} \tag{2.30}$$

et que la reconstruction utilise des fonctions de base de synthèse $h^j_{n-2^j k}$, appelées *ondelettes discrètes de synthèse*, et $g^J_{n-2^J k}$:

$$x_n = \sum_{j=1}^{J}\sum_k d_k^j\, h^j_{n-2^j k} + \sum_k c_k^J\, g^J_{n-2^J k}. \tag{2.31}$$

[6] Pourquoi, alors, donner un nom différent à ce qui est identique ? La dénomination «TOD» sera justifiée au chapitre 3, en faisant le parallèle avec les séries d'ondelettes. Il faut noter, tout de même, que la TOD a été définie par une approche «à la ondelettes» (multi-résolution), très différente de celle utilisée pour décrire les bancs de filtres.

[7] Ceci correpond exactement à une transposition de graphes de fluence [1], qui consiste à inverser le sens des directions du flot de données – les noeuds de sommation du graphe deviennent des noeuds de branchement et *vice versa* – et à conjuguer les multiplicateurs.

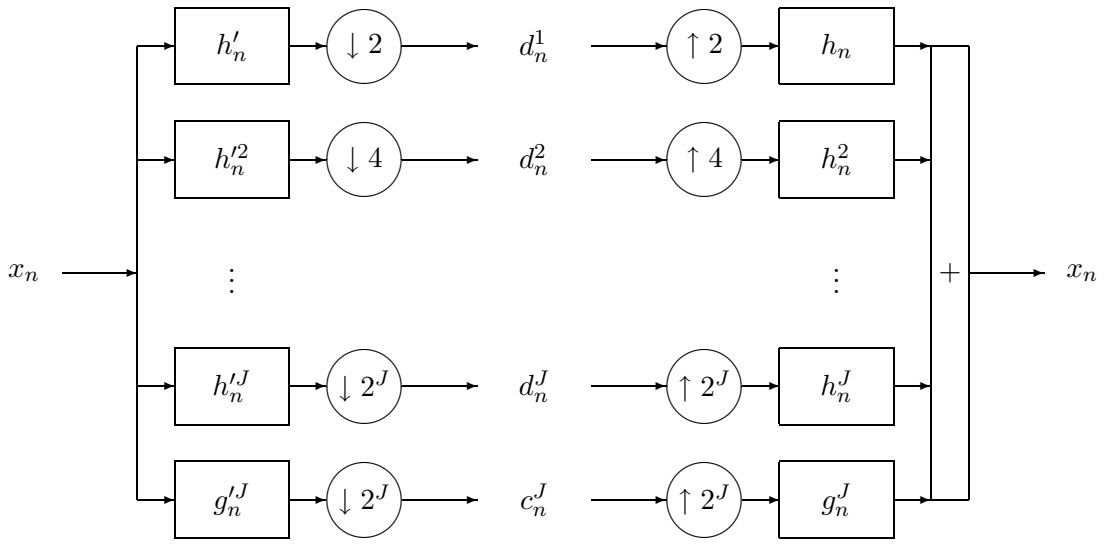


Figure 2.5: Fonctions de base dans un banc de filtres itéré en octaves. Ceci est une simple réécriture de la figure 2.1 (sous forme non-causale), qui laisse apparaître, pour chaque niveau de résolution, des réponses impulsionnelles équivalentes associées aux ondelettes discrètes.

La figure 2.5 illustre les formules précédentes sous forme de graphe de fluence. Les fonctions de base «ondelettes» sont simplement définies par doublements d'échelles itérés à partir des réponses des filtres $G(z)$, $H(z)$, $G'(z)$ et $H'(z)$. Celles d'analyse sont $\{\tilde{h'}^j_{n-2^jk}\}$, $j = 1, \ldots, J$, et $\{\tilde{g'}^J_{n-2^Jk}\}$, où :

$$\tilde{h'}^j = (\tilde{\mathbf{G}}'\uparrow)^{j-1}\tilde{h}' \qquad \text{et} \qquad \tilde{g'}^J = (\tilde{\mathbf{G}}'\uparrow)^{J-1}\tilde{g}'$$

et celles de synthèse sont $\{h^j_{n-2^jk}\}$, $j = 1, \ldots, J$, et $\{g^J_{n-2^Jk}\}$, où :

$$h^j = (\mathbf{G}\uparrow)^{j-1}h \qquad \text{et} \qquad g^J = (\mathbf{G}\uparrow)^{J-1}g.$$

La figure 2.5 permet d'identifier les ondelettes discrètes $\{\tilde{h'}^j\}$ et $\{h^j\}$ comme des réponses impulsionnelles de filtres passe-bande, alors que $\{\tilde{g'}^J\}$ et $\{g^J\}$ sont des réponses impulsionnelles de filtres passe-bas. En revenant aux transformées en z, il est facile de déterminer les fonctions de transfert des filtres itérés de la figure 2.5, dont les réponses impulsionnelles sont $\tilde{g'}^j_n$, $\tilde{h'}^j_n$, g^j_n, et h^j_n :

$$\tilde{G'}^j(z) = \tilde{G}'(z)\,\tilde{G}'(z^2)\,\tilde{G}'(z^4)\cdots\tilde{G}'(z^{2^{j-2}})\,\tilde{G}'(z^{2^{j-1}}) \tag{2.32}$$

$$\tilde{H'}^j(z) = \tilde{G'}^{j-1}(z)\tilde{H}'(z^{2^{j-1}}) \tag{2.33}$$

$$G^j(z) = G(z)\,G(z^2)\,G(z^4)\cdots G(z^{2^{j-2}})\,G(z^{2^{j-1}}) \tag{2.34}$$

$$H^j(z) = G^{j-1}(z)H(z^{2^{j-1}}). \tag{2.35}$$

2.4.1 Cas bi-orthogonal

La cas bi-orthogonal correspond au cas général. Nous avions déjà montré en § 2.1.3 que les fonctions de base vérifiaient la propriété de bi-orthogonalité dans le cas d'une seule décomposition ($J = 1$). Par extension directe de la formule (2.24), cette propriété de bi-orthogonalité s'exprime,

en fonctions des opérateurs, comme $\downarrow\mathbf{G}'\mathbf{G}\uparrow = \downarrow\mathbf{H}'\mathbf{H}\uparrow = \mathbf{Id}$ et $\downarrow\mathbf{G}'\mathbf{H}\uparrow = \downarrow\mathbf{H}'\mathbf{G}\uparrow = 0$. Maintenant, muni seulement de cette propriété établie pour un niveau de décomposition, il est facile de montrer qu'elle s'étend automatiquement à toute la TOD.

Par exemple, la bi-orthogonalité entre les ondelettes discrètes d'analyse et de synthèse correspondant à l'octave j se démontre par :

$$\begin{aligned}\langle h^j_{n-2^j_k}, \tilde{h'}^j_{n-2^j_l}\rangle &= \langle(\mathbf{G}\uparrow)^{j-1}\mathbf{H}\uparrow\delta_{n-k}, (\tilde{\mathbf{G}}'\uparrow)^{j-1}\tilde{\mathbf{H}}'\uparrow\delta_{n-l}\rangle \\ &= \langle\downarrow\mathbf{H}'(\downarrow\mathbf{G}')^{j-1}(\mathbf{G}\uparrow)^{j-1}\mathbf{H}\uparrow\delta_{n-k}, \delta_{n-l}\rangle \\ &= \langle\delta_{n-k}, \delta_{n-l}\rangle = \delta_{k,l}.\end{aligned}$$

La bi-orthogonalité des ondelettes discrètes est donc une conséquence directe de celle, établie pour un niveau de décomposition, des fonctions de base correspondant à un banc de filtres à reconstruction parfaite.

2.4.2 Cas orthogonal

Le cas orthogonal correspond, pour un niveau de décomposition du banc de filtres, au cas où les deux opérateurs de changement d'échelle (2.19) et (2.20) sont transposés hermitiens l'un de l'autre. De ce fait, imposer l'orthogonalité est équivalent à imposer que le banc de filtres d'analyse est le transposé conjugué de celui de synthèse. Ceci a d'ailleurs des conséquences importantes pour la dérivation d'algorithmes rapides (voir chapitre 6).

Là aussi, il est facile de voir que l'orthogonalité établie pour un niveau de décomposition s'étend à toute la TOD. Cela correspond au cas où les fonctions de base sont les mêmes que celles de synthèse : $\tilde{h'}^j = h^j$, $\tilde{g'}^J = g^J$. Alors, la famille de fonctions de base «ondelettes» $\{h^j_{n-2^j k}, g^J_{n-2^J k}\}$, pour $j = 1, \ldots, J$ et k entier, forment une base orthonormale.

2.5 Extensions possibles

Bien que cela pose quelques difficultées techniques, toute la description effectuée dans ce chapitre peut s'étendre au cas des filtres à réponse impulsionnelle infinie. Il est également possible d'étendre cette approche au cas bidimensionel non-séparable. On pourrait alors définir, par exemple, une transformée pyramidale non-séparable, et une TOD non-séparable, où le banc de filtres associé reste à deux sous-bandes.

Du point de vue des concepts théoriques, cependant, ces descriptions étendues n'apporteraient pas grand chose de plus. Le contenu de ce chapitre, correspondant au cas le plus simple, suffit à définir toutes les notions indispensables.

2.6 Conclusion

Nous avons montré, par une approche multi-résolution en temps-discret assez générale, que l'on pouvait définir deux principaux systèmes de décomposition multi-résolution : les transformées pyramidales et la transformée en ondelettes discrètes (TOD). Cette dernière a été définie sans qu'il ne soit nulle part mention des autres types de transformées en ondelettes, comme les séries d'ondelettes continues. Shensa [9] a, indépendamment de ce travail, insisté également sur la considération de la TOD comme technique à part entière.

Cependant, la TOD se trouve être exactement la même chose qu'un banc de filtres itéré en octaves. On peut donc interpréter un banc de filtres itéré en octaves comme une transformée associée à une décomposition multi-résolution du signal sur des fonctions de base («ondelettes»). La reconstruction parfaite du banc de filtres est équivalente à la propriété de *biorthogonalité* des fonctions de base.

Les fonctions de base «ondelettes» forment une base orthogonale dans le cas où le banc de filtres de synthèse est transposé de celui d'analyse. Le banc de filtres peut alors être vu comme une *transformée orthogonale* : ceci a mis en cause, dans la communauté du traitement du signal, les préjugés qui affirmaient que l'orthogonalité était l'apanage uniquement des transformées telles que la TCD (Transformée en Cosinus Discrète) ou la TFD (Transformée de Fourier Discrète). De fait, l'orthogonalité peut également être exploitée dans les systèmes de codage utilisant des bancs de filtres (*cf.* chapitre 7).

Nous avons également insisté sur le fait qu'il suffit de déterminer les propriétés d'orthogonalité ou de bi-orthogonalité (reconstruction parfaite) pour un seul niveau de décomposition. Ces propriétés s'étendent immédiatement, par itération en octaves, à toute la TOD.

Il reste, pour concrétiser les résultats de ce chapitre, à comparer la TOD avec la transformée en ondelettes classique utilisée pour les signaux analogiques. C'est cette comparaison, effectuée au chapitre suivant, qui nous permettra de déterminer ce qu'il y a de neuf, en pratique, dans l'approche «ondelettes» par rapport aux bancs de filtres. Pour l'heure, nous avons simplement vu que les deux approches sont basées sur des concepts différents, mais sont identiques du point de vue de la structure des calculs.

Bibliographie

[1] O. Rioul, "A discrete-time multiresolution theory," *IEEE Trans. Signal Processing*, vol. 41, no. 8, Aug. 1993. To appear.

[2] P. P. Vaidyanathan, "Quadrature mirror filter banks, M-band extensions and perfect-reconstruction techniques," *IEEE ASSP Magazine*, vol. 4, no. 3, pp. 4–20, July 1987.

[3] M. Vetterli and D. L. Gall, "Perfect reconstruction FIR filter banks : Some properties and factorizations," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 7, pp. 1057–1071, July 1989.

[4] P. P. Vaidyanathan and Z. Doganata, "The role of lossless systems in modern digital signal processing," *IEEE Trans. Education*, vol. 32, no. 3, pp. 181–197, Aug. 1989.

[5] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Comm. Pure Applied Math.* To appear.

[6] F. Mintzer, "Filters for distortion-free two-band multirate filter banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, no. 3, pp. 626–630, June 1985.

[7] M. J. T. Smith and T. P. Barnwell, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 3, pp. 434–441, June 1986.

[8] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math*, vol. XLI, no. 7, pp. 909–996, 1988.

[9] M. J. Shensa, "The discrete wavelet transform : Wedding the à trous and Mallat algorithms," *IEEE Trans. Signal Processing*, vol. 40, no. 10, pp. 2464–2482, Oct. 1992.

[10] T. Blu, "Iterated filter banks with rational factors : Links with discrete wavelet transforms," *IEEE Trans. Signal Processing.* Special issue on wavelets, to appear.

[11] Y. Meyer, *Ondelettes et Opérateurs, tome I*, Paris : Herrmann, 1990.

[12] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Comm.*, vol. 31, no. 4, pp. 532–540, Apr. 1983.

Chapitre 3

Ondelettes continues

Never create a problem for which you do not have the answer.
— BURKE'S RULE
Create problems for which only **you** *have the answer.*
— COROLLARY TO BURKE'S RULE

Les Séries d'Ondelettes (SO) en temps continu ont été brièvement présentées au chapitre d'introduction. Nous avons mentionné qu'elles constituent la technique classique qui est généralement référencée dans les travaux portant sur l'application des ondelettes au codage de source. Cependant, cette technique a été définie pour les signaux analogiques, alors que la plupart des problèmes de codage traités sont essentiellement des problèmes posés en temps-discret, où la source est numérique.

Ceci nous incite à effectuer une comparaison pratique entre les séries d'ondelettes, présentées au départ comme un outil mathématique, et la TOD, transformée en ondelettes «temps-discret» introduite au chapitre précédent, qui, elle, s'identifie aux techniques «bancs de filtres» du traitement du signal. Cette comparaison va nous permettre de reconnaître l'application de principes communs entre les deux techniques, et d'identifier les principes réellement nouveaux apportés par la théorie des ondelettes. Le contenu de ce chapitre se réfère en partie à l'article [1], reproduit à l'appendice A 2, et en partie à l'article [2], reproduit à l'appendice A 6.

3.1 Rappel sur les séries d'ondelettes

Il est inutile de s'attarder ici sur la théorie des séries d'ondelettes, qui est exposée dans de nombreuses publications (voir par exemple [3]–[8]). Nous nous bornerons à quelques formules essentielles permettant la comparaison avec la TOD. On se place ici dans le cadre général des séries d'ondelettes bi-orthogonales, traité par Cohen, Daubechies et Feauveau dans [5]. Le cas orthogonal, quant à lui, correspond à la théorie des espaces multi-résolution d'Yves Meyer [6], utilisé par Stéphane Mallat [7] dans le cadre du traitement d'images.

La décomposition, sur J octaves, associée aux séries d'ondelettes utilise deux familles de fonctions de base, l'une pour l'analyse et l'autre pour la synthèse. Les fonctions de base d'analyse sont constituées des ondelettes à temps continu, $\breve{\psi}^j(t - 2^j k)$, $j = 1, \ldots, J$, et des *fonctions d'échelle* $\breve{\varphi}^J(t - 2^J k)$. De la même façon, les fonctions de base de synthèse sont $\psi^j(t - 2^j k)$, $j = 1, \ldots, J$, et $\varphi^J(t - 2^J k)$.

Les coefficients en séries d'ondelettes sont des produits scalaires calculés lors de l'analyse :

$$\begin{aligned} D_k^j &= \langle x(t), \breve{\psi}^j(t-2^j k)\rangle \qquad \text{pour } j = -\infty, \ldots, J \\ C_k^J &= \langle x(t), \breve{\varphi}^J(t-2^J k)\rangle. \end{aligned} \tag{3.1}$$

La reconstruction du signal (synthèse) s'effectue en sommant des projections. C'est la formule de décomposition en séries d'ondelettes :

$$x(t) = \sum_{j=-\infty}^{J} \sum_k D_k^j \psi^j(t-2^j k) + \sum_k C_k^J \varphi^J(t-2^J k). \tag{3.2}$$

Toutes ces formules sont à comparer avec (2.30) et (2.31) pour la TOD : ce sont les mêmes[1]. La seule différence est que (3.1), (3.2) est écrite pour les signaux à temps continu t. Bien entendu, le produit scalaire utilisé dans ce cas est continu : $\langle x(t), y(t)\rangle = \int x(t)y^*(t)\,dt$.

La similarité des formules de calcul des coefficients de l'analyse et de reconstruction est rappelée dans la table 3.1. Ce tableau montre aussi que d'autres propriétés, notamment celles de bi-orthogonalité et d'orthogonalité, s'écrivent aussi de manière identique dans la cas continu (SO) et discret (TOD). Le parallélisme est saisissant : il suggère un lien très fort entre les transformées en ondelettes continues et discrètes, dont on va parler plus loin. Il justifie donc *a posteriori* la dénomination "Transformée en Ondelettes Discrètes", introduite au chapitre précédent comme un synonyme des bancs de filtres itérés en octaves.

Il y a en fait seulement deux différences majeures qui apparaissent dans la table 3.1 :

- Dans les deux cas, les ondelettes correspondant au niveau de résolution 2^{-j} se déduisent d'une seule ondelette prototype par changement (augmentation) d'échelle. Cependant, dans le cas continu, l'augmentation d'échelle est faite par une simple dilatation en temps. Il est bien sûr impossible de faire de même dans le cas discret : nous avons, au chapitre 2, utilisé les opérateurs de changement d'échelle pour pallier cet inconvénient. Il est à noter que ces changements d'échelle dépendent alors des filtres, paramètres de la transformée. Cette différence, cependant, ne porte pas vraiment à conséquence, car, comme on va le voir, les séries d'ondelettes sont implantées en pratique en utilisant une TOD.

- La principale différence, en fait, correspond à la case manquante de la table 3.1, qui concerne la «régularité». Cela signifie, pour le cas continu, que les fonctions de base «ondelettes» sont continues, dérivables, ou plus généralement, possèdent N dérivées continues ($\in C^N$). Cette propriété est considérée comme essentielle pour les mathématiciens [5, 6]. Bien entendu, ces conditions de régularité sont impossibles à définir telles quelles en temps discret.

En résumé, la TOD rassemble toutes les notions que la théorie des ondelettes continues a apporté au traitement du signal, mise à part, peut-être, celle de régularité[2].

[1] Dans cette présentation des séries d'ondelettes, j est l'opposé du paramètre j utilisé habituellement par les théoriciens comme Yves Meyer [6]. En effet, l'idée initiale, dans le cas continu, est que on ré-obtient le signal original en faisant tendre $j \to \infty$ (jusqu'à obtenir une résolution infinie). Dans le cas discret, le résolution du signal originel est finie (par convention égale à 1), ce qui correspond à $j = 0$. Si on avait suivi la convention habituelle, on n'aurait eu que des valeurs négatives de j dans le cas discret.

Cette discussion explique aussi la borne inférieure $-\infty$ au lieu de 1 dans les formules (3.1) et (3.2).

[2] Il y a aussi la propriété de *moments nuls* [5, 8, 6] qui se traduit également très bien dans le cas discret. Nous y reviendrons ultérieurement, au chapitre 4.

CONTINU	DISCRET
Changement d'échelle : Réduction par 2	
$x(t) \mapsto \sqrt{2}x(2t)$	$x_n \mapsto (\downarrow \mathbf{G}x)_n$
Aggrandissement par 2	
$x(t) \mapsto x(t/2)/\sqrt{2}$	$x_n \mapsto (\mathbf{G}\uparrow x)_n$
Ondelette et fonction d'échelle prototype d'analyse et de synthèse :	
$\breve{\psi}(t)$ et $\psi(t)$	$\tilde{h'}_n$ et h_n
$\breve{\varphi}(t)$ et $\varphi(t)$	$\tilde{g'}_n$ et g_n
Fonctions de base multi-échelles d'analyse :	
$\breve{\psi}^j(t) = 2^{-j/2}\breve{\psi}(2^{-j}t)$	$\tilde{h'}^j = (\mathbf{G}'\uparrow)^{j-1}\tilde{h'}$
$\breve{\varphi}^j(t) = 2^{-j/2}\breve{\varphi}(2^{-j}t)$	$\tilde{g'}^j = (\mathbf{G}'\uparrow)^{j-1}\tilde{g'}$
de synthèse :	
$\psi^j(t) = 2^{-j/2}\psi(2^{-j}t)$	$h^j = (\mathbf{G}\uparrow)^{j-1}h$
$\varphi^j(t) = 2^{-j/2}\varphi(2^{-j}t)$	$g^j = (\mathbf{G}'\uparrow)^{j-1}g$
Coefficients de transformée (analyse)	
$D_k^j = \langle x(t), \breve{\psi}^j(t-2^jk)\rangle$	$d_k^j = \langle x_n, \tilde{h'}^j_{n-2^jk}\rangle$
$j = -\infty, \ldots, J$, $k \in \mathbf{Z}$	$j = 1, \ldots, J$, $k \in \mathbf{Z}$
$C_k^J = \langle x(t), \breve{\varphi}^J(t-2^Jk)\rangle$	$c_k^J = \langle x_n, \tilde{g'}^J_{n-2^Jk}\rangle$
Reconstruction (synthèse)	
$x(t) = \sum_{j=-\infty}^{J}\sum_k D_k^j\psi^j(t-2^jk)$ $+\sum_k C_k^J\varphi^J(t-2^Jk).$	$x_n = \sum_{j=1}^{J}\sum_k d_k^j\, h^j_{n-2^jk}$ $+\sum_k c_k^J\, g^J_{n-2^Jk}.$
Biorthogonalité :	
$\langle\psi^j(t-2^jk), \breve{\psi}^j(t-2^jl)\rangle = \delta_{k,l}$	$\langle h^j_{n-2^jk}, \tilde{h'}^j_{n-2^jl}\rangle = \delta_{k,l}$
$\langle\varphi^j(t-2^jk), \breve{\varphi}^j(t-2^jl)\rangle = \delta_{k,l}$	$\langle g^j_{n-2^jk}, \tilde{g'}^j_{n-2^jl}\rangle = \delta_{k,l}$
$\langle\varphi^j(t-2^jk), \breve{\psi}^j(t-2^jl)\rangle = 0$	$\langle g^j_{n-2^jk}, \tilde{h'}^j_{n-2^jl}\rangle = 0$
$\langle\psi^j(t-2^jk), \breve{\varphi}^j(t-2^jl)\rangle = 0$	$\langle h^j_{n-2^jk}, \tilde{g'}^j_{n-2^jl}\rangle = 0$
Orthogonalité :	
$\langle\psi^j(t-2^jk), \psi^i(t-2^il)\rangle = \delta_{k,l}\delta_{i,j}$	$\langle h^j_{n-2^jk}, h^i_{n-2^il}\rangle = \delta_{k,l}\delta_{i,j}$
$\langle\varphi^j(t-2^jk), \varphi^j(t-2^jl)\rangle = \delta_{k,l}$	$\langle g^j_{n-2^jk}, g^j_{n-2^jl}\rangle = \delta_{k,l}$
Etendue temporelle :	
Support compact	Filtres RIF
Phase linéaire :	
$\varphi(t)$ ou $\breve{\varphi}(t)$ symétrique	g_n ou $\tilde{g'}_n$ symétrique
$\psi(t)$ ou $\breve{\psi}(t)$ symétrique/antrisymétrique	h_n ou $\tilde{h'}_n$ symétrique/antisymétrique
Régularité :	
$\varphi(t) \in C^N$	? ? (voir chapitre 4)
$\psi(t) \in C^N$	? ? (voir chapitre 4)

Table 3.1: Comparaison des ondelettes continues et discrètes.

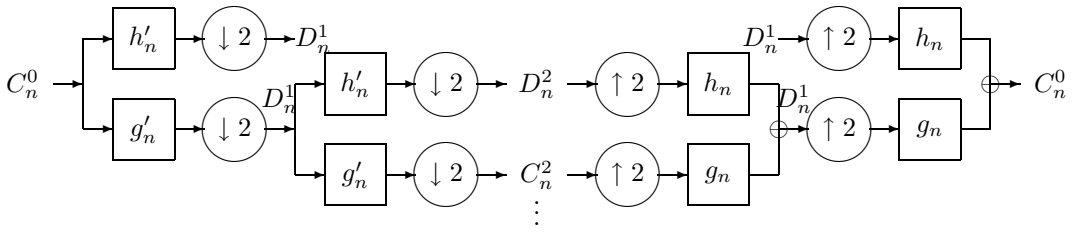


Figure 3.1: L'algorithme de S. Mallat. Deux itérations sont représentées ici. Les coefficients C^j et D^j, définis par (3.1), correspondent respectivement aux fonctions d'échelle et aux ondelettes *temps-continu*.

3.2 L'algorithme de S. Mallat

Malgré le parallélisme évident que nous venons d'établir entre la TOD et les séries d'ondelettes, nous avons déjà mentionné que la plupart des travaux sur le codage baptisé «par ondelettes» utilisent comme point de départ les séries d'ondelettes continues. Pour comprendre ce point, il convient donc de regarder la manière dont les séries d'ondelettes sont effectivement utilisées, en pratique, pour le codage ou toute autre application portant sur des signaux discrets.

Stéphane Mallat [7] fut le premier à donner un algorithme pratique permettant de calculer les coefficients en séries d'ondelettes d'un signal. Bien qu'il ait été proposé dans le cas orthogonal, nous l'exposons ici dans le cas bi-orthogonal traité par Cohen, Daubechies et Feauveau [5].

La dérivation de l'algorithme de S. Mallat est basée sur des propriétés particulières que vérifient les bases bi-orthogonales d'ondelettes [5]. Ces propriétés prennent la forme d'équations aux différences, à deux échelles (*two-scale difference equations*), où apparaissent des réponses impulsionnelles de filtres $\tilde{g'}_n$, g_n, $\tilde{h'}_n$, et h_n associées aux fonctions de base à temps-continu :

$$\begin{aligned}
\breve{\varphi}(t) &= \sqrt{2}\sum_n \tilde{g'}_n \breve{\varphi}(2t-n).\\
\varphi(t) &= \sqrt{2}\sum_n g_n \varphi(2t-n).\\
\breve{\psi}(t) &= \sqrt{2}\sum_n \tilde{h'}_n \breve{\varphi}(2t-n).\\
\psi(t) &= \sqrt{2}\sum_n h_n \varphi(2t-n).
\end{aligned} \tag{3.3}$$

Grâce à ces équations, on peut, à partir de la donnée d'une décomposition en série d'ondelettes particulière, en *déduire* un banc de filtres itéré en octaves (TOD) associé. Le banc de filtres d'analyse permet d'effectuer un calcul itératif des coefficients en séries d'ondelettes, comme montré dans la figure 3.1 : c'est l'algorithme de S. Mallat, dont la partie reconstruction utilise le banc de filtres de synthèse.

Cet algorithme permet donc d'aller dans le sens continu→discret, et de reconnaitre une TOD sous-jacente aux séries d'ondelettes. Cette TOD sera, en fin de compte, toujours utilisée dans l'implantation d'une série d'ondelettes. De plus, on peut montrer que toutes les propriétés des séries d'ondelettes se reportent sur les propriétés correspondantes de la TOD, c'est à dire que les propriétés décrites dans la colonne de gauche de la table 3.1 impliquent celles de la colonne de droite, pour la TOD associée. Par exemple, une série d'ondelettes orthogonales induit une TOD orthogonale.

Cet algorithme nous montre comment on peut appliquer les séries d'ondelettes à des signaux temps-discret : il suffit de considérer le signal temps-discret originel, x_n, comme la suite des coefficients d'échelle C_n^0, associées à l'analyse à la résolution 1 d'un signal analogique sous-jacent $x(t)$ par la formule :

$$x_n = C_n^0 = \int x(t)\varphi(t-k)\,dt. \tag{3.4}$$

On prend donc C_n^0 comme signal d'entrée de la TOD, qui est reconstruit à la synthèse.

Cependant, cette vision des choses, qui consiste à appliquer un modèle analogique sur un signal temps-discret pose problème, en particulier lorsque le signal originel n'est connu que sous forme numérisée, comme c'est souvent le cas en codage d'images. En effet, on est alors obligé de considérer que l'objet mathématique analysé est un signal analogique $x(t)$, duquel provient x_n par échantillonnage. En particulier, rien ne dit que ce signal analogique vérifie la relation (3.4), qui présuppose que la conversion numérique-analogique utilise précisément le filtre $\varphi(t)$, qui est un paramètre de la transformée !

On pourrait, comme il est fait dans [2], supposer une relation générale entre signaux discret et continu de la forme $x(t) = \sum_n x_n\chi(t-n)$. L'entrée de la TOD serait alors non pas x_n lui-même, mais x_n *préfiltré* par le filtre de réponse $f_n = \int \chi(t)\varphi^*(t-n)\,dt$, afin que les coefficients en série d'ondelettes calculés correpondent effectivement à $x(t)$. Mais ceci reporte le problème sur la détermination de $\chi(t)$ dans la calul de f_n.

Ces considérations nous confortent donc dans l'idée qu'il est plus logique d'utiliser directement une TOD pour les problèmes de nature discrète, comme c'est le cas en codage d'images. On considère alors directement le signal discret x_n comme l'«objet mathématique» sur lequel s'applique la théorie. Le seul inconvénient est qu'on perd virtuellement la notion de régularité des fonctions de base introduite au § 3.1. Cependant, grâce aux relations (3.3), la régularité des ondelettes continues devrait pouvoir impliquer des conditions particulières sur le banc de filtres/TOD associé.

Afin de préciser ce dernier point, on va maintenant décrire le passage inverse du cas discret au cas continu, qui va nous permettre d'établir une véritable équivalence entre séries d'ondelettes et TOD.

3.3 L'algorithme d'I. Daubechies

Le passage discret→continu consiste à construire les ondelettes temps-continu associées à une décomposition en séries d'ondelettes, à partir de la donnée d'un banc de filtres itéré en octaves (TOD). C'est l'approche généralement suivie pour construire des bases d'ondelettes bi-orthogonales [5] ou orthogonales [8]. Ainsi les bancs de filtres apparaissent non seulement lorsque l'on veut implanter une décomposition en séries d'ondelettes, mais aussi pour déterminer les ondelettes continues elles-mêmes.

Cette détermination est faite par un algorithme de subdivision, utilisé pour la première fois par Ingrid Daubechies [8] dans le cas orthogonal. Cet algorithme consiste à considérer les fonctions de base discrètes du banc de filtres itéré en octaves, g_n^j, h_n^j, $\tilde{g'}^j{}_n$, et $\tilde{h'}^j{}_n$, et à faire tendre j vers l'infini (le banc de filtres est itéré indéfiniment).

Considérons par exemple g_n^j, qui correpond au filtre RIF passe-bas $G(z)$ de réponse impulsionnelle g_n et de longueur L. D'après la formule (2.34) du chapitre précédent, la transformée en z correspondante est :

$$G^j(z) = G(z)\,G(z^2)\cdots G(z^{2^{j-1}}).$$

La longueur de la séquence g_n^j est donc $(2^j-1)(L-1)+1$: elle est a peu près doublée à chaque itération, c'est à dire, avec les notations du chapitre précédent, à chaque doublement d'échelle par l'opérateur $\mathbf{G}\uparrow$. Un exemple est présenté sous forme de graphiques pour différentes valeurs de j à la figure 3.2, où l'on a ramené toutes les courbes à une même étendue en traçant les valeurs de g_n^j en fonction de l'abscisse $n2^{-j}$.

On observe, que *sous certaines conditions sur le filtre* $G(z)$, les courbes g_n^j, tracées en fonction de $n2^{-j}$, convergent[3], lorsque j croît, vers une fonction limite qui définit la fonction d'échelle $\varphi(t)$ (*cf.* figure 3.3). On définit de la même manière $\psi(t)$, $\breve{\varphi}(t)$ et $\breve{\psi}(t)$ à partir de h_n^j, $\tilde{g'}_n^j$, et $\tilde{j'}_n^j$. Notons que les courbes limites obtenues sont nécessairement à *support compact* (étendue limitée) puisque les filtres considérés sont RIF. On se placera dans ce cas tout au long de cette thèse.

La figure 3.3 (b) montre que les filtres classiquement utilisés pour les décompositions en sous-bandes [9] ne génèrent pas forcément des courbes régulières. Ingrid Daubechies [8] a donné des exemples pour lesquels la convergence a lieu, ponctuellement, vers des fonctions de base effectivement *régulières*. On peut montrer qu'alors [8, 5] les fonctions de base à temps-continu obtenues définissent bien une décomposition en séries d'ondelettes, dont les propriétés (orthogonalité, etc.) s'obtiennent à partir de celles de la TOD. Ainsi, les propriétés de la colonne de droite de la table 3.1 impliquent celles de la colonne de gauche. De plus, les fonctions limites obtenues vérifient les relations (3.3) : la TOD qui permet l'implantation du calcul de séries d'ondelettes est donc la même que celle qui a servi à construire les ondelettes continues. L'équivalence, dans ce cas, est totale[4].

Tour ceci justifie pleinement de ne se consacrer qu'au cas discret pour des applications telles que le codage : le seul intérêt supplémentaire qu'apportent les séries d'ondelettes est la propriété de régularité. Mais puisqu'il y équivalence, on doit pouvoir traduire la notion de régularité en temps-discret, grâce à l'algorithme d'I. Daubechies. D'ailleurs, la construction de bases d'ondelettes à support compact régulières [8] s'est faite précisément à partir de filtres ayant de bonnes propriétés.

3.4 Conclusion

Nous avons montré que, de façon générale, la plupart des propriétés apportées par les ondelettes se retrouvent sur les bancs de filtres, pour la TOD, et nous avons rappelé qu'une décomposition en séries d'ondelettes se calcule, au bout du compte, par un banc de filtres/TOD. Il n'est donc pas étonnant que les ondelettes aient trouvé des applications immédiates en codage : elles constituent essentiellement une technique de codage en sous-bandes, qui a, comme on l'a mentionné au chapitre d'introduction, déjà été utilisée avec succès pour des problèmes de compression d'images ou de parole.

Nous avons néanmoins identifié la *régularité* des fonctions de base comme la seule propriété réellement nouvelle apportée par la théorie des ondelettes (par le biais des séries d'ondelettes) dans le cadre des bancs de filtres itérés en octave.

Malgré cela, l'outil temps-discret est plus adapté aux problèmes concernant les signaux temps-discret (comme le codage d'images). Nous avons en effet insisté sur les problèmes posés

[3] On donnera des définitions plus précises au chapitre suivant.

[4] Certains auteurs [5, 10] ont recherché des conditions minimales pour lesquelles l'équivalence entre TOD et séries d'ondelettes a lieu. Ici, on a mentionné qu'une possiblité, celle où les processus d'itération sur les filtres (fonctions de base discrètes) converge vers les fonctions limites régulières. C'est néanmoins le cas le plus intéressant, car il correspond à une propriété supplémentaire des séries d'ondelettes qu'on veut pouvoir traduire sur la TOD.

(a)

(b)

(c)

Figure 3.2: Réponses impulsionnelles de filtres itérés. On a représenté ici, pour un exemple de filtre $G(z)$, les courbes g_n^j, tracées en fonction de $n2^{-j}$, pour (a) $j = 1$, (b) $j = 2$, et (c) $j = 3$.

(a)

(b)

Figure 3.3: Fonctions $\varphi(t)$ obtenues après convergence. (a). Courbe limite correspondant au processus itératif représentée à la figure 3.2. Elle est régulière (deux fois continûment dérivable). (b). Courbe limite correspondant à un filtre de Johnston [9], classiquement utilisé dans des bancs de filtres à deux sous-bandes. Elle est continue, mais pas dérivable.

lorsque l'on veut utiliser l'outil analogique (séries d'ondelettes) pour des signaux temps-discret. En conséquence, nous nous consacrerons qu'aux ondelettes discrètes (bancs de filtres) dans la suite de la thèse.

Il s'agit maintenant de pouvoir traduire la notion de régularité en temps-discret afin de pouvoir déterminer si cette notion nouvelle apporte un intérêt éventuel pour les applications. Or, nous avons vu que sous certaines conditions de convergence de filtres itérés vers des fonctions limites régulières, l'équivalence entre ondelettes discrètes et continues est totale. On doit donc pouvoir, grâce à ce processus itératif, traduire la propriété de régularité sur les réponses impulsionnelles discrètes. Autrement dit, interpréter la régularité comme une nouvelle contrainte sur les filtres utilisés dans une TOD/banc de filtres. Ceci fait l'objet du chapitre suivant, qui motive également l'étude de la régularité en montrant l'intérêt potentiel de celle-ci pour les applications de codage.

Bibliographie

[1] O. Rioul, "A discrete-time multiresolution theory," *IEEE Trans. Signal Processing*, vol. 41, no. 8, Aug. 1993. To appear.

[2] O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 569–586, Mar. 1992. Part II. Special issue on Wavelets and Multiresolution Signal Analysis.

[3] C. K. Chui, *Wavelet Analysis and its applications, vol. I. An Introduction to Wavelets*, San Diego : Academic Press, 1992.

[4] I. Daubechies, *Ten lectures on wavelets*, Philadelphia : CBMS-NSF Series in Appl. Math., SIAM Publ., 1992.

[5] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Comm. Pure Applied Math.* To appear.

[6] Y. Meyer, *Ondelettes et Opérateurs, tome I*, Paris : Herrmann, 1990.

[7] S. Mallat, "A theory for multiresolution signal decomposition : The wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674–693, July 1989.

[8] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math*, vol. XLI, no. 7, pp. 909–996, 1988.

[9] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1980, pp. 291–294.

[10] W. M. Lawton, "Necessary and sufficient conditions for constructing orthonormal wavelet bases," Tech. Rep. AD900402, Aware, inc., 1990.

Partie II

Outils

Chapitre 4

Régularité

It's a simple task to make things complex, but a complex task to make them simple.
— MEYER'S LAW
Complex problems have simple, easy-to-understand wrong answers.
— GROSSMAN'S MISQUOTE OF H. L. MENCKEN
Nobody wants to read anyone else's formulas.
— FINMAN'S LAW OF MATHEMATICS

La régularité a été introduite au chapitre précédent comme la nouvelle propriété apportée par la théorie des ondelettes par rapport aux bancs de filtres. Nous avons vu qu'il y a, sous réserve de convergence d'un algorithme de subdivision de Daubechies, équivalence entre séries d'ondelettes à temps-continu et transformée en ondelettes discrètes. Il doit donc être possible de caractériser la régularité directement sur les réponses impulsionnelles des filtres dans un banc de filtres itérés (ce que l'on a appelé fonctions de base discrètes au chapitre 2).

Notre point de départ, pour l'étude de la régularité, est précisément l'algorithme de subdivision de Daubechies, qui permet, à partir de la donnée de filtres, d'obtenir les ondelettes/fonctions d'échelle continues par passage à la limite. La relation entre ondelettes continues et filtres est loin d'être triviale, et la régularité est définie mathématiquement sur les fonctions limites. Caractériser la régularité sur les filtres est donc un problème difficile, que nous tentons de résoudre dans ce chapitre.

4.1 Préliminaires

4.1.1 Cadre de l'étude

Rappelons tout d'abord que l'algorithme de subdivision consiste à observer la limite, lorsque $j \to \infty$, des courbes «discrètes» consituées des valeurs des fonctions de base g_n^j, h_n^j, ${\tilde{g'}}_n^j$, et ${\tilde{h'}}_n^j$, tracées en fonction de $n2^{-j}$. Un exemple graphique a été donné au chapitre précédent (figures 3.2 et 3.3). Ces fonctions de base discrètes, ont, quant à elles, été définies au chapitre 2 comme le résultat d'un opérateur de doublement d'échelle itéré j fois.

Ainsi, $g_n^j = (\mathbf{G}\uparrow)^j \delta_n$ est la réponse impulsionnelle de $(\mathbf{G}\uparrow)^j$, associé au filtre passe-bas $G(z)$:

$$\mathbf{G}\uparrow : x_n \mapsto y_n = \sum_k x_k g_{n-2k}.$$

En termes de transformée en z, la suite g_n^j est donnée par (équation (2.34) du chapitre 2)

$$G^j(z) = G(z)\,G(z^2)\,G(z^4)\cdots G(z^{2^{j-2}})\,G(z^{2^{j-1}}).$$

S'il y a convergence (en un certain sens) de l'algorithme, la fonction limite obtenue est la fonction d'échelle de synthèse à temps continu $\varphi(t)$. Le processus est le même pour $\tilde{g'}_n^j$, mais cette fois-ci l'opérateur de doublement d'échelle est associé au filtre passe-bas $\tilde{G}'(z)$, et fournit, par itération, la fonction d'échelle d'analyse $\breve{\varphi}(t)$. Il y a donc, dans le cas général (bi-orthogonal) deux régularités différentes à l'analyse et à la synthèse. (Dans le cas orthogonal, elles coïncident du fait que les fonctions de base correspondantes sont égales).

Les ondelettes à temps continu, sont, quant à elles, obtenues par le même procédé d'itération des mêmes opérateurs ; Simplement l'initialisation du processus est différente : par exemple, $h_n^j = (\mathbf{G}\uparrow)^{j-1}h_n$: la séquence initiale est ici la suite des coefficients de filtre passe-haut.

On pourrait, bien sûr, se contenter de regarder la limite visuellement comme à la figure 3.3, et en déduire (subjectivement) que la courbe obtenue est plus ou moins régulière. Cependant, on montrera dans ce chapitre que ceci peut conduire à certaines surprises si l'on compare cette appréciation avec le résultat mathématique objectif. De plus, l'objet de cette thèse étant d'étudier précisément le rôle de la régularité, il est nécessaire d'obtenir une caractérisation mathématique précise et optimale (pas une simple estimation grossière).

L'objet de ce chapitre est donc d'étudier les conditions (nécessaires et suffisantes) sur les filtres pour lesquelles la convergence de l'algorithme a lieu et fournit une fonction limite *régulière* (continue, dérivable, etc.). Ceci nous permettera de mesurer la «quantité» (à définir) de régularité d'une fonction limite à partir de la donné de filtres quelconques (problème d'estimation de régularité)[1]. On se restreint ici néanmoins au cas des filtres RIF, qui génèrent, comme on l'a vu au chapitre précédent, des fonctions limites à support compact. Cette restriction, déjà mentionnée au chapitre d'introduction, est justifiée plus loin.

Les résultats présentés dans ce chapitre proviennent de l'article [1], reproduit à l'appendice A 3. On y trouvera tous les détails mathématiques nécessaires.

4.1.2 Cas "pathologiques" instables

Dans ce qui suit, on montrera une équivalence complète entre les définitions mathématiques de régularité sur les fonction limites (à temps-continu) et leurs caractérisation temps-discret, *sauf* pour quelques cas «pathologiques», dits *instables*. Dans ce chapitre, on donnera les résultats généraux sans mentionner ces cas. Le lecteur trouvera matière à les traiter dans [1].

Ce qui sera toujours vrai, c'est le fait qu'une propriété «temps-discret» implique la propriété de régularité correspondante en temps-continu (les estimations de régularité donneront donc toujours un résultat correct). Par contre, précisons-le une fois pour toutes, l'implication réciproque, (et en particulier, l'optimalité des estimations de régularité), ne sera vrai que pour les cas non-pathologiques («stables»).

Il est en fait facile d'en montrer la raison, à partir l'équation à deux échelles (3.3) sur la fonction limite $\varphi(t)$ [1], qui s'écrit

$$\varphi(t) = \sum_n g_n \varphi(2t-n). \tag{4.1}$$

[1]Notons que ce problème s'inscrit dans un cadre plus général que celui des bancs de filtres à reconstruction parfaite. Sauf mention contraire, on ne fera donc pas appel à des propriétés telles que l'orthogonalité des filtres.

En prenant $t = n2^{-j}$ et en itérant on obtient [1] :

$$\Phi^j(z) = \Phi(z)G^j(z)$$

où $\Phi^j(z)$ est la transformée en z de la suite $\varphi(n2^{-j})$ et $\Phi(z) = \Phi^1(z)$. Cette équation nous permet de relier les valeurs discrètes g_n^j aux échantillons $\varphi(n2^{-j})$. Par conséquent, pour obtenir des conditions de régularité sur g_n^j à partir de celles sur $\varphi(t)$, le filtre RIF $\Phi(z)$ doit être inversé. Puisque l'on se restreint aux suites finies, cela équivaut à la condition que $\Phi(z)$ n'ait pas de zéro sur le cercle unité. Or, en écrivant (4.1) pour $t = n$, $\Phi(z)$ vérifie de plus $\Phi(z^2) = \frac{1}{2}(\Phi(z)G(z) + \Phi(-z)G(-z))$. Clairement, le filtre $\Phi(z)$ sera *instable* si $G(z)$ *admet une paire de zéros opposés sur le cercle unité*. On conjecture dans [1] que l'instabilité est en fait équivalente à cette condition simple sur le filtre passe-bas $G(z)$.

Grâce à cette condition, on peut montrer que l'on ne rencontre jamais (à moins de le vouloir vraiment) ces cas pathologiques «instables» dans les bancs de filtres utilisés pour les applications. En effet, pour un banc de filtres orthogonaux (chapitre 2), $G(z)$ satisfait à $G(z)\tilde{G}(z) + G(-z)\tilde{G}(-z) = 2z^{-d}$, et la condition d'instabilité sur $G(z)$ est impossible à réaliser. Dans le cas bi-orthogonal, bien qu'il soit en théorie possible d'obtenir la condition d'instabilité sur $G(z)$ ou $G'(z)$, cela signifierait que la réponse fréquentielle du filtre s'annule en deux fréquences ω et $\omega+\pi$: le filtre ne pourrait alors pas être un filtre passe-bas approximativement demi-bande. Par conséquent, même dans ce cas, la condition d'instabilité ne se rencontre jamais en pratique.

4.1.3 Motivation

Au chapitre précédent, nous avons motivé les raisons pour lesquelles on se consacre désormais uniquement à l'étude de la transformée en ondelettes discrètes, c'est à dire des bancs de filtres itérés en octaves. Les ondelettes à temps-continu n'interviennent plus que en tant que fonctions limites du processus itératif qu'on vient de décrire, et nous nous proposons, dans la suite, d'étudier les proprités des fonctions limites. Or, l'obtention d'une fonction suppose implicitement que le banc de filtres est itéré *indéfiniment*, ce qui n'est bien sûr jamais le cas en pratique. Pour des applications telles que le codage d'images, le niveau maximal d'itération dépasse rarement 4.

Il serait donc souhaitable de définir la notion de régularité sur les fonctions de base discrètes telles que g_n^j, sans qu'il soit nécessaire de faire tendre j vers l'infini. Cependant, une condition de régularité sur la fonction limite va effectivement imposer que l'évolution temporelle des réponses impulsionnelles discrètes est «douce». C'est cette «douceur» (*smoothness*) que l'on veut pouvoir quantifier par l'approche qui consiste à itérer indéfiniment le schéma itératif. Puisque l'on va caractériser la régularité en temps-discret, en prendra l'habitude de qualifier également ces fonctions de base de «régulières». On parlera également de «filtre régulier», lorsqu'on voudra dire, par exemple, qu'un filtre génère une fonction limite $\varphi(t)$ «régulière».

L'étude qui sera faite dans ce chapitre ne sera justifiée que dans le cas où la convergence du schéma itératif est rapide (comme c'est le cas dans la figure 3.2). On donnera donc dans la suite des indications quant à la *vitesse de convergence* de l'algorithme de subdivision, qui justifieront l'approche du problème adoptée ici.

Jusqu'à présent, nous avons motivé l'étude de la régularité par le fait qu'elle constitue l'aspect nouveau essentiel de la théorie des ondelettes dans le cadre des bancs de filtres. Cependant, puisque la régularité impose une évolution «douce» des fonctions de base discrètes dans un banc de filtres, on peut penser que cette propriété de «douceur» a effectivement un intérêt potentiel en codage, et motiver ainsi l'étude de la régularité :

Figure 4.1: Exemple de schéma de subdivision divergent. On tracé ici la courbe g_n^j obtenue après $j = 6$ itérations. Le choix des filtres correspond ici au cas orthogonal, où le filtre passe-bas $\{g_n\}$ est donné par $g_0 = g_1 = 0.7$ et $g_2 = -g_3 = 0.1$.

A l'analyse : Supposons qu'une portion «douce» d'un signal d'entrée x_n soit analysée par des filtres non-réguliers, dont les réponses impulsionnelles présentent rapidement des discontinuités ou des allures fractales au fur et à mesure que j croît (*cf.* figure 4.1). Alors, ces discontinuities «artificielles» – qui ne sont pas dues au signal x_n lui-même – apparaissent dans les coefficients en ondelettes discrètes, lesquelles sont, comme on l'a vu au chapitre 2, les produits scalaires du signal contre ces ondelettes discrètes. Par conséquent, la régularité permetterait une «meilleure» représentation du signal par les coefficients de la transformée en ondelettes discrètes.

A la synthèse : Supposons maintenant qu'une erreur – comme par exemple une erreur de quantification – soit faite sur un coefficient en ondelettes correspondant à un niveau de résolution j. Dans le signal reconstruit, cette erreur est perçue comme une perturbation proportionnelle à la fonction de base correspondante, h_n^j. Dans des applications telles que le codage d'images, il est naturel d'imposer que cette perturbation soit «douce» : une perturbation présentant des discontinuités est probablement plus frappante à l'œil qu'une perturbation «douce», et ceci pour la même valeur de l'erreur quadratique moyenne, et donc pour le même niveau du rapport signal à bruit.

Bien entendu, ces considérations ne sont qu'hypothétiques, et l'objet de cette thèse est précisément de se donner les moyens de les confirmer ou de les infirmer.

4.1.4 Historique

Depuis les travaux de Daubechies [2], de nombreux travaux [3]–[7] ont été consacré au problème de l'estimation de régularité à partir des coefficients des filtres. Les approches suivies dans ces publications sont principalement basées sur méthodes *spectrales*. Certaines d'entre elles fournissent même les meilleurs résultats (pour certains ensembles particuliers de filtres) que l'on puisse obtenir par ces méthodes. Malheureusement, comme on va le voir, les estimations de régularité obtenues par des méthodes spectrales sont par nature sous-optimales du point de vue

des définitions naturelles de continuité de la fonction limite et de ses dérivées. De plus, certaines de ces méthodes d'estimation de régularité sont lourdes à mettre en œuvre.

Daubechies et Lagarias sont les premiers à adopter une approche directe (temporelle) pour l'estimation de régularité. Leur approche est basée sur des équations à deux échelles (dont nous avons parlé au chapitre 3), et non, comme ici sur le schéma d'itération de filtres (la motivation «temps-discret» n'est donc pas apparente). En conséquence, bien que leur cadre soit légèrement plus général [1], il donne lieu à des développements assez complexes, et leur algorithme d'estimation de régularité est basé sur un calcul matriciel lourd, où l'optimalité n'a été prouvée que sur des cas très particuliers.

Par contraste, l'approche suivie ici est temporelle, mais pour le *temps discret*. Elle donne lieu à des développements plus simples qui sont interprétables «physiquement», et permet non seulement de prouver, dans le cas général, l'optimalité des résultats, mais aussi, fournit un algorithme pratique, et facilement implantable, d'estimation de régularité optimale.

4.2 Analyse théorique

4.2.1 Convergence uniforme et continuité

Une difficulté technique est de définir la *convergence* du schéma itératif de subdivision précisément, sous forme de définition mathématique. Nous devons définir en effet des suites discrètes qui «tendent» vers une fonction d'une variable continue. Plusieurs définitions ont été proposées [2, 9]. Une approche typique [2] consiste à définir la convergence pour des fonctions «en escalier» dont les valeurs à $t = n2^{-j}$ sont les g_n^j (ou les h_n^j).

En fait, on peut montrer [1] que du moment que la fonction limite résultante est régulière, le type de convergence du schéma itératif est *uniforme*. Une définition possible de la convergnce uniforme, écrite pour g_n^j, est

$$\lim_{j \to \infty} \max_{|n-t2^j| \leq c} |\varphi(t) - g_n^j| = 0, \tag{4.2}$$

où c est une constante. En fait, toutes les définitions de convergence uniforme proposées dans la littérature sont équivalentes [1], et il n'y a donc pas lieu de s'inquiéter outre-mesure en distinguant plusieurs définitions. Ce type de convergence est «fort» : il implique notamment la convergence ponctuelle pour chaque valeur de t et interdit, de plus, des phénomènes locaux oscillatoires comme le célèbre phénomène de Gibbs.

Nous avons dit au chapitre précédent que le schéma itératif utilisant des filtres RIF, s'il converge, donne nécessairement une fonction limite à support compact. Grâce à cette propriété, il est facile de montrer que les propriétés de régularité de la fonction limite ne dépendent pas de la séquence utilisée pour initialiser le schéma itératif [1]. Ainsi, les propriétés de régularité de la fonction d'échelle $\varphi(t)$ et de l'ondelette associée $\psi(t)$ sont identiques. On se restreint donc dans la suite à l'étude de la convergence (uniforme) des réponses passe-bas g_n^j vers $\varphi(t)$.

Une caractérisation plus intuitive de la convergence uniforme est la combinaison de trois propriétés [1] :

$$G(z = 1) = 2, \tag{4.3}$$

$$G(z = -1) = 0, \tag{4.4}$$

et il existe un $0 < \alpha \leq 1$ tel que

$$\max_n |g_{n+1}^j - g_n^j| < c\, 2^{-j\alpha} \tag{4.5}$$

Figure 4.2: Divergence due à $G(-1) \neq 0$. Cet exemple correspond à un filtre de type Smith/Barnwell [8] utilisé dans les bancs de filtres orthogonaux. Le filtre passe-bas, de longueur 16, possède 25dB d'atténuation dans la bande passante pour une bande de transition égale, en fréquence normalisée, à 0.1 (voir chapitre 5).

où c est une constante. Les équations (4.3) et (4.4) sont des conditions de base sur le filtre passe-bas $G(z)$, nécessaires pour l'obtention de la convergence ponctuelle. Elles sont connues depuis longtemps [2, 9]. La condition (4.3) est simplement une normalisation sur $G(z)$. Si celle-ci n'est pas satisfaite, comme pour les bancs de filtres orthogonaux pour lesquels $G(1) = \sqrt{2}$, la convergence a lieu néanmoins avec le facteur de normalisation $2^{j/2}$ sur g_n^j. La condition (4.4) dit que la réponse fréquentielle de filtre passe-bas $G(e^{j\omega})$ s'annule à la fréquence de Nyquist ($\omega = \pi$). Nous y reviendrons plus loin. Pour l'instant, remarquons que cette condition est cruciale pour obtenir la convergence et la régularité : la figure 4.2 montre un exemple pour lequel $G(-1) \approx 0.05$, ce qui produit, dans les courbes g_n^j, des oscillations de faible amplitude, mais tellement rapides qu'elles interdisent toute convergence [1].

La condition (4.5) est peut-être la plus importante : elle traduit le fait que la différence entre deux valeurs consécutives de g_n^j doit tendre vers zéro, et ceci *uniformément* sur tout le graphe. Donc, aucun «saut» ou discontinuité ne doit apparaître où que ce soit dans les réponses de filtres itérées g_n^j au fur et à mesure que j augmente. Intuitivement, cela garantit la continuité de la fonction limite $\varphi(t)$. En fait, l'article [1] montre que la convergence uniforme et la continuité de $\varphi(t)$ sont deux choses *équivalentes*.

Cependant, même lorsque $\varphi(t)$ est continue, elle peut très bien appaître comme étant peu régulière, comme le montre la figure 4.3. Ceci peut s'expliquer de la manière suivante : la condition (4.5) demande que les *pentes* :

$$\delta g_n^j = \frac{g_{n+1}^j - g_n^j}{2^{-j}} \tag{4.6}$$

de la courbe «discrète» $\{g_n^j\}$, tracée en fonction de $n2^{-j}$, ne croissent pas plus rapidement (en valeur absolue) que $2^{j(1-\alpha)}$ lorsque $j \to \infty$. Ainsi, les discontinuités n'apparaissent que si les pentes augmentent au moins comme 2^j.

Même sous la condition de continuité, les pentes peuvent donc croître indéfiniment, conduisant à une courbe d'aspect fractal, comme celle de la figure 4.3. Par conséquent, si l'on veut

Figure 4.3: Un exemple de fonction limite continue. Certes, elle n'apparaît pas être effectivement très régulière. Mais malgré les apparences, on peut montrer que cette fonction est continue au sens mathématique. (Ordre de régularité au sens de Hölder : compris entre 0.19 et 0.25, au sens de Sobolev : -0.0715.)

pouvoir garantir que les courbes limites aient réellement un aspect régulier, on doit demander plus que la continuité, c'est à dire que $\varphi(t)$ possède un certain nombre N de dérivées continues, et relier cela au comportement des pentes. C'est ce qu'on va faire maintenant.

4.2.2 Dérivées

Afin de quantifier le «degré» de régularité, on va, pour l'instant, définir l'*ordre de régularité* comme étant le nombre de fois que $\varphi(t)$ est continûment dérivable ($\varphi(t) \in C^N$). Pour caractériser cet ordre de régularité en temps-discret, il suffit de noter que le rôle de la dérivée $N^{\text{ième}}$ de $\varphi(t)$ est joué, en temps-discret, par la *différence finie* d'ordre N de g_n^j [1]. La différence finie d'ordre 1 est simplement la suite des pentes δg_n^j (4.6). La transfomée en z associée à δg_n^j est donc $\Delta G^j(z) = 2^j(1 - z^{-1})G^j(z)$. En appliquant N fois l'opérateur δ, on obtient la différence finie d'ordre N, $\delta^N g_n^j$, définie par $\Delta^N G^j(z) = 2^{jN}(1 - z^{-1})^N G^j(z)$.

Plaçons nous dans le cas où g_n^j converge uniformément vers $\varphi(t)$. D'après (4.4), $G(z)$ possède au moins un zéro à $z = -1$, et $\Delta G^j(z)$ peut donc être mis sous la forme :

$$\Delta G^j(z) = (1 - z^{-2^j})F^j(z), \tag{4.7}$$

où $F^j(z)$ est définie de la même manière que $G^j(z)$, à partir de

$$F(z) = 2\frac{G(z)}{1 + z^{-1}}. \tag{4.8}$$

Ainsi, les δg_n^j suivent le même type de schéma itératif que les g_n^j. Nous pouvons donc leur appliquer les résultats du § 4.2.1 : si δg_n^j converge uniformément, alors sa fonction limite associée, que l'on peut montrer être précisément la dérivée de $\varphi(t)$ [1], est continue, et $\varphi(t) \in C^1$.

Cela se généralise aisément par récurrence [1] et l'on obtient que l'*ordre de régularité N est caractérisé par la convergence uniforme de $\delta^N g_n^j$*. L'interprétation en termes de pentes est

la même qu'en § 4.2.1, mais s'applique aux suites des pentes, des pentes des pentes, etc., ce qui requiert une évolution temporelle de g_n^j de plus en plus douce.

4.2.3 Zéros à la fréquence de Nyquist

Nous venons de voir qu'oter un zéro à $z = -1$ dans le filtre passe-bas $G(z)$ (4.8) revient à dériver la fonction limite (à ceci près que l'initialisation du schéma itératif est différente (4.7)). Puisque $F(z)$ doit avoir un autre zéro à $z = -1$ pour que δh_n^j puisse converger, une condition nécessaire pour un ordre de régularité 1 est que $G(z)$ ait au moins 2 tels zéros. Par récurrence, on obtient [1] que $G(z)$ *doit avoir au moins* $N + 1$ *zéros à* $z = -1$, c'est à dire à la fréquence de Nyquist, *pour qu'on puisse obtenir un ordre de régularité égal à* N. C'est un résultat constructif, déjà obtenu dans plusieurs articles [2, 9], qui donne une règle empirique très simple pour construire des filtres réguliers (*cf.* chapitre 5).

Cependant, il ne faut pas faire l'amalgame entre ordre de régularité et nombre de zéros à $z = -1$: certes, de tels zéros ont un effet favorable pour la régularité, puisque en ajouter N à un filtre $G(z)$ fait croître l'ordre de régularité de N [1]. C'est peut-être pour cette raison que Daubechies [2] a construit ses filtres orthogonaux «ondelettes» en imposant un maximum de zéros à $z = -1$ pour une longueur de filtre donnée : puisqu'imposer de tels zéros dans $G(z)$ revient à imposer une réponse fréquentielle «plate» autour de la fréquence de Nyquist $\omega = \pi$, les filtres de Daubechies peuvent être qualifiés de filtres «à platitude maximale» (*maximally flat*, voir chapitre 5). Cependant, il se trouve qu'ils ne sont pas à «régularité maximale» pour une longueur donnée. La raison est que l'effet bénéfique des zéros à $z = -1$ peut être entamé, voire annulé par l'effet, généralement destructeur, des *autres* zéros de $G(z)$, qui ne sont pas situés à $z = -1$. Comme l'a montré I. Daubechies elle-même [3], il est possible de construire des filtres plus réguliers, pour lesquels l'effet destructeur est moins important et compense le fait que $G(z)$ ne possède pas un maximum de zéros à $z = -1$.

En résumé, dù à l'effet destructeur pour la régularité des zéros du filtre non situés à $z = -1$, ceux situés à $z = -1$ sont nécessaires, mais pas suffisants pour garantir un ordre de régularité donné. Un exemple simple, bien qu'un peu artificiel, est $G(z) = (1 + z^{-3})^N$, pour lequel il n'y a jamais convergence, bien que $G(z)$ contienne autant de zéros à $z = -1$ que l'on désire [1].

4.2.4 Moments nuls

Pour être complet, signalons que le nombre $N + 1$ de zéros à $z = -1$ dans le filtre passe-bas $G(z)$ est égal au nombre de moments nuls pour le filtre passe-haut correspondant, et donc au nombre de moments nuls pour les ondelettes générées [3]. L'annulation de $N + 1$ moments sur une ondelette $\psi(t)$ s'exprime par $\int t^k \psi(t)\, dt = 0$ pour $k = 0, \ldots, N$. Dans son travail sur les «coiflettes» [3], I. Daubechies impose de plus un certain nombre de moments nuls (pour $k > 0$) sur la fonction d'échelle $\varphi(t)$ générée par $G(z)$, ce qui équivaut à imposer des zéros à $z = 1$ dans $2 - G(z)$. Ces propriétés sont étroitement liées aux décompositions multi-résolution de polynômes, et pourraient avoir un intérêt pour quelques applications [3]. Néanmoins, la propriété de moments nuls, en soi, semble être d'un intérêt purement anecdotique en codage, bien qu'elle soit une condition nécessaire pour la régularité. Nous n'y reviendrons donc pas explicitement dans la suite.

4.3 Estimations de régularité

Afin de comprendre le rôle de la régularité dans les applications, on a besoin d'estimations précises. On aimerait donc disposer d'une boîte noire, qui, étant donné un choix de filtres, nous donne l'ordre de régularité exact des fonctions limites correspondantes. Nous venons de voir qu'un ordre de régularité N nécessite au moins $(N+1)$ zéros à $z=-1$ dans le filtre passe-bas $G(z)$. Par conséquent, l'ordre de régularité est borné (strictement) par le nombre total de tels zéros dans $G(z)$. Mais cette borne supérieure est en général assez faible, sauf pour des cas particuliers comme $G(z)=(1+z^{-1})^{N+1}$ [1]. Par conséquent, le nombre de zéros à $z=-1$ ne donne pas une bonne idée de l'ordre de régularité du filtre.

Afin de caractériser et d'estimer finement la régularité, il est nécessaire de considérer l'«effet destructeur» des autres zéros (mentionné ci-dessus) qui, typiquement, détruit 80% de la régularité apportée par les zéros à $z=-1$. On a donc besoin de mesurer la quantité de régularité perdue par les zéros de $G(z)$ non situés en $z=-1$.

4.3.1 Régularité de Sobolev et de Hölder

Nous commençons par étendre la définition d'ordre de régularité (le nombre de dérivées continues de $\varphi(t)$) à des valeurs réelles quelconques. Il y a plusieurs façon de faire, les plus communes utilisant les espaces fonctionnels de Sobolev et de Hölder. Le cadre qui en résultera est plus commode pour quantifier précisément la régularité et pour trouver des algorithmes d'estimation de régularité optimale.

La définition de la régularité *au sens de Sobolev* vient d'une approche spectrale, qui considère la régularité comme une localisation spectrale : on dira que $\varphi(t)$ est régulière d'ordre de régularité r au sens de Sobolev[2] si l'on a :

$$\int |\omega|^{2r+1}|\Phi(\omega)|^2\,d\omega<\infty. \tag{4.9}$$

où $\Phi(\omega)$ est la transformée de Fourier de $\varphi(t)$. Il est bien connu que si $\varphi(t)$ a un ordre de régularité au sens de Sobolev $r>N$, où N est entier, alors elle possède bien N dérivées continues.

Cette définition spectrale, parmi d'autres [1], a été jusqu'à présent la plus populaire dans la littérature «ondelettes» [2]–[7]. En effet, on peut l'appréhender facilement par des estimations simples sur le module de la réponse fréquentielle du filtre passe-bas $G(z)$, puisque l'on a [2] :

$$\lim_{i\to\infty} G^i(e^{j\omega 2^{-i}})=\Phi(\omega). \tag{4.10}$$

Il faut noter aussi que l'effet bénéfique des zéros à $z=-1$ du filtre $G(z)$, a, dans ce cadre, une interprétation spectrale naturelle : si la réponse fréquentielle $|G(e^{j\omega})|$ est «plate» autour de la fréquence de Nyquist, alors les spectres répétés dans le terme de gauche de (4.10) seront atténuées, faisant de $\Phi(\omega)$ une fonction localisée, et donc $\varphi(t)$ une fonction régulière au sens de Sobolev.

Cependant, la définition de Sobolev souffre de limitations. Tout d'abord, c'est une définition fréquentielle, qui masque l'effet de la régularité sur la forme d'onde temporelle de $\varphi(t)$. Ensuite, seul le *module* de $G(e^{j\omega})$ est pris en compte : l'information sur la phase est ignorée. Enfin, la figure 4.3 a déjà montré un exemple pour lequel le *meilleur* ordre de régularité au sens de Sobolev est *négatif*, alors qu'on peut montrer par ailleurs que la fonction est continue.

[2]Pour être précis, on a utilisé l'espace de Sobolev d'exposant s, H^s, comme espace des fonctions $\varphi(t)$ qui ont pour ordre de régularité de Sobolev $r=s-1/2$.

Le définition de la régularité *au sens de Hölder*, introduite récemment pour les ondelettes [4], permet d'éviter ces inconvénients. C'est une définition temporelle, qui se révèle être appropriée pour la suite. L'idée est de regarder les pentes infinitésimales de $\varphi(t)$ en temps, $|\varphi(t+h) - \varphi(t)|/|h|$, et de contrôler la façon dont ces pentes croissent lorsque $h \to 0$. On dira que $\varphi(t)$ (à support compact) est d'ordre de régularité α au sens de Hölder, pour $0 < \alpha \leq 1$, si, pour tout t et h,

$$|\varphi(t+h) - \varphi(t)| < c\,|h|^{\alpha} \tag{4.11}$$

où c est une constante indépendante de t et de h. Cette définition implique clairement la continuité, mais n'est pas utilisable si $\alpha > 1$, puisque cela voudrait dire que la dérivée de $\varphi(t)$ s'annule identiquement, d'où $\varphi(t) \equiv 0$. Par conséquent, pour des ordres de régularité plus élevées $r = N + \alpha$, où $N = 1, 2, \ldots$ et $0 < \alpha \leq 1$, on utilise la même définition, mais sur la dérivée $N^{\text{ième}}$ de $\varphi(t)$ [1].

Cette définition est évidemment compatible avec les ordres de régularité entiers N, mis à part le fait qu'un ordre de régularité $r = N$ n'implique pas que la $N^{\text{ième}}$ dérivée soit continue : elle ne le sera que si $r = N+\varepsilon$, où $\varepsilon > 0$ est arbitrairement petit. Pour simplifier la présentation, nous laisserons de côté le ε dans la suite et considérons systématiquement les ordres de régularité à ε près (voir [1] pour un traitement plus précis). Par contre, on peut montrer [1] que la continuité de la fonction limite $\varphi(t)$ implique (4.11). En conséquence, on a l'équivalence suivante : $\varphi(t)$ possède N dérivées continues si et seulement si l'ordre de régularité de $\varphi(t)$ est supérieur à N. Cette propriété, comme on l'a vu, n'est pas vraie pour la régularité au sens de Sobolev. Elle montre que la définition de Hölder conduit effectivement à des résultats optimaux du point de vue de la continuité de $\varphi(t)$ et de celle de ses dérivées.

Notons que la différence entre les deux régularités de Sobolev, r_S, et de Hölder, r_H ne dépend que de la *phase* du filtre $G(z)$, et peut en fait être estimée [1] : on a, pour des fonctions à support compact :

$$r_s \leq r_H \leq r_s + 1/2 \tag{4.12}$$

La régularité de Sobolev donne donc à la fois une borne inférieure et supérieure de la régularité hölderienne. Ceci sera illustré plus loin.

4.3.2 Caractérisation discrète de la régularité

En utilisant la définition de Hölder, on va maintenant traduire la propriété de régularité par une condition équivalente sur les suites discrètes g_n^j. La définition «temps-discret» de régularité qui en résulte est une étape préliminaire au calcul d'une estimée optimale.

Nous avons vu en § 4.2.1 que la continuité est équivalente aux conditions (4.3)–(4.5), et que le nombre α dans (4.5) donne la loi de croissance des pentes δg_n^j : celles-ci ne croissent pas plus vite que $2^{j(1-\alpha)}$. Cette contrainte est d'autant plus forte et impose des évolutions temporelles de g_n^j d'autant plus douces que α augmente de 0 à 1. En fait, α est précisément l'ordre de régularité de $\varphi(t)$ au sens de Hölder lorsque $\alpha < 1$ [1] : il y a équivalence entre (4.5) et (4.11) dans ce cas (ce qui, à y regarder de plus près, n'est pas surprenant, puisque (4.5) n'est finalement qu'une version «discrétisée» de (4.11)).

De ce fait, la régularité hölderienne a une interprétation graphique tout-à-fait naturelle en temps-discret : si les «pentes» de la courbe discrète g_n^j croissent comme $2^{j(1-\alpha)}$, $0 < \alpha < 1$, la fonction limite a pour ordre de régularité α et n'est pas plus régulière que cela. Pour des ordres de régularité plus élevés, on aura forcément $\alpha = 1$ dans (4.5) [1]. Il suffit alors de considérer les dérivées de $\varphi(t)$ associées aux différences finies de g_n^j [1] : l'extension est immédiate.

4.3.3 Vitesse de convergence

La régularité hölderienne nous donne aussi une indication intéressante sur la vitesse de convergence des courbes g_n^j vers la fonction limite $\varphi(t)$. Cette vitesse est exponentielle : on a, pour $0 < \alpha \leq 1$,

$$\max_{|n-t2^j| \leq c'} |\varphi(t) - g_n^j| \leq c\, 2^{-j\alpha}. \tag{4.13}$$

En d'autres termes, la vitesse de convergence croît avec la régularité. Ceci s'étend immédiatement à la vitesse de convergence des pentes pour des ordres de régularité plus élevées. En pratique, la convergence est très rapide (comme on peut le remarquer d'après les figures 3.2 et figure 3.3 (a) du chapitre précédent. Pour des filtres suffisamment réguliers, on ne peut plus distinguer les courbes successives après quelques itérations. Ceci justifie *a posteriori* l'étude de la régularité comme définie sur la fonction limite, même si le nombre d'itérations est limité en pratique.

4.3.4 Estimation optimale

Armés des caractérisations temps-discret de la régularité hölderienne, nous pouvons maintenant nous attaquer au problème de l'estimation optimale de l'ordre de régularité associé à un filtre passe-bas donné. Puisque nous avons parlé d'ordre de régularité à ε près, où $\varepsilon > 0$ est arbitrairement petit, nous définirons l'optimalité également à ε près : une estimée r de régularité sera donc dite *optimale* si on peut montrer que $\varphi(t)$ est au moins régulière d'ordre $r - \varepsilon$ et n'est pas plus régulière qu'un ordre $r + \varepsilon$.

Un fait remarquable, qui simplifie grandement la suite du développement, est que la caractérisation temps-discret de la régularité de Hölder $N+\alpha$ est équivalente à la même condition dans laquelle N et $-\alpha$ sont simultanément augmentés d'un entier positif [1] (on ne peut pas en dire autant, bien sûr, de la définition de Hölder initiale). Cela signifie qu'on peut l'étendre à des valeurs négatives de α (tant que $N + \alpha$ reste positif), autrement dit que l'on peut se permettre de «différencier trop de fois» et d'avoir quand même une caractérisation optimale de régularité. (Cette propriété permet d'ailleurs de définir des régularités hölderiennes négatives !).

Maintenant, supposons que $G(z)$ ait exactement K zéros à $z = -1$. On sait, d'après § 4.2.3, que l'ordre de régularité de Hölder de la fonction limite $\varphi(t)$ est inférieure ou égale à K. De plus, on peut montrer [1] que cette limite serait atteinte si $G(z)$ ne possédait que des zéros à $z = -1$ ($\varphi(t)$ serait alors une fonction spline $K - 1$ continûment dérivable, d'ordre de régularité hölderienne égale à K). Or, en prenant la caractérisation temps-discret pour $N = K - 1$, on obtient un ordre de régularité (exact) égal à $K-1+\alpha$, où α peut être négatif. On a donc identifié la quantité exacte de régularité perdue par l'effet «destructeur» (§ 4.2.3) des zéros de $G(z)$ non situés à $z = -1$ comme étant égale à $1 - \alpha$.

Pour obtenir une estimation optimale de régularité à partir d'un filtre quelconque, il suffit donc de pouvoir estimer exactement α dans des conditions comme (4.5). Ceci pose problème, car de telles conditions doivent être vérifiées pour un nombre infini de valeurs de j et pour une constante c que l'on ne connait pas. Il est heureusement possible de réduire cette tâche à un algorithme ne demandant qu'un nombre fini d'opérations. Le lecteur trouvera tous les détails dans [1].

Nous nous bornons ici à donner la forme finale de l'algorithme d'estimation optimale de régularité [1]. Cet algorithme ne sera en fait qu'asymptotiquement optimal, mais donnera toujours des estimations (bornes inférieures) de régularité :

Algorithme 1 *Soit $K > 0$ le nombre exact de zéros à $z = -1$ dans le filtre RIF passe-bas $G(z)$, qui est normalisé de telle façon que $G(1) = 2$.*

1. *Si $G(z)$ n'a pas d'autre zéro, l'ordre exact de régularité de Hölder est K.*

2. *Afin d'estimer la quantité de régularité perdue par la présence d'autres zéros dans $G(z)$, considérons $F(z)$, défini par :*

$$G(z) = \left(\frac{1+z^{-1}}{2}\right)^K F(z)$$

3. *Soit $j > 0$ un entier quelconque et considérons*

$$F^j(z) = F(z)\,F(z^2)\cdots F(z^{2^{j-1}}) \tag{4.14}$$

associé à la suite f_n^j.

4. *Alors, la fonction limite $\varphi(t)$ associée à g_n^j est régulière, au sens de Hölder, d'ordre au moins égal à*

$$K - 1 + \frac{1}{j}\log_2 \max_{0\le n<2^j} \sum_k |f^j_{n+2^j k}|. \tag{4.15}$$

De plus, lorsque $j \to \infty$, cette estimée converge (au pire en $1/j$) vers l'ordre de régularité optimale de $\varphi(t)$.

L'estimée donnée par cet algorithme s'améliore (globalement) lorsque j croît. En pratique, la convergence vers l'ordre de régularité optimale est très rapide (voir [1, Fig. 4]). (En général, après une vingtaine d'itérations, l'erreur commise n'affecte plus que la deuxième ou troisième décimale).

De plus, il est très facile d'implanter cet algorithme de façon récursive : en effet, les suites décimées $f^j_{n+2^j k}$ s'obtiennent comme sorties du graphe décrit à la figure 4.4 (a), lequel, par la relation (4.14), est équivalent à celui de la figure 4.4 (b). En notant $n = \varepsilon_j \cdots \varepsilon_1$ l'écriture binaire du nombre n, on peut transformer ce dernier graphe comme montré à la figure 4.4 (c), qui, à son tour, peut être redessiné sous forme d'arbre binaire (figure 4.4 (d)). L'algorithme 1 peut donc être implanté par appels récursifs à une même routine de filtrage où on calcule séparément les sorties d'indices pairs et impairs.

Comme le montre l'article [1], il est possible, à partir de l'algorithme 1, de retrouver les résultats «matriciels» de Daubechies et Lagarias [4]. On peut d'ailleurs retrouver un grand nombre d'estimations de régularité publiées dans des travaux antérieurs, y compris, comme on va le voir, une estimation optimale de la régularité au sens de Sobolev.

4.3.5 Une borne supérieure fine

Comme l'algorithme 1 revient à parcourir un arbre binaire complet, la charge de calcul nécessaire à la réalisation de l'algorithme est exponentiel en j : le nombre total d'opérations est environ multipliée par deux à chaque itération. Maintenant, supposons que l'on ne prenne en compte, dans la figure 4.4 (d), que les deux chemins correspondant aux branches les plus hautes d'une part, et les plus basses d'autre part. Cela revient à ne calculer le maximum intervenant dans (4.15) que pour les valeurs $n = 0$ et $2^j - 1$. L'avantage est qu'on réduit singulièrement la charge de calcul, qui devient linéaire en j : nous avons un algorithme bien plus rapide. Le prix à payer est que la valeur de l'estimée ne donne, lorsque $j \to \infty$, qu'une borne supérieure sur la régularité hölderienne. Néanmoins, cette borne supérieure se trouve être fine dans beaucoup

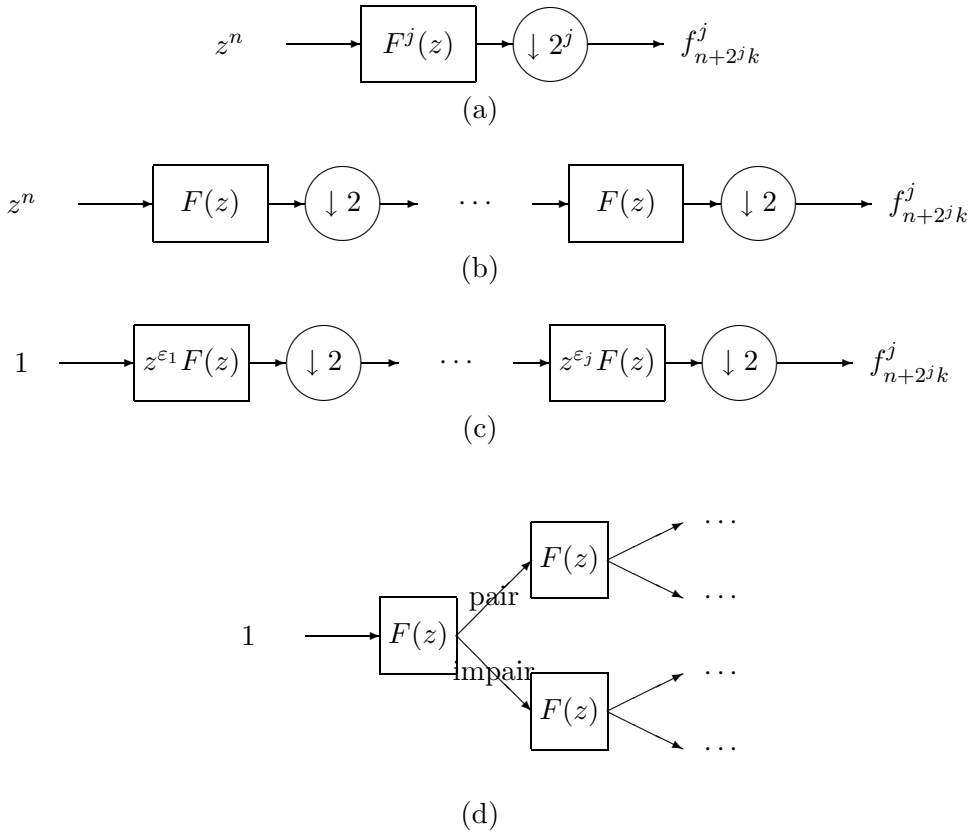


Figure 4.4: Implantation de l'algorithme 1.

L	[2]	Alg. 1	Alg. 2
4	0.499	0.550	0.550
6	0.915	1.083	1.088
8	1.275	1.606	1.617
10	1.596	1.942	1.969
12	1.888	2.163	2.189
14	2.158	2.434	2.461
16	2.414	2.735	2.761
18	2.661	3.043	3.074
20	2.902	3.309	3.382

Table 4.1: Estimations de régularité pour les ondelettes de Daubechies. On compare ici, pour diverses longueurs de filtres L, les estimations obtenues dans [2], avec celles provenant de l'algorithme 1 (après 20 itérations) et les bornes supérieures fournies par l'algorithme 2.

d'exemples : d'ailleurs, la méthode de base proposée par Daubechies et Lagarias [4] ne donne une estimation optimale que dans le cas où cette borne supérieure coïncide avec l'ordre exact de régularité [1] (comme c'est par exemple le cas pour les ondelettes de Daubechies de longueur ≤ 8).

De plus, l'algorithme permettant de calculer cette borne supérieure se simplifie grandement, puisqu'il se réduit au calcul d'un rayon spectral de matrice [1] :

Algorithme 2 (Borne Supérieure.) *Avec les mêmes notations que l'algorithme 1, définissons la matrice carrée* $\mathbf{F} = (\mathbf{F}_{i,j})$, $0 \leq i,j \leq M-3$ *où* M *est la longueur du filtre* $F(z)$, *par :*

$$\mathbf{F}_{i,j} = f_{2i-j+1}$$

L'ordre de régularité de $G(z)$ *ne dépasse pas* $K - 1 - \log_2 \max(|f_0|, |f_{M-1}|, \rho)$, *où* ρ *est le rayon spectral (maximum des modules des valeurs propres) de la matrice* $\mathbf{F}$.

Cet algorithme est mis en œuvre dans la fonction MATLAB `reg()` décrit à l'appendice C.

La table 4.1 donne les estimations trouvées pour les algorithmes 1 et 2, appliqués aux filtres à phase minimum de Daubechies [2]. Il est clair que ces algorithmes, qui sont utilisables en pratique, donnent des encadrements suffisemment précis de l'ordre de régularité (au sens de Hölder) optimal, pour pouvoir être utilisés comme outils lorsqu'on veut mesurer le rôle de la régularité dans le cadre d'une application donnée.

4.4 Cas particuliers et extensions

4.4.1 Cas symétrique

L'algorithme 1 se simplifie notablement dans le cas où les filtres utilisés sont à phase «strictement» linéaire, c'est à dire à phase linéaire, sans sauts de phase de π. Cette simplification est traitée dans [1]. Nous exposons ici l'algorithme obtenu dans le cas d'application qui nous intéresse le plus, à savoir celui d'un banc de filtres orthogonaux (*cf.* chapitre 2). Cet algorithme fournit, dans ce cas, une estimation optimale de la régularité, *au sens de Sobolev*, des ondelettes orthogonales associées .

Rappelons (§ 2.1.4) que dans le cas orthogonal, un seul filtre, $G(z)$ est à déterminer, et que la condition de reconstruction parfaite fait intervenir le filtre «produit» $P(z) = G(z)\tilde{G}(z)$, dont on vérifie qu'il est à phase strictement linéaire. On applique alors la version simplifiée de l'algorithme 1 (décrit dans [1]) sur $P(z)$. Intuitivement, puisqu'il n'y a «plus de phase», cet algorithme va nous donner un ordre de régularité optimal au sens de Hölder, qui va correspondre à l'ordre de régularité optimal, au sens de Sobolev, pour $G(z)$. En effet, on a vu (§ 4.3.1) que les deux définitions ne différaient que par la phase du filtre. Cette intuition est effectivement confirmée dans [1]. De plus, comme l'algorithme simplifié ne nécessite que le calcul d'un rayon spectral de matrice, il est particulièrement simple à mettre en œuvre.

Algorithme 3 (Régularité de Sobolev optimale) *Avec les notations utilisées dans l'algorithme 1, définissons*

$$Q(z) = F(z)F(z^{-1})z^{-(M-1)}$$

où M est la longueur du filtre $F(z)$, et formons la matrice carrée $\mathbf{A} = (\mathbf{A}_{i,j})$, $0 \leq i,j \leq (M-3)/2$, définie par

$$\mathbf{A}_{i,j} = \begin{cases} a_{M-1-2i} & \text{if } j = 0, \\ a_{M-1-2i-j} + a_{M-1-2i+j} & \text{sinon.} \end{cases}$$

La régularité au sens de Sobolev optimale est $K - \frac{1}{2}\log_2 \rho$, où ρ est la rayon spectral de la matrice $\mathbf{A}$.

La fonction MATLAB correspondante, `sobreg()`, est décrite à l'appendice C. Il se trouve que cette méthode coïncide avec une autre, trouvée indépendamment et simultanément par Cohen et Daubechies [5], qui se base sur la théorie de Littlewood/Paley.

La combinaison de l'algorithme 1 et 3 peut être mise à profit pour déterminer la quantité de régularité effectivement due à la phase du filtre, qui est la différence entre la régularité de Hölder et celle de Sobolev. Enfin, on a mentionné que la régularité de Sobolev donnait à la fois une borne inférieure et supérieure pour la régularité au sens de Hölder (relations (4.12)). La figure 4.5 montre ces bornes, en comparaison avec celles, beaucoup plus précises, fournies par les algorithmes 1 et 2.

4.4.2 Régularité locale et "fractale"

Jusqu'a présent, toutes les définitions de régularité que l'on a vues sont «globales», c'est à dire qu'on estime le «pire» (minimum) ordre de régularité que l'on observe sur tous les points d'une courbe. Mais il est également possible d'estimer un ordre de régularité hölderienne «local», qui reflète le comportement des pentes infinitésimales de $\varphi(t)$ autour d'un point donné t.

On peut penser que l'étude de la régularité locale nous permetterait de quantifier plus précisément l'«allure» de la courbe limite obtenue, afin d'obtenir une nouvelle mesure qui distinguerait deux courbes ayant la même régularité globale mais, par le jeu des comportements locaux, ont des allures très différentes. Autrement dit, on s'approcherait plus de des considérations subjectives sur l'allure des fonctions limites obtenues.

Dans leur article [4], Daubechies et Lagarias ont étendu leurs résultats au cas de la régularité locale. La même extension est possible dans l'approche présentée ici. (Les ordres α dépendent alors de t, et on considère $|g^j_{n+1} - g^j_n|$ lorsque n est, par exemple, le plus proche entier de $2^j t$, au lieu de $\max_n |g^j_{n+1} - g^j_n|$.) Cependant, elle conduit à des résultats tellement similaires que je me bornerai ici à rappeler brièvement les résultats de [4].

Figure 4.5: Bornes inférieures et supérieures de régularité. En trait pointillé, bornes fournies par l'algorithme 3 (estimations optimales au sens de Sobolev). En trait plein, bornes fournies par les algorithmes 1 et 2 (estimations de la régularité hölderienne).

Considérons le cas $N = 1$ (ordre de régularité compris entre 0 et 1). Le cas où les ordres de régularité sont supérieurs à 1 peut se traiter de la même manière en considérant des dérivées. Dû à la structure particulière du schéma d'itération des filtres, la régularité à l'endroit t ne dépend que de sa partie décimale, et on peut supposer $0 \leq t \leq 1$. Décomposons t en base deux, $t = \sum_{j \leq 1} \varepsilon_j 2^{-j}$, où $\varepsilon_i = 0$ ou 1. Le nombre n que l'on doit prendre pour calculer la somme apparaissant en (4.15), à savoir $\sum_k |f^j_{n+2^j k}|$, est donc $n = \varepsilon_1 \cdots \varepsilon_j$, écrit en base deux, ce qui correspond à un chemin particulier dans l'arbre binaire de la figure 4.4 (d). Rappelons que dans le cas de la régularité globale, la maximum de cette somme sur n tend, lorsque $j \to \infty$, vers $2^{-\alpha}$ où α est la régularité (globale) optimale. Grâce à la formulation matricielle de Daubechies et Lagarias [4], on peut plus généralement, borner cette somme par une expression de la forme

$$c\, 2^{-(\sum_i \varepsilon_i)\beta} 2^{-(j - \sum_i \varepsilon_i)\gamma}. \tag{4.16}$$

Il faut noter que (4.16) n'est qu'une majoration, qui va donc conduire à des estimations sous-optimales.

En faisant tendre j vers l'infini, Daubechies et Lagarias obtiennent un ordre de régularité locale égal à $\lambda\beta + (1 - \lambda)\gamma$ autour de t, si t a la propriété particulière que la proportion de «1» dans son développement en base deux converge vers λ. Puisque les nombres correspondant à $\lambda = 1/2$ possèdent un sous-ensemble de mesure 1, ceci permet de dire que la régularité est *presque partout* égale à $(\beta + \lambda)/2$, ce qui est supérieur ou égal à la régularité globale [4]. On a donc ici une nouvelle notion de régularité globale (presque partout), qu'on pourrait étudier dans le cadre de cette thèse. Malheureusement, en revenant à l'approche temps-discret, on s'apercoit que de telles valeurs de t ne sont obtenues qu'à l'infini, il n'est donc pas justifié que la notion de régularité «presque partout» ait un intérêt particulier.

En fait, dans le schéma d'itération des filtres, on calcule les valeurs des courbes aux points

dyadiques $t = n2^{-j}$ uniquement, et les valeurs de $\varphi(t)$ ne sont accessibles qu'en ces points [1]. Il se trouve que ces points dyadiques sont précisément très particuliers du point de vue de la régularité locale [4], car on peut approcher un point dyadique de deux façons (de la gauche ou de la droite), où les expansions binaires nous donnent respectivement β et γ comme ordres de régularité à gauche et à droite de t. Le minimum des deux correspond souvent à l'ordre de régularité global, alors que l'autre correspond à une régularité «maximale» des points de la courbe.

Toutes ces définitions de régularité locale reposent donc principalement sur deux valeurs, l'une correspondant à la régularité minimale (globale) et l'autre à la régularité maximale que l'on peut atteindre sur la courbe limite. Du point de vue de l'aspect visuel des courbes, il n'est pas évident de réaliser si ces notions ont un intérêt particulier.

C'est pourquoi je propose une autre définition de régularité, qui est liée à la dimension fractale (au sens de Hausdorff) de la courbe limite. Il s'agit d'une régularité locale moyennée : Au lieu de $\max_n |g^j_{n+1} - g^j_n|$, on considère la «moyenne» $2^{-j} \sum_n |g^j_{n+1} - g^j_n|$ et on étudie son comportement lorsque $j \to \infty$. En fait, on peut montrer que, par exemple, les approximations en escalier des réponses itérées g^j_n tracées en fonction de $n2^{-j}$, ont une longueur totale égale, à une constante près, à $L_j = \sum_n |g^j_{n+1} - g^j_n|$.

La dimension fractale de Hausdorff de la courbe limite se définit alors par la formule

$$d = \frac{\log_2(2^j L_j)}{j}$$

et la régularité locale moyennée $\bar{\alpha}$ peut donc se définir par

$$\bar{\alpha} = 2 - d.$$

On peut montrer facilement, grâce à l'inégalité $|g^j_{n+1} - g^j_n| \leq c \max_n \sum_k |f^j_{n+2^j k}|$ [1], que $\bar{\alpha} \geq \alpha$, où α est la régularité globale. Ainsi, si $\varphi(t)$ est continûment dérivable, sa dimension fractale est nécessairement égale à 1 (ce qui correspond bien à l'intuition). Par contre, si $\varphi(t)$ a un ordre de régularité optimal $\alpha = \varepsilon$ (où ε est très petit), la dimension fractale de $\varphi(t)$ peut s'approcher de 2 : la courbe deviendrait tellement oscillante qu'elle «couvrirait presque tout le plan».

Grâce au majorant (4.16) établi pour la régularité locale, on peut également trouver un autre minorant de $\bar{\alpha}$. On a :

$$\begin{aligned} L_j &\leq c \sum_{n=\varepsilon_1 \cdots \varepsilon_j} 2^{-(\sum_i \varepsilon_i)\beta} 2^{-(j-\sum_i \varepsilon_i)\gamma} \\ &\leq c \sum_l \mathrm{C}^l_j 2^{-l\beta} 2^{-(j-l)\gamma} \\ &\leq c\,(2^{-\beta} + 2^{-\gamma})^j \end{aligned}$$

Je conjecture qu'on a en fait équivalence, et donc que la régularité locale moyennée s'exprime en fonction de β et γ par la formule

$$2^{-\bar{\alpha}} = \frac{2^{-\beta} + 2^{-\gamma}}{2}.$$

ce qui permetterait de trouver des algorithmes d'estimation précise de $\bar{\alpha}$.

Nous nous trouvons donc devant un certain nombre de notions différentes de régularité. Le problème de savoir laquelle est réellement utile pour les applications reste ouvert. Dans la suite de la thèse, on se consacrera uniquement aux notions de régularité globale, que l'on maîtrise mieux.

4.4.3 Autres extensions possibles

Il faut tout d'abord noter que le processus d'itération de filtres considéré dans ce chapitre est identique à celui de *schéma de subdivision binaire* étudié dans de nombreuses publications de mathématiques appliquées aux constructions de courbes ou surfaces assistées par ordinateur (*Computer-aided geometric design*) [9]. Certains des résultats de ces articles s'appliquent également au problème de régularité posé par la théorie des ondelettes. (L'article [1] a d'ailleurs été écrit dans ce cadre.) Il est possible d'étendre les travaux faits au cas des surfaces. Cela a déjà été fait partiellement (pour les ordres de régularité entiers) dans [9].

Il est possible également de suivre l'approche «discrète» de la régularité pour des bancs de filtres bidimensionnels non-séparables [10], comme par exemple ceux associées au schéma «en quinconce» (deux sous-bandes) [11]. La difficulté majeure, dans ce cas, est l'absence d'une règle de factorisation des transformées en z bidimensionnelles. Cependant, on arrive à des algorithmes d'estimation de régularité ressemblant à l'algorithme 1, qu'il reste néanmoins à simplifier grandement si on veut les voir un jour utilisé en pratique. Cohen et Daubechies [5] ont par ailleurs, en suivant l'approche de Daubechies et Lagarias [4], trouvé des algorithmes similaires.

Par contre, l'étude présentée ici s'étend très facilement au cas des bancs de filtres à plusieurs sous-bandes, et plus généralement au cas des bancs de filtres «rationnels» [12], ou la régularité ne sert pas qu'à obtenir des fonctions de base «adoucies».

4.5 Conclusion

Dans ce chapitre, on a donné une caractérisation complète de la notion «temps-continu» de régularité en temps discret, sur les coefficients des filtres. L'approche discrète suivie ici est efficace (on obtient des résultats optimaux) et générale (on retrouve un certain nombre d'estimations de régularité connues). Comparée à une autre approche temporelle, celle de Daubechies et Lagarias [4], elle est sans doute également plus accessible.

Cette approche a permis de fournir un certain nombre d'algorithmes d'estimations de régularité (au sens de Hölder ou de Sobolev) optimales ou asymptotiquement optimales. Ces algorithmes sont suffisamment précis pour être utilisés avec profit comme des outils permettant de quantifier précisément l'effet de la propriété de régularité dans les applications telles que le codage.

Cependant, il reste à intégrer cette connaissance de la régularité sur le calcul des filtres utilisés dans un schéma de codage, afin de comparer la régularité à d'autres propriétés des filtres, comme la sélectivité en fréquence. Ceci fait l'objet du chapitre suivant.

Bibliographie

[1] O. Rioul, "Simple regularity criteria for subdivision schemes," *SIAM J. Math. Anal.*, vol. 23, no. 6, pp. 1544–1576, Nov. 1992.

[2] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math*, vol. XLI, no. 7, pp. 909–996, 1988.

[3] I. Daubechies, "Orthonormal bases of compactly supported wavelets II. Variations on a theme," *SIAM J. Math. Anal.*, vol. 24, 1993. To appear.

[4] I. Daubechies and J. C. Lagarias, "Two-scale difference equations I. Local regularity, infinite products of matrices and fractals," *SIAM J. Math. Anal.*, vol. 23, pp. 1031–1079, 1992.

[5] A. Cohen and I. Daubechies, "Non-separable bidimensional wavelet bases," *Rev. Mat. Iberoamericana*. To appear.

[6] L. Villemoes, "Sobolev regularity of wavelets and stability of iterated filter banks," in *Proc. Int. Colloqueum "Wavelets and Applications"*, (Toulouse, France), June 1992.

[7] H. Volkmer, "On the regularity of wavelets," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 872–876, Mar. 1992.

[8] M. J. T. Smith and T. P. Barnwell, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 3, pp. 434–441, June 1986.

[9] N. Dyn, "Subdivision schemes in CADG," in *Advances in numerical analysis II : Wavelets, subdivision algorithms and radial functions*, W. A. Light, Ed., London : Oxford University Press, 1991, pp. 36–104.

[10] O. Rioul, "A discrete approach to regularity of 1D or 2D wavelets," in *Proc. 1992 Digital Signal Processing Workshop*, (Utica, IL), Sept. 1992, pp. 1.9.1–2.

[11] M. Vetterli, "Multi-dimensional subband coding : Some theory and algorithms," *Signal Processing*, vol. 6, no. 2, pp. 97–112, Feb. 1984.

[12] T. Blu and O. Rioul, "Wavelet regularity of iterated filter banks with rational sampling changes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Minneapolis, MN), Apr. 1993. To appear.

Chapitre 5

Calcul de bancs de filtres réguliers

Given any problem containing n equations, there will always be $n+1$ unknowns.
— SAFU'S 1ST EQUATION
Badness comes in waves.
— SAFU'S 4TH EQUATION

Ce chapitre est consacré au calcul de bancs de filtres RIF à reconstruction parfaite, générant des ondelettes régulières, en vue d'application au codage. D'après le chapitre précédent, la propriété de régularité peut être vue comme une nouvelle contrainte sur les filtres. Le problème du calcul de nouveaux filtres «réguliers» est crucial dans le cadre de cette thèse, qui s'est donné comme objectif de pouvoir mesurer les effets de la régularité pour des applications de compression d'images. En effet, il faut réussir à intégrer la régularité avec un certain nombre d'autres critères ou paramètres, plus classiques, que l'on considère généralement comme utiles pour le codage :

- Selectivité en fréquence (spécification d'un gabarit).
- Filtres (plus ou moins) orthogonaux.
- Filtres (plus ou moins, ou exactement) à phase linéaire.
- Nombre de coefficients des filtres (longueurs).

A cette liste, on pourrait rajouter d'autres critères, par exemple sur les formes des réponses impulsionnelles des filtres. Chaque paramètre correspond en général à des arguments particuliers pour une application donnée :

Ainsi, pour le codage de source utilisant une transformée, il est important que la transformée décorrèle le plus possible les différentes sorties entre elles, afin qu'elles puissent être codées facilement et indépendamment les unes des autres. Pour des filtres idéaux (qui sont de longueur infinie), la décorrélation est totale. La sélectivité en fréquence se justifie donc par le fait qu'on approxime des filtres idéaux par des filtres RIF. Une façon classique pour ce faire est d'imposer un gabarit fréquentiel (approximation au sens de Tchebichev). Dans ce chapitre, nous parlerons aussi d'approximation quadratique.

L'orthogonalité est également perçue par de nombreux chercheurs comme une propriété essentielle pour le codage [2]. Ceci vient probablement du fait que pour les signaux stationnaires, la transformée de Karhunen-Lœve [3] (dépendant du signal), qui réalise la décorrélation parfaite des sorties, est orthogonale (ses fonctions de base sont des vecteurs propres de la matrice

symétrique d'autocorrélation du signal). Néanmoins, l'orthogonalité d'un banc de filtres ne garantit pas, en général, la décorrélation entre bandes (sauf si l'entrée est un bruit blanc !).

De plus, il est impossible de réaliser un banc de filtres orthogonaux à deux sous-bandes pour lequel les filtres soient à phase linéaire (sauf cas trivial $G(z) = (1 + z^{-1})/\sqrt{2}$, qui correspond à une décomposition de Haar [4]). Or, la phase linéaire est également considérée comme importante pour le codage d'images : en effet, l'œil est très sensible à la phase présente dans une image. Cependant, rien ne dit que cette sensibilité reste cruciale dans le domaine transformé.

Enfin, on demande généralement des filtres relativement courts (par exemple, de longueur inférieure à 16) pour conserver une charge de calcul raisonnable dans un schéma de compression d'images. Peut-être aussi est-ce pour éviter des artéfacts aux contours présents dans une image (*ringing effects*), qui seraient dus aux oscillations présentes dans les réponses des filtres, dont l'étendue augmente avec la longueur.

En résumé, l'utilité chaque propriété se base sur un certain nombre de «croyances communes», qui restent parfois hypothétiques. Il en est de même pour la régularité, pour laquelle nous avons donné des arguments de type «psychovisuel» au chapitre précédent. De plus, certaines des propriétés de filtres sont, comme on va le voir, contradictoires. Afin de se donner les moyens de confirmer ou d'infirmer la validité des argumentations que l'on vient d'exposer, il est nécessaire de construire des familles de filtres dont les caractéristiques réalisent les meilleurs *compromis* possibles entre les différentes propriétés énoncées.

Malheureusement, les familles de filtres «réguliers» disponibles dans la littérature [5]–[8] – dont la plus célèbre est sans doute la famille des filtres de Daubechies [5, 6] – sont loin d'offrir de tels compromis. En effet, à l'intérieur d'une famille donnée comme celle de Daubechies, les contraintes sont très fortes, si bien que toutes les propriétés sont soit fixées définitivement, soient totalement dépendantes les unes des autres. Il est alors impossible de déterminer si telle ou telle propriété, plutôt qu'une autre, est utile pour une application donnée. On a donc clairement besoin de nouveaux algorithmes de calcul de filtres : ce n'est qu'en réalisant un certain nombre de compromis dans ces algorithmes que l'on arrivera à rendre les différentes propriétés des filtres aussi indépendantes que possibles.

Alors que les critères classiques de calcul de filtres, qu'on vient de passer en revue, portent directement sur les coefficients du filtre ou sur sa réponse en fréquence, le critère de régularité est, comme on l'a vu au chapitre précédent, plus difficile à contrôler. En fait, la seule contrainte simple, nécessaire pour la régularité, est la contrainte de «platitude» de spectre autour de la fréquence de Nyquist : on impose un certain nombre K de zéros à $z = -1$ dans le filtre passe-bas. Il serait éventuellement envisageable d'incorper des contraintes plus complexes provenant des algorithmes d'estimation de régularité présentés au chapitre précédent, mais ceci compliquerait grandement notre tâche. On se limitera donc au critère de «platitude». Bien que celui-ci n'est, en théorie, qu'une condition nécessaire, on montrera, en calculant des ordres de régularité *a posteriori*, qu'elle permet effectivement d'obtenir de la régularité. De plus, on observera que l'ordre de régularité croît généralement avec K, bien que, comme on l'a mentionné au chapitre précédent, cette tendance puisse être inversée dans certains cas.

Ce chapitre est divisée en deux sections principales : le cas orthogonal, pour lequel on se réfèrera (pour la plus grande partie) à l'article [1] reproduit à l'appendice A 4, et le cas «phase linéaire», pour lequel on pourra se réfèrer à l'article [17]. Les filtres construits sont à coefficients réels.

5.1 Filtres orthogonaux

5.1.1 Méthode générale

Les équations permettant le calcul de bancs de filtres orthogonaux ont été rappelées au chapitre 2 (§ 2.1.4). Il ne reste qu'un seul filtre, disons le filtre passe-bas $G(z)$, de longueur L paire, à déterminer. La condition d'orthogonalité (sous forme causale, qui implique la reconstruction parfaite à un retard de $L-1$ échantillons près) s'écrit :

$$P(z) - P(-z) = z^{-(L-1)} \tag{5.1}$$

où

$$P(z) = G(z)G(z^{-1})z^{-(L-1)} \tag{5.2}$$

est un filtre à phase linéaire (réponse impulsionnelle symétrique) dont la réponse fréquentielle $P(e^{j\omega}) = |H(e^{j\omega})|^2$ est non-négative.

La contrainte de «platitude» d'ordre K du spectre de $G(e^{j\omega})$ autour de la fréquence de Nyquist s'écrit comme une condition linéaire sur les coefficients de $P(z)$:

$$(1+z^{-1})^{2K} \text{ divise } P(z) \tag{5.3}$$

C'est cette condition qui va nous permettre d'obtenir des filtres réguliers.

La méthode générale[1], classique [9, 10], de la détermination de $G(z)$ peut être vue en deux étapes :

1. On détermine d'abord $P(z)$ qui satisfait aux contraintes, notamment les spécifications fréquentielles du filtre.

2. La solution $G(z)$ se déduit ensuite en factorisant $P(z)$ sous la forme (5.2) (ce qu'I. Daubechies a appelé «lemme de Riesz» [5]).

Alors que l'on peut contrôler la réponse en amplitude (fréquentielle) du filtre dans l'étape 1, on n'a accès qu'a un nombre limité de choix de la *phase* du filtre dans l'étape 2. Dans la suite, on focalisera donc sur l'étape 1.

5.1.2 Approximation au sens de Tchebichev

Nous venons de voir que l'étape principale du calcul des filtres consiste à imposer des contraintes sur $P(z)$ pour contôler la réponse fréquentielle $|G(e^{j\omega})|^2$ du filtre solution $G(z)$. Du point de vue fréquentiel, imposer une sélectivité en fréquence sur $G(z)$ (ou $P(z)$) revient à approcher au mieux un filtre passe-bas *idéal*, dont la réponse fréquentielle est constante dans la bande passante, et nulle dans la bande atténuée. Pour ce faire, il faut définir un critère d'approximation.

Le critère le plus couramment utilisé est un critère de minimisation de norme $\mathbf{L}^\infty$, ou «min-max», qui correspond à une *approximation au sens de Tchebichev* dans les bandes passante et atténuée. On cherche donc, sur ces bandes, à minimiser le maximum de l'erreur entre $|G(e^{j\omega})|^2$ et la réponse idéale. Cela revient à imposer un *gabarit* du type décrit à la figure 5.1, où l'on

[1] Une autre méthode classique consiste à calculer le filtre sous sa représentation en treillis de Vaidyanathan [11]. Cette représentation présente certains avantages, comme celui de conservation exacte de la reconstruction parfaite et de l'orthogonalité par numérisation des coefficients du filtre, mais semble mal adaptée aux contraintes de régularité.

Figure 5.1: Gabarit du filtre passe-bas $P(e^{j\omega})$ utilisé pour le programme de Tchebichev.

cherche à minimiser la tolérance δ pour une bande de transition (ω_p, ω_s) donnée (qui définit les bandes passante $(0, \omega_p)$ et atténuée (ω_s, π).

Un autre critère fera l'objet du § 5.1.7, qui correspond à un type d'approximation «quadratique». Néanmoins, le critère de Tchebichev est tellement utilisé dans la littérature qu'on parle souvent de «sélectivité en fréquence» en faisant référence à celui-ci. Par conséquent, dans la suite de cette thèse, lorsqu'on dira qu'un filtre a une bonne sélectivité en fréquence, cela signifiera que son atténuation dans la bande passante (égale à $-10\log_{10}\frac{\delta}{2-\delta}$ (en dB) pour la figure 5.1) est importante par rapport aux spécifications (ω_p, ω_s) de son gabarit fréquentiel.

Il est facile de voir que la contrainte (5.3) se réduit à K équations linéaires sur les $L/2$ coefficients distincts de $P(z)$, ce qui donne $L/2 - K$ degrés de liberté pour la résolution du programme d'approximation. On peut distinguer deux cas extrêmes, correspondant à $K = L/2$ et $K = 0$.

5.1.3 Les filtres de Daubechies

Les filtres de Daubechies correspondent au cas $K = L/2$ (contrainte de platitude maximale). Il n'y a alors plus de degré de liberté, et $P(z)$ est déterminé de façon unique pour une longueur donnée. En fait, les solutions $P(z)$ sont des filtres (demi-bande) à platitude maximale (*maximally flat filters*), qui sont connus depuis Herrmann [12]. Les ondelettes orthogonales correspondantes ont été proposées par I. Daubechies [5, 6], aussi a-t-on l'habitude de qualifier les solutions $G(z)$ de filtres de Daubechies.

Il existe un grand nombre d'expressions exactes donnant $P(z)$ dans la littérature des filtres à platitude maximale [12, 13], qui permettent de calculer les filtres de Daubechies [5]. La seule difficulté pour ce calcul apparaît lorsqu'on veut déterminer $G(z)$ à partir de $P(z)$, ce qui fait

Figure 5.2: Caractéristiques d'amplitude des filtres de Daubechies et de Mintzer/Smith/ Barnwell. On a représenté, en dB, l'amplitude de la réponse fréquentielle de deux filtres passe-bas de longueur 10. La solution de type Mintzer/Smith/Barnwell (MSB) a été calculé pour une largeur de bande de transition égale à 0.14 (en fréquence normalisée).

appel à une extraction de racines de $P(z)$: parce que $P(z)$ contient $2K = L$ racines multiples à $z = -1$, la routine d'extraction de racines induit un bruit de calcul important pour les grandes longueurs. A toutes fins utiles, je signale donc une nouvelle formule donnant $P(z)$ directement sous forme factorisée $P(z) = (1 + z^{-1})^{2K} Q(z)$, qui permet de calculer les filtres de Daubechies avec une grande précision. Les coefficients (symétriques) de $Q(z)$ sont donnés par [1]

$$q_n \propto \frac{\sum_{i=0}^{n} (-1)^i \left(\mathrm{C}_{2K-1}^{i}\right)^2}{\mathrm{C}_{2K-2}^{n}}$$

pour $n = 0, \ldots, 2K - 2$. (Les q_n sont normalisées de façon que $P(1) = 2$.) Cette formule peut se voir comme une conséquence du fait que les solutions $P(z)$ correspondant aux filtres de Daubechies sont des filtres d'interpolation de Lagrange [14]. Elle nous sera utile dans la suite.

Il est clair, d'après la figure 5.2, que les filtres de Daubechies, bien que réguliers, sont très peu sélectifs en fréquence.

5.1.4 Les filtres de Mintzer/Smith/Barnwell

L'autre cas extrême correspond au cas où l'on n'impose pas de contrainte de platitude ($K = 0$). Mintzer [9] et Smith et Barnwell [10] ont montré qu'alors, les solutions $P(z)$ à sélectivité en fréquence maximale peuvent être calculés par des méthodes traditionnelles de résolution du critère de Tchebichev grâce au théorème d'alternance. De ce fait, on peut utiliser les algorithmes classiques de Parks-McClellan [15] ou de Hoffsetter-Oppenheim-Siegel [16] pour déterminer le

filtre $P(z)$, satisfaisant au gabarit de la figure 5.1. Ceci nécessite une élévation de la réponse fréquentielle par δ (voir [9, 10]).

Comme il est montré dans la figure 5.2, ces filtres sont très (maximalement) sélectifs en fréquence, mais sont bien moins réguliers que les filtres de Daubechies. Ils ne sont en fait pas réguliers du tout – le schéma d'itération des filtres diverge – lorsque $L/2$ est pair, car on a effectivement $K = 0$ et $G(-1) \neq 0$ (*cf.* chapitre 4). Par contre, en conséquence du théorème d'alternance, ces filtres ont une platitude d'ordre $K = 1$ lorsque $L/2$ est impair, ce qui donne un ordre de régularité positif (mais faible).

5.1.5 Résolution du programme de Tchebichev

Dans la suite, on recherche des solutions intermédiaires entre les filtres de Daubechies et ceux de Mintzer/Smith/Barnwell, en cherchant à obtenir le meilleur compromis possible entre régularité et sélectivité en fréquence. On cherche donc à imposer le gabarit de la figure 5.1, où la tolérance δ est minimale étant donné les contraintes simultanées d'orthogonalité (5.1), de positivité sur $P(e^{j\omega})$, et de platitude d'ordre K (5.3) (problème d'optimisation au sens de Tchebichev avec contraintes).

En réécrivant $P(z)$ sous la forme [1]

$$|P(e^{j\omega})| = P(e^{j\omega})e^{j(L-1)\omega} = 1 + \sum_{n=1}^{L/2} a_n \cos(2n-1)\omega,$$

on s'aperçoit que, par construction, la réponse fréquentielle admet $\omega = \pi/2$ comme point d'antisymétrie, et on peut donc se limiter à la demi-bande $[0, \pi/2]$. Le problème d'optimisation se réduit alors à :

$$\min_{a_n} \omega$$

sous les contraintes inégalités de spécification du gabarit (figure 5.1) et de la positivité de $P(e^{j\omega})$:

$$\begin{aligned}
\sum_{n=1}^{L/2} a_n \cos(2n-1)\omega &\leq 1, \qquad \text{pour } \omega \in [0, \pi/2] \\
\sum_{n=1}^{L/2} a_n \cos(2n-1)\omega &\geq 1 - 2\delta, \qquad \text{pour } \omega \in [0, \omega_p]
\end{aligned}$$

et sous les contraintes égalités de platitude [1] :

$$\begin{aligned}
\sum_{n=1}^{L/2} a_n &= 1 \quad (\text{si } K > 0) \\
\sum_{n=1}^{L/2} a_n (2n-1)^{2k} &= 0, \qquad \text{pour } k = 1, \ldots, K-1.
\end{aligned}$$

En intégrant δ comme variable, nous avons un problème d'optimisation linéaire avec contraintes linéaires, qui peut donc facilement se résoudre à l'aide d'une routine de programmation linéaire. Des commentaires sur la réalisation de ce programme sont faits dans [1]. Nous reviendrons à la programmation linéaire à l'occasion d'un autre type d'optimisation.

En fait, pour ce problème, il est possible d'aller plus loin et de le résoudre par un algorithme d'échange de zéros de Remez assez particulier, qui se fonde sur une variante du théorème classique d'alternance. On trouvera tous les détails dans [1], reproduit à l'appendice A 4. Nous nous bornons ici à donner les principales étapes de la dérivation de cet algorithme.

Tout d'abord, il est possible de réécrire $P(z)$ en intégrant les contraintes d'orthogonalité et de platitude simultanément. On obtient, en notant $x = \cos\omega$ et $P(x) = P(e^{j\omega})$,

$$P(x) = P_K(x) + x(1-x^2)^K R(x^2)$$

où $P_K(x)$ est le polynôme solution de Daubechies de degré $2K-1$, et où les $L/2-K$ variables indépendantes sont regroupés dans le polynôme $R(x)$ de degré $L/2-K-1$. Ceci permet, par changement de variable $y = x^2$, de réécrire le problème d'optimisation initial sous la forme classique d'une minimisation d'erreur au sens de Tchebichev :

$$\min_{R(y)} \max_{y\in[\cos^2\omega_p,1]} |W(y)(D(y)-R(y)|$$

où on a posé $W(y) = \sqrt{y}(1-y)^K$ (poids) et $D(y)W(y) = 2-\delta-P_K(\sqrt{y})$ (réponse désirée). La grande différence avec le cas classique est que la tolérance δ intervient dans l'expression de $D(y)$, ce qui rend impossible l'utilisation directe des programmes de calcul de filtres classiques comme l'algorithme de Parks-McCellan.

Malgré cela, on peut montrer [1] que le théorème d'alternance, sous une forme générale, s'applique. On obtient la caractérisation suivante de la solution unique du problème d'optimisation (solution «*equiripple*») : il faut et il suffit que la réponse fréquentielle présente au moins $L/2-K+1$ alternances dans la bande passante (l'erreur pondérée alterne sur $L/2-K+1$ valeurs de y entre son maximum et l'opposé de son maximum).

On montre [1], de plus, que l'on peut toujours se restreindre au cas $L/2-K$ pair. (Si $L/2-K$ est impair, le filtre optimal $G(z)$ a automatiquement un zéro à $z=-1$ de plus, ce qui augmente de 1 la valeur de K). Le nombre $L/2-K$ donne alors le nombre total d'oscillations de la réponse fréquentielle du filtre dans la bande $[0,\pi]$, comme montré en [1, Fig. 4].

Grâce au théorème d'alternance, on peut facilement en déduire une variante de l'algorithme d'échange de zéros de Remez, qui converge rapidemment vers la solution optimale [1]. Cet algorithme a été implanté dans la fonction MATLAB `remezwav()` décrite à l'appendice C, qui est reproduite dans [1]. Il converge en trois ou quatre itérations seulement (en obtenant une valeur de δ précise à 10^{-9} près) pour des longueurs $L \leq 20$, ce qui est, bien entendu, beaucoup plus efficace qu'une résolution par programmation linéaire.

5.1.6 Résultats

Une réponse d'amplitude en fréquence typique est donnée dans la figure 5.3. La figure 5.4 montre que la méthode de calcul de filtres présentée ici permet effectivement d'obtenir un large éventail de possibilités pour la régularité et la sélectivité en fréquence, avec compromis optimal. La sélectivité des filtres de Daubechies peut être grandement améliorée en relâchant quelques contraintes de platitude. Pour des grandes longueurs de filtres $L > 20$, on peut même améliorer à la fois la sélectivité en fréquence et la régularité (qui ne croît plus toujours avec K). Inversement, la régularité des filtres de Mintzer/Smith/Barnwell est grandement améliorée en imposant quelques contraintes de platitude. Remarquons que les cas extrêmes (Daubechies et MSB) sont des cas particuliers fournis par l'algorithme général que l'on vient de présenter.

(a)

(b)

Figure 5.3: Un exemple de filtre régulier maximalement sélectif en fréquence. Cet exemple correpond à une longueur $L = 18$, une platitude d'ordre $K = 3$, et une bande de transition normalisée $(\omega_s - \omega_p)/2\pi = 0.14$. (a) Réponse fréquentielle. L'allure est celle d'une filtre *equiripple* classique, où on observe une "chute" en dB à la fréquence de Nyquist, correspondant à la contrainte de platitude. (b) Ondelette (orthogonale) correspondante, d'ordre de régularité de Sobolev égal à $1.3843\ldots$.

Figure 5.4: Eventail de possibilités fourni par la méthode de calcul des filtres orthogonaux. On a tracé l'atténuation de la bande atténuée, en dB, en fonction de l'ordre de régularité au sens de Sobolev. Les pointillés correspondent aux valeurs constantes de $\mu = L/4 - K/2$. Comme on l'a vu au chapitre 4, la définition de Sobolev correspond aux spécifications spectrales du filtre. On aurait pu, après avoir déterminé un choix de phase pour les filtres solutions, utiliser l'ordre de régularité au sens de Hölder : cela aurait donné une figure similaire. On voit que la méthode présentée ici permet d'effectuer une transition douce entre la famille des filtres de Daubechies ($L/2 = K$) et celle des filtres de Mintzer/Smith/Barnwell ($K = 0$).

Figure 5.5: Réponse en fréquence d'un filtre régulier calculé par approximation "quadratique". Pour cet exemple, les paramètres sont les mêmes qu'à la figure 5.3.

Une discussion quant aux choix de la phase du filtre $G(z)$ se trouve dans l'article [1]. On peut déterminer le choix qui conduit à la phase la plus «proche» de la phase linéaire (au sens où la variation du retard de groupe est la plus faible). Dans le cas des filtres de Daubechies, par exemple, la distortion de phase peut être toujours choisie inférieure à un échantillon seulement.

5.1.7 Cas de l'approximation quadratique

Jusqu'à présent, on a approché des filtres idéaux par optimisation, suivant une critère $\mathbf{L}^\infty$ (de Tchebichev). Un autre type d'approximation envisageable consiste à minimer l'erreur «quadratique» (ou plutôt, l'erreur de type $\mathbf{L}^1$ pour le carré du module de la réponse fréquentielle du filtre $G(z)$) dans la bande passante ou atténuée. Le problème d'optimisation, écrit sur les coefficients a_n, devient alors, en bande passante :

$$\min_{a_n} \int_0^{\omega_p} \left|2 - |P(e^{j\omega})|\right| d\omega$$

ce qui s'écrit :

$$\max_{a_n} \sum_{n=1}^{L/2} a_n \frac{\sin(2n+1)\omega_p}{2n+1}$$

sous les contraintes inégalités de positivité de $P(e^{j\omega})$:

$$\sum_{n=1}^{L/2} a_n \cos(2n-1)\omega \leq 1, \qquad \text{pour } \omega \in [0, \pi/2]$$

et sous les mêmes contraintes égalités de platitude écrites plus haut.

Là encore, on a affaire à une optimisation linéaire, dont la solution peut s'obtenir à l'aide d'une routine de programmation linéaire. Un exemple typique de filtre solution est donné à

L	K	$\mathbf{L}^\infty$	$\mathbf{L}^1$
6	1	0.0536	0.2265
	3	0.9150	0.9150
8	2	0.5409	0.7446
	4	1.2756	1.2756
10	1	0.2263	0.3665
	3	0.9795	1.2157
	5	1.5968	1.5968
12	2	0.7370	0.9398
	4	1.3868	1.6173
	6	1.8884	1.8884
14	1	0.3664	0.4475
	3	1.1899	1.3971
	5	1.7803	1.9762
	7	2.1587	2.1587
16	2	0.9434	1.1100
	4	1.6223	1.8232
	6	2.1604	2.3627
	8	2.4147	2.4147
18	1	0.4555	0.4853
	3	1.4152	1.5975
	5	2.0389	2.2499
	7	2.5356	2.7369
	9	2.6617	2.6617
20	2	1.1319	1.2506
	4	1.8476	2.0368
	6	2.4341	2.6197
	8	2.8785	3.1469
	10	2.9027	2.0927

Table 5.1: Comparaison entre régularités obtenues par les deux méthodes de calcul de filtres. Les valeurs listées sont les ordres de régularité au sens de Sobolev, pour une bande de transition normalisée égale à 0.14. On a retenu que les valeurs de $K > 0$ ($K = 0$ donnerait des ordres de régularité négatifs). Les valeurs sont les même dans le cas des filtres de Daubechies ($K = L/2$), qui ne dépendent pas de la méthode choisie. On notera, là aussi, que l'ordre de régularité ne croît pas toujours avec K.

la figure 5.5. Comparés aux filtres précédents, la table 5.1 montre que ces filtres sont plus réguliers à bande de transition donnée. Ceci est naturel, puisque ces filtres sont, grâce au critère choisi, plus «écrasés» près de la fréquence de Nyquist.

5.2 Filtres à phase linéaire

5.2.1 Généralités

Nous avons déjà dit qu'imposer la phase linéaire sur les filtres d'un banc de filtres à reconstruction parfaite est incompatible avec l'orthogonalité (pour des longueurs de filtres supérieures à 2). Si l'on désire imposer la phase linéaire, on a donc forcément deux filtres différents à déterminer, ce qui rend les choses plus complexes que dans le cas orthogonal.

Rappelons (§ 2.1.1) les équations satisfaites pour un banc de filtres à deux sous-bandes et à reconstruction parfaite, dans le cas général (bi-orthogonal). En se ramenant à la détermination des deux filtres passe-bas de synthèse $G(z)$ et d'analyse $G'(z)$, la condition de reconstruction parfaite se réduit à :

$$P(z) - P(-z) = 2z^{-d}, \tag{5.4}$$

où $P(z)$ est le filtre «produit» des passe bas :

$$P(z) = G(z)G'(z). \tag{5.5}$$

Si l'on impose, de plus, la phase linéaire, c'est à dire que les deux filtres passe-bas présentent une réponse impulsionnelle symétrique, il en sera de même de $P(z)$, et on obtient que le retard

total $d = (L + L')/2 - 1$ est impair, où L et L' désignent les longueurs respectives de $G(z)$ et $G'(z)$. On distingue donc traditionnellement deux types de solutions non-triviales [8] :

Type A : Les deux longueurs sont impaires et diffèrent d'un multiple impair de 2. Par exemple, $L' = L + 2$.

Type B : Les deux longueurs sont paires et diffèrent d'un multiple pair de 2. En particuliers elles peuvent être égales.

Bien que le type B puisse admettre des solutions de longueurs paires égales (comme dans le cas orthogonal), il nécessite des réponses impulsionnelles de filtres passe-haut antisymétriques. Pour des raisons de simplicité d'implantation, on a souvent préféré le type A dans des applications de codage d'images [18], pour lequel toutes les réponses impulsionnelles sont symétriques, et ce malgré la dissymétrie des longueurs. Dans la suite, on se placera indifféremment dans l'un ou l'autre cas.

On peut reconnaitre deux méthodes simples de calcul de filtres bi-orthogonaux [8][2]. La première méthode consiste d'abord à calculer le filtre produit $P(z)$ en imposant des contraintes (notamment (5.4)), puis à en déduire $G(z)$ et $G'(z)$ en le factorisant selon (5.5). Malheureusement, cette méthode ne permet d'imposer efficacement des contraintes de sélectivité en fréquence sur aucun des deux filtres. Imposer par exemple un gabarit sur le filtre produit ne signifie aucunement que chacun des filtres $G(z)$ et $G'(z)$ soit sélectif en fréquence. De plus, à chaque solution $P(z)$ correspond de nombreux choix possibles pour $(G(z), G'(z))$ ayant des propriétés très différentes.

La deuxième méthode consiste à calculer un filtre $G(z)$ indépendamment de l'autre, sous toutes les contraintes désirées, puis à en déduire son *complément bi-orthogonal* $G'(z)$ de telle sorte que la condition de reconstruction parfaite (5.4) soit vérifiée. Cette méthode ne permet pas d'imposer, par exemple, des contraintes de gabarit directement sur le complément bi-orthogonal. En faisant quelques essais, on s'aperçoit rapidemment que prendre pour $G(z)$ un filtre passe-bas sélectif en fréquence ne signifie absolument pas que $G'(z)$ le soit. La réponse fréquentielle de $G'(z)$ peut même «exploser».

Ces considérations négatives ont poussé quelques auteurs [19, 20] a écrire un problème d'optimisation général portant sur les deux filtres $G(z)$ et $G'(z)$ simultanément. Les algorithmes de calcul de filtres qui en résultent peuvent donner de bons résultats, mais sont très lourds à mettre en œuvre. De plus, rien ne garantit que les critères (parfois multi-objectifs) choisis par ces auteurs permette à l'algorithme d'optimisation de ne pas converger vers un minimum local.

5.2.2 Détermination du complément bi-orthogonal

Nous avons choisi [17] de pousser un peu plus loin la deuxième méthode simple qui consiste à déduire un complément bi-orthogonal d'un premier filtre calculé de manière indépendante. Commençons par regarder d'un peu plus près la détermination du complément bi-orthogonal, afin de comprendre la manière dont on peut imposer des contraintes de platitude d'ordre K (K zéros à $z = -1$) à la fois dans $G(z)$ et $G'(z)$.

Si la longueur de $G(z)$, L, est impaire, on peut en déduire un unique complément bi-orthogonal $G'(z)$ de longueur $L' = L - 2$. En effet, la condition de reconstruction parfaite

[2] Une troisième méthode classique utilise la représentation en treillis de Vaidyanathan [19] pour les filtres à phase linéaire. Cependant, comme dans le cas orthogonal, cette représentation n'est pas bien adaptée aux contraintes de régularité. De plus, cette représentation n'est complète que pour des longueurs très courtes.

se réécrit alors comme un système à $(L-1)/2$ inconnues – les coefficients de $G'(z)$ – et à $(L-1)/2$ équations linéaires (dont les coefficients viennent de $G(z)$). Le complément bi-orthogonal s'obtient donc simplement comme la solution d'un système linéaire (notons que la dépendance entre les coefficients des deux filtres est donc hautement non-linéaire). Appelons CB_0 cet algorithme de détermination du complément bi-orthogonal sans contrainte de platitude ($K = 0$).

Dans le cas où L est pair, il existe un complément bi-orthogonal unique de longueur $L' = L$. L'algorithme qui le détermine, appelé CB_1 dans la suite, se déduit immédiatement du précédent (CB_0). En effet, puisque la longueur des filtres (symétriques) est paire, ces filtres admettent nécessairement chacun un zéro à $z = -1$. On peut donc écrire $G(z) = (1 + z^{-1})F(z)$ et $G'(z) = (1 + z^{-1})F'(z)$. En observant (5.4) on remarque qu'alors le couple formé des filtres de longueur impaire, $(1 + z^{-1})^2F(z)$ et $F'(z)$ est solution, où $F'(z)$ se détermine à partir de $(1+z^{-1})^2F(z)$ par l'algorithme CB_0. Il suffit donc, pour déterminer $G'(z)$, de d'ajouter un zéro à $z = -1$ à $G(z)$, d'appliquer CB_0, et d'ajouter encore un zéro à $z = -1$ au résultat pour obtenir $G'(z)$.

Cette idée se généralise immédiatement à l'algorithme CB_K, qui permet à partir d'un filtre passe-bas $G(z)$ de longueur L ayant K zéros à $z = -1$, de déterminer un complément bi-orthogonal $G'(z)$ ayant également K zéros à $z = -1$. Pour ce faire, il suffit d'ajouter K tels zéros à $G(z)$, d'appliquer CB_0, puis d'ajouter K zéros à $z = -1$ au résultat. La longueur de $G'(z)$ est alors $L' = L+2(K-1)$. Imposer une forte contrainte de «platitude» augmente donc la dissymétrie des longueurs, bien que L et L' ne diffèrent que d'au plus 2 dans les cas $K = 0$, 1, ou 2. Cet algorithme a été implanté en une fonction MATLAB `biorth()` décrite en appendice C.

Cette méthode pourrait se généraliser au cas où l'on imposerait un nombre différent de zéros à $z = -1$ dans les deux filtres. On a préféré cependant utiliser le même K, afin de simplifier le problème et de s'approcher du cas orthogonal en imposant des ordres de régularité voisins à l'analyse et à la synthèse.

5.2.3 Méthode choisie

Notre méthode[3] de calcul de bancs de filtres à phase linéaire prend en compte un paramètre essentiel ignoré jusqu'ici dans le calcul de banc de filtres, celui du *décalage en fréquence* $\Delta\omega$ *de la bande de transition*. Ce paramètre varie à l'intérieur d'une boucle d'optimisation :

1. Déterminer $G(z)$ par un algorithme de calcul de filtres classique (comme un algorithme de type Remez), de telle sorte que $G(z)$ ait K zéros à $z = -1$ et une bande de transition décalée égale à $(\omega_1+\Delta\omega, \pi-\omega_1+\Delta\omega)$. Le filtre $G(z)$ obtenu est donc maximalement sélectif sous ces contraintes. (Ce calcul a été implanté comme fonction MATLAB `remezz()` décrite à l'appendice C.)

2. Appliquer l'algorithme CB_K à $G(z)$ (*cf.* § 5.2.2) pour obtenir son complément birothogonal $G'(z)$, qui possède également K zéros à $z = -1$.

L'itération s'arrête dès que un *critère* donné atteint son minimum en fonctions de $\Delta\omega$ (et éventuellement d'autres paramètres variables[4] introduites dans l'étape 1).

Plusieurs critères sont envisageables. On pourrait, par exemple, maximiser l' atténuation dans la bande passante calculée sur la réponse fréquentielle de $G'(z)$. Néanmoins, ce critére est

[3]Ce travail a été effectué en commmun avec Pierre Siohan et Hervé Le Bihan du CCETT à Rennes

[4]Pour raffiner les solutions obtenues, le facteur de pondération δ_1/δ_2 – fraction des tolérances d'erreur dans la bande passante et atténuée du filtre $G(z)$ – a été également introduit comme paramètre variable dans l'étape 1.

difficile à mettre en œuvre, car la détermination *a posteriori* de la bande passante de $G'(z)$ pose problème. Le critère à minimiser le plus simple à mettre en œuvre, qui a finalement été choisi, est la différence quadratique entre les deux réponses des filtres :

$$C = \sum_n (g_n - g'_n)^2,$$

où l'indicage est choisie de telle sorte que les filtres soient à phase nulle (la mise en correspondance des coefficients des deux filtres est cohérente avec leur symétrie).

Ce critère permet donc de sélectionner une paire de solutions $(G(z), G'(z))$ «maximalement» proche de l'orthogonalité (qui correspondrait au cas où $G(z) = G'(z)$). De plus, comme le critère C est également la différence quadratique entre les deux réponses fréquentielles des filtres, $C \propto \int |G(e^{j\omega}) - G'(e^{j\omega})|^2 \, d\omega$, la minimisation de C, alliée au fait que $G(z)$ est maximalement sélectif en fréquence, impliquera que la solution optimale $G'(z)$ est également sélective.

Cet algorithme utilise une routine d'optimisation d'une fonction numérique d'une ou deux variables, ce qui est relativement facile à maitriser. En particulier, par inspection systématique, on peut garantir un minimum global du critère.

Notons que la procédure totale consiste en deux minimisations imbriquées et que l'on peut donc craindre un temps de calcul élevé. Néanmoins, l'étape 1 utilise un algorithme efficace (de type Remez) et on a initialisé systématiquement cet algorithme avec la solution $G(z)$ provenant de l'itération précédente, ce qui l'accélère d'un facteur 6 en moyenne. De cette façon, on a obtenu un algorithme suffisemment efficace pour être utilisé en pratique. L'algorithme complet a été implanté, sur Macintosh, dans une fonction MATLAB, `biwav()`, décrite à l'appendice C.

5.2.4 Résultats

Un exemple typique de paires de solutions $(G(z), G'(z))$ de type B est présentée à la figure 5.6, où on la compare avec une solution orthogonale «voisine». Les ondelettes correspondantes sont représentées à la figure 5.7. Dans ce cas, les ordres de régularité des deux familles d'ondelettes bi-orthogonales sont très proches.

Dans le cas où l'on n'impose pas de contrainte de régularité, notre méthode donne des résultats tout-à-fait comparables avec les travaux antérieurs [19, 20], du point de vue des gabarits de filtres obtenus. Cependant, notre méthode est plus facile d'utilisation, met en avant le rôle crucial joué par le paramètre de décalage en fréquence (quelle que soit la méthode utilisée) et permet, bien sûr, d'obtenir des filtres réguliers (au prix, éventuellement, d'une dissymétrie des longueurs).

5.2.5 Extensions possibles

De nombreuses variations sur l'algorithme sont possibles, notamment vis-à-vis du choix du critère (sélectivité en fréquence de $G'(z)$, décorrélation inter-bandes, etc.). Des résultats plus complets seront publiés ultérieurement.

5.3 Conclusion

Nous avons présenté, dans ce chapitre, un certain nombre d'outils, simples à utiliser, de calcul de filtres «ondelettes» permettant d'obtenir des compromis plus ou moins optimaux entre de nombreuses propriétés de filtres différentes. Dans le cas orthogonal, le compromis sélectivité/

(a)

(b)

Figure 5.6: Exemple de solutions bi-orthogonales à phase linéaire. (a) Réponses fréquentielles de $G(z)$ et $H(z)$ (obtenu par modulation à partir de $G'(z)$). Les longueurs des filtres sont $L = 18$ et $L' = 26$, respectivement. La contrainte de platitude est d'ordre $K = 5$. Les paramètres du premier filtre sont : une bande de transition normalisée égale à 0.14, décalée de $\Delta\omega = -0.02$, et un facteur de pondération $\delta_1/\delta_2 = 0.1$. (b) A titre de comparison, on a tracé ici les courbes de réponse fréquentielle d'une solution orthogonale $G(z), H(z)$, pour des valeurs de paramètres comparables (longueur $L = 18$, platitude $K = 5$, bande de transition = 0.14).

(a)

(b)

Figure 5.7: Ondelettes (antisymétriques) correspondant à l'exemple de la figure 5.6. On a tracé les graphes de (a) $\psi(t)$, d'ordre de régularité de Hölder = 2.34, et (b) $\psi'(t)$, d'ordre de régularité de Hölder = 2.43. A titre de comparaison, l'ordre de régularité de Hölder de l'ondelette othogonale correspondant à la figure 5.6 (b), est égal à 2.35.

régularité est optimal. Seul le choix de la phase est limité : on peut néanmoins choisir un filtre le plus proche possible du cas «phase linéaire», avec une variation totale du retard de groupe dépassant rarement 1 ou 2 échantillons. Dans le cas bi-orthogonal à phase linéaire, notre méthode permet d'obtenir des filtres «réguliers», sélectifs en fréquence, et proches du cas orthogonal. Là aussi, le même compromis (probablement sous-optimal) entre régularité et sélectivité en fréquence apparaît.

On peut résumer en disant que l'on a obtenu, pour différentes longueurs, des filtres avec un bon compromis régularité/sélectivité en fréquence, qui sont soit orthogonaux et «presque à phase linéaire», soit à phase linéaire et «presque orthogonaux».

Les nombreux choix de compromis entre propriétés des filtres, rendus possibles par l'utilisation des algorithmes présentés dans ce chapitre, permet d'effectuer une étude exhaustive de l'influence de telle ou telle propriété pour une application donnée. Dans la suite de cette thèse, cette étude sera effectuée pour un schéma particulier de compression d'images fixes.

L'utilisation d'un schéma de codage «par ondelettes/bancs de filtres» suppose un calcul intensif de bancs de filtres itérés en octaves, dont la complexité arithmétique croît avec la longueur des filtres utilisés. Avant de présenter le schéma de compression proprement dit, on va donc développer, au chapitre suivant, un certain nombre d'outils de réduction de la charge de calcul nécessaire à la réalisation de bancs de filtres/ondelettes.

Bibliographie

[1] O. Rioul and P. Duhamel, "A Remez exchange algorithm for orthonormal wavelets," *IEEE Trans. Circuits Syst.—II : Analog and Digital Signal Processing.* Submitted for publication.

[2] A. N. Akansu and R. A. Haddad, *Multiresolution signal decomposition*, New York : Academic Press, 1992.

[3] A. K. Jain, "A fast Karhunen-Loeve transform for a class of random processes," *IEEE Trans. Comm.*, pp. 1023–1029, Sept. 1976.

[4] J. Shore, "On the application of Haar functions," *IEEE Trans. Comm.*, vol. 21, pp. 209–216, Mar. 1973.

[5] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math*, vol. XLI, no. 7, pp. 909–996, 1988.

[6] I. Daubechies, "Orthonormal bases of compactly supported wavelets II. Variations on a theme," *SIAM J. Math. Anal.*, vol. 24, 1993. To appear.

[7] A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Comm. Pure Applied Math.* To appear.

[8] M. Vetterli and C. Herley, "Wavelets and filter banks : Theory and design," *IEEE Trans. Signal Processing*, vol. 40, no. 9, Sept. 1992.

[9] F. Mintzer, "Filters for distortion-free two-band multirate filter banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, no. 3, pp. 626–630, June 1985.

[10] M. J. T. Smith and T. P. Barnwell, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 3, pp. 434–441, June 1986.

[11] P. P. Vaidyanathan and P.-Q. Hoang, "Lattice structures for optimal design and robust implementation of two-band perfect reconstruction QMF banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 1, pp. 81–94, Jan. 1988.

[12] O. Herrmann, "On the approximation theorem in nonrecursive digital filter design," *IEEE Trans. Circuit Theory*, vol. 18, no. 3, pp. 411–413, May 1971.

[13] B. C. Jinaga and S. C. D. Roy, "Explicit formula for the coefficients of maximally flat nonrecursive digital filter transfer functions expressed in powers of $\cos\omega$," *Proc. IEEE*, vol. 73, no. 6, pp. 1135–1136, June 1985.

[14] M. J. Shensa, "The discrete wavelet transform : Wedding the à trous and Mallat algorithms," *IEEE Trans. Signal Processing*, vol. 40, no. 10, pp. 2464–2482, Oct. 1992.

[15] T. W. Parks and J. H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. CT-19, no. 2, pp. 189–194, Mar. 1972.

[16] E. Hoffsetter, A. V. Oppenheim, and J. Siegel, "On optimum nonrecursive digital filters," in *Proc. 9th Allerton Conf. Circuit System Theory*, Oct. 1971.

[17] H. L. Bihan, P. Siohan, O. Rioul, and P. Duhamel, "Une méthode simple de calcul de bancs de filtres/ondelettes bi-orthogonales," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Sept. 1993. To appear.

[18] M. Antonini, M. Barlaud, and P. Mathieu, "Image coding using lattice vector quantization of wavelet coefficients," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Toronto, Canada), 1991, pp. 2273–2276.

[19] T. Q. Nguyen and P. P. Vaidyanathan, "Two-channel perfect reconstruction FIR QMF structures which yield linear phase analysis and synthesis filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 5, pp. 676–690, May 1989.

[20] B. R. Horng and J. A. N. Willson, "Lagrange multiplier approaches to the design of two-channel perfect reconstruction linear phase FIR filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Albuquerque, MN), 1990, pp. 1731–1734.

Chapitre 6

Algorithmes rapides

As time goes to zero, effort goes to infinity.
— EDWARDS' TIME/EFFORT LAW
Any given program costs more and takes longer.
— 2ND LAW OF COMPUTER PROGRAMMING
Anything is easier to take apart than to put together.
— WASHLESKY'S LAW

Ce chapitre présente, dans un cadre général, un certain nombre de techniques de réduction de charge de calcul pour la Transformée en Ondelettes Discrètes, implantée sous forme d'un banc de filtres itéré en octaves.

L'objectif est ici de montrer que, grâce à une réorganisation des cellules de base qui interviennent dans le calcul, l'application de techniques de filtrage rapide permet, à peu de frais, d'obtenir une réduction sensible de la complexité arithmétique d'une TOD.

On ne traite ici que le cas général d'un banc de filtres itérés en octaves, en ne se préoccupant que ponctuellement des contraintes particulières sur les coefficients filtres (cas orthogonal, phase linéaire, etc.). Dans ce cadre, l'accélération des calculs est rendue possible grâce à la structure de filtrage omniprésente dans les bancs de filtres, tout en conservant une structure globale de l'algorithme régulière, où l'on applique de façon répétitive des cellules de base identiques.

Le contenu de ce chapitre a été exposé dans l'article [1], reproduit à l'appendice A 5. Cet article se place en fait dans un cadre encore plus général, où des algorithmes, basées sur les bancs de filtres, sont données pour l'implantation d'autres types de transformées en ondelettes (séries d'ondelettes, transformée en ondelettes continue pour l'analyse de signaux).

6.1 Généralités

6.1.1 Complexité arithmétique

L'obtention d'algorithmes rapides se base surtout sur la réduction de complexité arithmétique des bancs de filtres. Ici, on définit la «complexité» comme le nombre total d'opérations (multiplications et additions) nécessaire, *par échantillon du signal d'entrée*, au calcul d'un banc de filtres (TOD) itéré en J octaves. Puisque la TOD est à échantillonnage critique, la complexité par point d'entrée est égale à celle déterminée par point de sortie.

Bien entendu, la complexité arithmétique n'est pas le seul critère à prendre en compte, bien qu'elle soit la plus facile à mesurer. Le fait qu'un algorithme ait une structure de calculs régulière

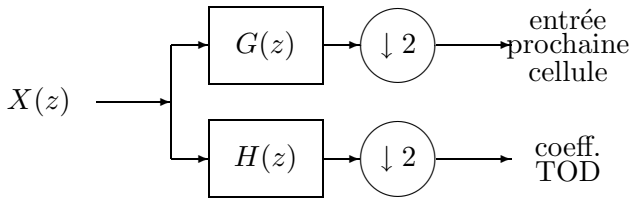


Figure 6.1: Cellule élémentaire de calcul dans une TOD.

est également un facteur important pour des problèmes d'implantation. Cependant, comme ce sera le cas de tous les algorithmes présentés ici, la mesure de la complexité arithmétique est suffisamment instructive pour nous permettre de comparer les différents algorithmes entre eux.

Nous avons choisi, dans ce chapitre, de prendre le nombre total d'opérations (multiplications + additions) comme critère. Avec la technologie actuelle, ce critère est plus pertinent que simplement le nombre de multiplications, tout au moins pour une implantation sur ordinateur (non dédié). Un autre choix possible eut été de compter le nombre de multiplications-accumulations. Nous nous restreignons également au cas de signaux et de filtres à valeurs réelles, bien que le cas complexe soit facile à traiter à partir du cas réel [1].

6.1.2 La TOD standard est naturellement rapide

Il est important de noter, au départ, que la réalisation «directe» d'un banc de filtres itéré en octaves, sans réorganisation des calculs, est déjà naturellement «rapide» [2]. En fait, un tel banc de filtres d'analyse, par exemple, utilise répétivement des cellules identiques, dont le prototype est décrit à la figure 6.1. De plus, grâce aux opérations de sous-échantillonnage, chaque cellule est «nourrie» par un signal dont le taux d'échantillonnage est divisé par deux à chaque itération.

En supposant ques les filtres sont de même longueur L (on fera cette hypothèse dans le cas ou les longueurs sont comparables), la complexité d'une cellule élémentaire, par point d'entrée, est de L multiplications et $L-1$ additions [1]. Au total, pour obtenir la complexité d'un banc de filtres d'analyse, par point d'entrée du signal initial, il suffit de multiplier celle d'une cellule par $(1 + 1/2 + \cdots + 2^{-(J-1)})$, ce qui donne un total de

$$\begin{aligned} 2L(1-2^{-J}) \qquad & \text{multiplications et} \\ 2(L-1)(1-2^{-J}) \qquad & \text{additions par point.} \end{aligned}$$

Grâce aux opérations de sous-échantillonnage, cette complexité est bornée lorsque l'on itère de plus en plus, en faisant croître J. Cette complexité est linéaire par rapport à la longueur des filtres : la TOD entière est équivalente, en termes de complexité, à un filtre de longueur $2L$[1].

L'objet de ce qui va suivre et de réduire encore cette complexité initiale. Ceci permettra, par exemple, d'utiliser des filtres plus longs ; ou de réduire encore la différence entre les complexités nécessaires au calcul de la TOD (en deux dimensions) et de transformées plus classiques pour la compression d'images comme la TCD (Transformée en Cosinus Discrète, appliquée à des blocs de taille 8×8).

[1] Au début du développement de la théorie des ondelettes, la TOD était calculée directement par l'intermédiaire de fonctions de base pré-stockées, sans utiliser la structure hiérarchique des calculs intervenant dans un banc de filtres. La complexité «naïve» qui en résultait était d'environ JL multiplications et JL additions [1]. L'utilisation d'un banc de filtres permet, sans peine, de réduire cette complexité d'un facteur $J/2$.

6.1.3 Calcul de la TOD inverse

Dans la suite, on ne dérivera pas explicitement les algorithmes de calcul de la TOD inverse (banc de filtres de synthèse), pour la bonne raison qu'il est très simple de déduire de tels algorithmes à partir des algorithmes fournis pour la TOD (analyse) [1]. En effet, nous avons vu au chapitre 2 que dans le cas orthogonal, l'algorithme inverse est exactement le *transposé* de l'algorithme direct, à la fois en termes de graphes de fluence et de représentation matricielle. Dans le cas général, c'est la structure des calculs qui se transpose lorsqu'on passe de l'analyse à la synthèse. Il est alors facile de montrer (en supposant les mêmes longueurs de filtres à l'analyse et à la syntyhèse) que les deux complexités associés à la TOD et à la TOD inverse sont strictement *égales*.

6.2 Réorganisation des calculs

On a vu, dans un cas particulier, que la complexité arithmétique totale d'une TOD est obtenue à partir de la complexité arithmétique d'une cellule élémentaire en la multipliant par le facteur $2(1-2^{-J})$. Puisque tous les algorithmes présentés ici vont conserver la même disposition en cellules, cette propriété est générale. On va donc se concentrer, dans ce qui va suivre, sur la réduction de charge de calcul nécessaire à la réalisation d'*une* cellule : il suffit d'appliquer des techniques d'accélération de calculs à une cellule pour obtenir un algorihme rapide pour la TOD entière.

La cellule de base de la figure 6.1 contient deux filtres, et, comme on l'a mentionné en introduction, la réduction de complexité découlera de techniques de filtrage rapide appliquée à ces filtres. Cependant, il ne serait pas bien adapté appliquer directement ces techniques aux deux filtres de la figure 6.1, puisque ceux-ci sont immédiatement suivis d'un sous-échantillonnage. Il est donc nécessaire de ré-organiser les calculs dans une cellule, de telle sorte que de «vrais» filtres (dont on retient toutes les sorties) apparaissent.

A cette fin, on applique une décomposition «biphasée» au signal d'entrée et aux filtres, c'est à dire que l'on définit $X_0(z)$ (échantillons d'indices pairs) et $X_1(z)$ (échantillons d'indices impairs) par la relation

$$X(z) = X_0(z^2) + z^{-1}X_1(z^2)$$

et on procède de même pour les filtres $G(z)$ et $H(z)$. Les sorties d'indices pairs qui sont finalement retenus après filtrage, par exemple par $G(z)$, s'identifient alors à la partie paire de $X(z)G(z)$, c'est à dire [1] à $G_0(z^2)X_0(z^2) + z^{-2}G_1(z^2)X_1(z^2)$. A cause de la dépendance en z^2, la cellule réorganisée fera intervenir quatre «sous-filtres» $G_0(z)$, $H_0(z)$, $G_1(z)$, et $H_1(z)$, fonctionnant à un taux d'échantillonnage moitié. La cellule modifiée qui en résulte est décrite à la figure 6.2.

Pour l'instant, aucune réduction de charge de calcul n'a été effectuée : seule une réorganisation des calculs a été faite, qui va permettre maintenant l'élaboration d'algorithmes rapides par application de techniques de filtrage rapide aux quatres filtres $G_0(z)$, $H_0(z)$, $G_1(z)$, et $H_1(z)$ de longueur $L/2$.

6.3 Algorithmes rapides basés sur la TFR

La technique de filtrage rapide la plus classique utilise la Transfomée de Fourier Rapide (TFR). Il existe des méthodes bien connues pour réaliser un filtrage RIF à partir d'une TFR d'une taille $N = 2^n$ donnée : ce sont les méthodes d'addition-recouvrement (*overlap-add*)

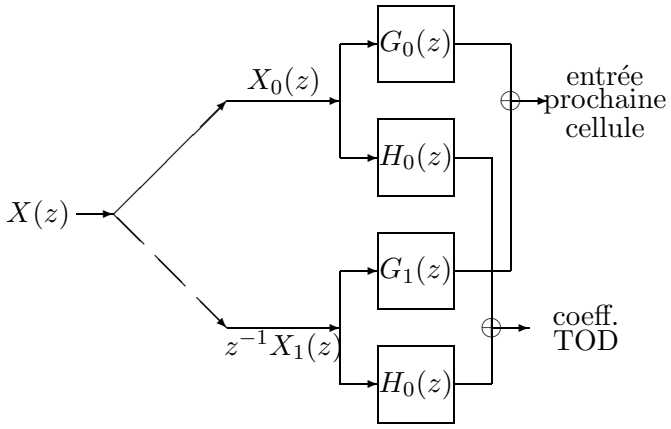


Figure 6.2: Cellule modifiée.

ou de recouvrement-mémoire (*overlap-save*) [3]. Rappelons brièvement que ces méthodes (qui requièrent la même complexité) utilisent un découpage de l'entrée (ou de la sortie) du filtre en blocs de taille B, qui est choisi, pour une taille donnée N de la TFR, de telle sorte que les effets de recouvrement cyclique (*wrap-around*) dans un bloc soient évités.

En appliquant ces techniques de TFR sur chacun des quatre filtres de la cellule modifiée (figure 6.2), et en tenant compte du fait que la même entrée est partagée par deux des quatre filtres, on obtient, en utilisant l'algorithme «à racine double» de TFR (*split-radix FFT*) [4], une complexité arithmétique donnée dans [1], qui dépend de la taille N de la TFR choisie.

En minimisant le nombre total d'opérations (multiplications + additions) en fonction de la taille N de la TFR, on obtient [1], pour une cellule, une complexité par point égale à $4\log_2 L + O(\log\log L)$. Ceci est à comparer avec la complexité «directe», égale à $2L-1$ opérations par point, obtenue en § 6.1.2.

Les résultats, pour différentes longueurs de filtres, sont présentés sous forme de tableau dans [1][2] : l'amélioration, d'un facteur asymptotiquement égal à $L/(2\log_2 L)$, est significative pour les grandes longueurs de filtres. Par contre, on ne réalise pas d'amélioration pour les filtres courts ($L < 16$). On va donc maintenant présenter un algorithme rapide de TOD établi spécialement pour les filtres courts.

6.4 Algorithmes rapides pour les filtres courts

Toujours en partant de la cellule modifiée de la figure 6.2, nous appliquons ici, sur chacun des quatre filtres, une technique de filtrage RIF rapide connue sous le nom de *Fast FIR filtering* [6]. Cette classe d'algorithmes est décrite en détail dans [6]. Elle est particulièrement intéressante du fait qu'elle retient une partie de la structure «multiplication-accumulation» qu'on retrouve dans l'opération de filtrage, ce qui rend son implantation facile.

Rappelons brièvement le principe de ces algorithmes de *fast FIR filtering*. Un filtre de longueur L est réalisé de la façon suivante : les séquences entrée, sortie et réponse impulsionnelle

[2]On décrit également dans [1] un autre algorithme original basée sur la TFR, initialement proposé par Martin Vetterli [5]. Cet algorithme regroupe, dans la calcul du banc de filtres, deux ou trois cellules successives. L'amélioration par rapport à l'algorithme présenté ici est relativement faible, et nécessite une implantation plus complexe.

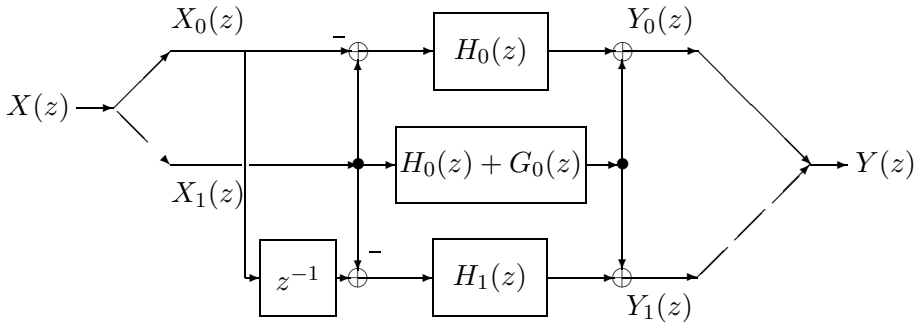


Figure 6.3: Un exemple d'algorithme de filtrage RIF rapide. Les indices 0 et 1 indiquent les composantes biphasées (d'indices pairs et impairs) des signaux d'entrée et de sortie, et de la réponse impulsionnelle du filtre initial $H(z)$.

sont divisées en sous-séquences décimées par un certain facteur R, choisi tel que L est un multiple de R. Le filtrage se fait en trois étapes :

1. L'entrée est divisée en R sous-séquences décimées qui sont combinées d'une certaine manière. Cette opération nécessite un certain nombre d'additions par point d'entrée A_e, et fournit M séquences sous-échantillonnées.

2. Ces M séquences servent d'entrées à M sous-filtres de longueur L/R.

3. Les sorties de ces sous-filtres sont recombinées pour redonner les sorties décimées exactes du filtre initial. Ceci nécessite A_s additions par point.

La figure 6.3 donne un exemple correspondant à $R = 2$, $A_e = 2$, $M = 3$, et $A_s = 2$. On a également appliqué d'autres algorithmes décrits dans [6], correspondant aux valeurs $R = 3$, $A_e = 4$, $M = 6$, $A_s = 6$, et $R = 5$, $A_e = 14$, $M = 12$, $A_s = 26$.

Le principe de cette décomposition peut être répété : les sous-filtres de longueur L/R peuvent à leur tour être calculés par un algorithme du type *fast FIR filtering*. Ainsi, pour implanter un filtre de longueur 6, on a le choix entre appliquer seulement un algorithme correspondant à $R = 3$ ou $R = 2$, ou appliquer successivement des algorithmes correspondant à $R = 3$ puis $R = 2$ (ce qu'on notera «algorithme 3×2»), ou à $R = 2$ puis $R = 3$ («algorithme 2×3»).

Chacun de ces algorithmes requiert une complexité différente. Nous avons choisi, pour chaque cas, de prendre celui qui minimise le critère «nombre de multiplications + additions», et, en cas d'égalité, de retenir celui demandant le moins de multiplications. Nous nous sommes néanmoins restreints à deux niveaux de décomposition au maximum, afin d'obtenir des algorithmes relativement simples, même si cela doit se payer par une légère augmentation de charge de calcul.

Les algorithmes de la classe «*fast FIR filtering*» sont appliqués à chacun des quatre filtres présents dans la cellule de la figure 6.2, en tenant compte du fait que deux paires de filtres partagent le même signal d'entrée : les pré-additions, au nombre de A_e, peuvent être ainsi combinées sur la même entrée.

Par exemple, dans le cas $L = 12$, les filtres sont de longueur $L/2 = 6$, et, comme on vient de le voir, on a le choix entre quatre algorithmes différents. Nous avons trouvé que le meilleur,

selon le critère choisi, est l'algorithme 2×3, pour lequel la complexité par point pour une cellule élémentaire est de 6 multiplications et 12 additions. Ceci est à comparer avec la méthode directe, qui nécessite 12 multiplications et 11 additions par point, et la méthode utilisant la TFR, qui, elle, nécessite 6.18 multiplications et 16.73 additions par point (pour une taille de TFR optimale égale à 16).

Une comparaison précise pour différentes longueurs de filtres est donnée dans [1]. Elle montre que les algorithmes rapides qu'on vient de décrire sont plus efficaces que les autres pour les filtres courts, jusqu'à $L = 18$. Le gain obtenu est variable selon les longueurs : en gros 20% du nombre total d'opérations, ce qui est appréciable lorsque l'on a besoin d'un calcul intensif de TOD.

6.5 Cas particuliers et extensions

Dans notre présentation d'algorithmes rapides, nous n'avons pas tenu compte des contraintes particulières sur les coefficients des filtres telles que l'orthogonalité ou la phase linéaire. Dans ces cas particuliers, une implantation en treillis de Vaidyanathan [7] permet une réduction de charge de calcul de 50%, ce qui est préférable aux algorithmes décrits ci-dessus dans la plupart des cas pratiques. Cependant, la représentation en treillis ne préserve pas la structure de filtrage (produits scalaires) présente dans un banc de filtres, et de plus, dans le cas de filtres à phase linéaire, cette représentation n'est pas complète en général [8].

Dans le cas orthogonal, des développements préliminaires montrent qu'une réduction de 25% de la complexité arithmétique est possible tout en préservant la structure de filtrage. Ceci indique que cette réduction de 25% sera préservée par appplication des algorithmes rapides présentés dans ce chapitre. Le problème de savoir si 50% de réduction de charge de calcul peut être obtenu tout en conservant la structure de filtrage (au contraire d'une implantation en treillis) reste ouvert.

Nous nous sommes également restreints dans ce chapitre au cas ou les différents filtres présents dans une TOD ont des longueurs égales (ou presque égales). Cependant, il est éventuellement possible de considérer le cas où les longueurs des filtres passe-bas et passe-haut sont très inégales. La discussion menée dans l'article [1] montre que, dans le cas des algorithmes rapides de TOD utilisant la TFR, il est souvent préférable de se ramener au cas de longueurs de filtres égales. De toute façon, il est, en règle générale, toujours meilleur, du point de vue de la charge de calcul, de réduire au maximum la dissymétrie des longueurs des filtres.

Pour implanter ces algorithmes en deux dimensions dans le cas séparable, il suffit de les appliquer successivement dans chaque dimension (ligne/colonne). Il est, néanmoins, certainement possible de réduire encore la charge de calcul dans ce cas, tout au moins pour les algorithmes basés sur la TFR, où l'application de TFR bidimensionnelles permettraient une plus grande efficacité.

Enfin, signalons à nouveau que l'article [1] étend la classe des algorithmes décrits dans ce chapitre au calcul des séries d'ondelettes et de la transformée en ondelettes continues (pour l'analyse de signaux par exemple). Il est intéressant de constater que, pour tous les types de transformées en ondelettes, une implantation par bancs de filtres s'impose naturellement et peut conduire à une réduction notable de la charge de calcul.

6.6 Conclusion

Nous avons décrit, dans un cadre général, plusieurs types d'algorithmes rapides pour le calcul de la TOD (banc de filtres itérés en octaves). Les résultats obtenus montrent qu'il est toujours possible, à peu de frais, de réduire substanciellement la charge de calcul, simplement par application de techniques de filtrage rapide sur une cellule élémentaire modifiée. Ceci est réalisé tout en conservant une structure globale «régulière» de l'algorithme, où interviennent plusieurs cellules élémentaires identiques.

Il est intéressant de constater, au vu des complexités arithmétiques obtenues, que les algorithmes rapides de TOD présentés ici sont utiles pour des filtres plus courts que dans le cas d'un algorithme de filtrage simple, alors que la réduction de complexité arithmétique, bien que substancielle, est moins importante.

En résumé, nous disposons maintenant d'algorithmes rapides de TOD, basées sur la Transformée de Fourier Rapide ou sur des algorithmes de filtrage RIF rapide, qui nous permettent de réduire la charge de calcul dans tous les cas possibles. Cette possibilité s'avère importante dans la cas où la TOD (directe et inverse) est la principale source de complexité dans un schéma pratique de codage d'images, par exemple (en supposant que les opérations de quantification et de codage «entropique» requièrent une complexité moins importante). Le chapitre suivant décrit le schéma de compression d'images fixes utilisé pour l'étude expérimentale menée dans cette thèse.

Bibliographie

[1] O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 569–586, Mar. 1992. Part II. Special issue on Wavelets and Multiresolution Signal Analysis.

[2] T. A. Ramstad and T. Saramäki, "Efficient multirate realization for narrow transition-band FIR filters," in *IEEE Int. Symp. Circuit Systems*, 1988, pp. 2019–2022.

[3] H. J. Nussbaumer, *Fast Fourier Transform and convolution algorithms*, Berlin : Springer, 1981.

[4] P. Duhamel, "Implementation of split-radix FFT algorithms for complex, real, and real-symmetric data," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 285–295, Apr. 1986.

[5] M. Vetterli, *Analyse, synthèse et complexité de calcul de bancs de filtres numériques*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 1986.

[6] Z. J. Mou and P. Duhamel, "Short-length FIR filters and their use in fast nonrecursive filtering," *IEEE Trans. Signal Processing*, vol. 39, pp. 1322–1332, June 1991.

[7] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications : A tutorial," *Proc. IEEE*, vol. 78, pp. 56–93, Jan. 1990.

[8] M. Vetterli and D. L. Gall, "Perfect reconstruction FIR filter banks : Some properties and factorizations," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 7, pp. 1057–1071, July 1989.

Partie III

Application à la compression d'images

Chapitre 7

Schéma de compression d'images fixes

New systems generate new problems.
— THE FUNDAMENTAL THEOREM
Everything is part of a larger system.
— 2ND FUNDAMENTAL POSTULATE OF ADVANCED SYSTEMS THEORY
The total behavior of large systems cannot be predicted.
— THE GENERALIZED UNCERTAINTY PRINCIPLE

RAPPELONS que cette thèse s'est donné pour objectif d'étudier l'influence des propriétés des filtres/ondelettes sur les performances de compression d'images fixes par Transformée en Ondelettes Discrètes (TOD). L'analyse théorique qui a précédé, ainsi que les outils développés aux chapitres précédents, vont en partie déterminer les éléments constitutifs du système de compression utilisé (à titre d'exemple) pour effectuer cette étude. La description de ce schéma de compression d'images fixes est l'objet de ce chapitre : il sera suffisamment élaboré pour que les résultats expérimentaux aient valeur de référence, tout au moins avec les restrictions que l'on s'est imposés.

Néanmoins, il est hors de propos ici d'optimiser particulièrement les différents éléments du système de compression (cela demanderait beaucoup plus de travail et la donnée d'une application précise) : on se concentre uniquement, dans cette thèse, sur l'influence des paramètres de la transformée. Par conséquent, la donnée d'un schéma de compression permettra uniquement d'effectuer des comparaisons réalistes (en termes de performance de codage) entre les différents filtres utilisés dans une TOD.

Le schéma général, décrit à la figure 7.1, est celui d'un codage par transformée classique, dont nous allons décrire chaque élément fondamental à tour de rôle : transformée, quantification et codage, en justifiant leur présence. Nous décrirons ensuite une méthode d'allocation optimale de débits binaires dans chaque sous-image, proposée par Martin Vetterli et K. Ramchandran [18] dans le cadre plus général des paquets d'ondelettes. Une partie du contenu de ce chapitre a été brièvement exposé dans [1], reproduit à l'appendice A 6.

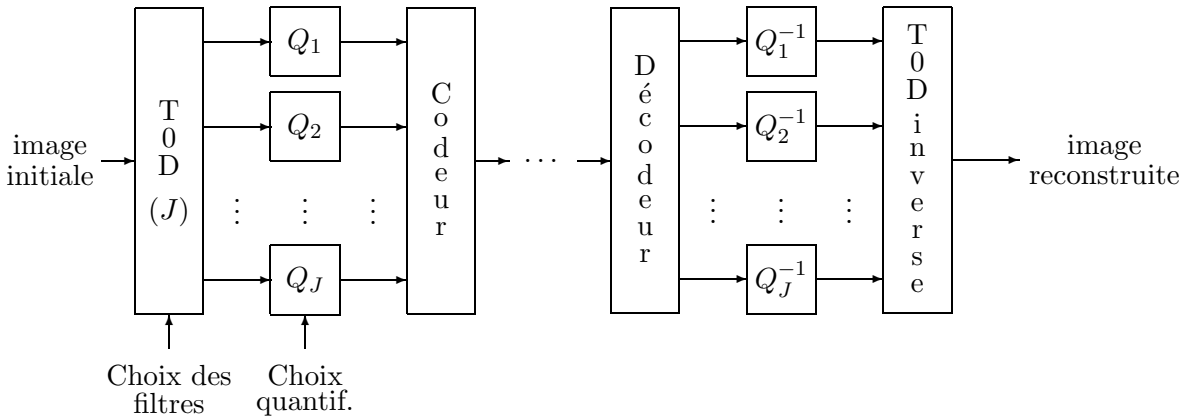


Figure 7.1: Schéma général de compression d'images fixes. La partie «codage» utilise tour à tour une *transformée* (la TOD ou banc de filtres itérés en octaves), qui décompose l'image en composantes (sous-images) associées aux différentes sous-bandes, une *quantification* pour chaque sous-image, et un *codage* "entropique" (sans pertes). L'image codée est ensuite soit transmise (supposément sans erreur), soit stockée. La partie «reconstruction» applique successivement les opérations inverses de décodage, de "quantification inverse", qui consiste simplement à ramener les pixels numérisés dans l'intervalle d'amplitude initial (remise à l'échelle), et la TOD inverse, qui reconstruit une image à partie des sous-images quantifiées.

7.1 Partie "transformée"

7.1.1 TOD bidimensionnelle séparable

Dans les chapitres qui précèdent, nous nous sommes, la plupart du temps, restreints à l'étude d'une transformée en ondelettes discrètes monodimensionnelle, équivalente à un banc de filtres RIF itérés on octaves. On utilisera donc ici une transformée en ondelettes bidimensionnelle séparable, où l'on effectue, séparément sur les lignes et les colonnes de l'image, une TOD monodimensionnelle. Cela constitue l'exemple le plus simple de transformée en ondelettes que l'on puisse étudier : il existe en effet de nombreuses extensions, par exemple utilisant des filtres RII, bidimensionnels non-séparables, etc., pour lesquelles on peut trouver certains avantages pour le codage [11].

Néanmoins, le choix d'une TOD séparable permet d'effectuer une étude précise de l'influence des propriétés des filtres telles que la régularité sur le codage. En effet, on dispose, dans ce cas, d'estimations de régularité optimales (chapitre 4), et de compromis plus ou moins optimaux dans les méthodes de calcul des filtres (chapitre 5). Cela serait difficile à satisfaire si l'on ne se restreignait pas à des filtres RIF, et nécessiterait une étude complémentaire conséquente dans le cas bidimensionnel non-séparable, où les méthodes de calul de filtres réguliers restent mal maitrisées.

La TOD bidimensionnelle séparable, qui privilégie les directions horizontales et verticales, a été abondamment décrite dans la littérature (voir par exemple [3]) : il est donc inutile d'en refaire une description plus précise. Notons tout de même que dans le cas séparable, à chaque niveau de décomposition, l'image d'entrée et transformée en quatre sous-images, correspondant à quatre combinaisons possibles de filtrage passe-bas ou passe-haut sur les lignes et les colonnes. C'est la partie «passe-bas/passe-bas» qui sert ensuite d'entrée à l'itération suivante.

7.1.2 Traitement aux bords

La seule difficulté éventuelle d'implantation de cette transformée provient des bords de l'image. La TOD, en effet, utilise des filtres et est donc mal adpatée à des signaux d'étendue finie : il s'agit de décider quel type d'extension des données on peut effectuer au bords de l'image. Une étude des différents types d'extensions proposés dans [4] a été mené pour la TOD séparable par Lamia Karray [5] lors d'un stage de DEA effectué sous ma direction. Il s'avère facile de réaliser la plupart des extensions (par des zéros, par des constantes, par périodicité, etc.) en suivant l'algorithme décrit par la cellule de base introduite au chapitre précédent (figure 6.2), où seule une modification au niveau des registres à décalage des filtres monodimensionnels utilisés, à l'initialisation et/ou à la fin du filtrage, est nécessaire pour tenir compte d'un type d'extension particulier.

Le plupart de ces extensions, comme l'extension par des zéros, nécessite *a priori* de conserver plus de coefficients sur les bords de l'image, si l'on veut préserver en théorie la reconstruction parfaite aux bords, et donc garantir que l'erreur obtenue à la reconstruction n'est due qu'à l'opération de quantification. Dans ce cas, une décomposition effectuée sur une image de taille $N \times M$ pixels produira quatre sous-images de tailles supérieures à $N/2 \times M/2$. Ceci n'a pas empêché certains auteurs [6] de s'affranchir de conserver ces coefficients supplémentaires ; les artéfacts aux bords obtenus alors sur l'image reconstruite sont dus principalement au choix de l'algorithme utilisé. Cependant, conserver des coefficients supplémentaires pour reconstruire parfaitement les bords nécessite l'introduction de quelques subtilités, comme des troncatures de taille sur les images au fur et à mesure de la reconstruction si l'on veut éviter de manipuler, au bout du compte, un nombre bien plus important de données.

Il existe principalement deux types d'extensions permettant d'éviter ces problèmes. Le premier est le type d'extension le plus couramment utilisé dans des schémas de codage [6], à savoir l'extension aux bords par symétrie. Cependant, celui-ci s'adapte mal dès que les filtres utilisés ne sont pas à phase linéaire, par exemple dans le cas orthogonal. Même si les filtres le sont, il est nécessaire de distinguer, suivant les longueurs, des types de symétrie par rapport à un point ou au milieu de deux points consécutifs, et, suivant que le filtre passe-haut présente une réponse symétrique ou antisymétrique, des extensions par symétrie ou antisymétrie aux bords de l'image. Le deuxième type est l'extension par périodicité. Bien que d'autres types d'extension puissent donner de meilleurs résultats parce qu'introduisant des discontinuités moins abruptes aux bords de l'image analysée, nous avons choisi d'effectuer l'étude expérimentale pour ce type d'extension, qui est probablement le plus simple à mettre en œuvre, et qui permet, de plus, de conserver l'orthogonalité aux bords (ce qui sera utile pour l'allocation de bits optimale). La résolution de tous les problèmes techniques dus aux bords de l'image, notament celui de la conservation de l'orthogonalité, suscite actuellement de nombreux travaux [7, 8].

7.1.3 Rôle de la transformée

Le rôle d'une transformée comme la transformée en ondelettes pour le codage est indirect, mais essentiel : il consiste à «mettre en forme» l'image en la décomposant en sous-images plus ou moins décorrélées entre elles, afin qu'elle soit plus facile à coder. En effet, la distribution statistique de l'image initiale est généralement multipôle, à variance large. Après transformée, bien que la dynamique d'amplitude des sous-images transformés augmente légèrement, ces sous-images présentent des variances bien plus réduites, ce qui facile la compression de données par quantification et codage entropique.

Il existe un certain nombre de mesures «standard» destinées à quantifier la gain en

compression apporté par la transformée, par rapport à une quantification/codage directe sur l'image initiale (codage MIC).

On peut par exemple, calculer l'*entropie* de l'image initiale par rapport à celle, cumulée, des sous-images quantifiées. D'après la théorie de Shannon, l'utilité d'une transformée est contradictoire, parce que l'entropie croît toujours après décomposition, si les différentes composantes ne sont pas statistiquement indépendantes. Cependant, l'entropie au sens de Shannon est une notion théorique qu'il est impossible de mesurer sur une image donnée : en effet, elle est définie comme la limite lorsque $n \to \infty$ (si elle existe), de l'entropie d'ordre n définie par le fameuse formule $H_n = -\sum_i p_i \log_2 p_i$, où les p_i sont les probablités d'occurence des symboles d'une source aléatoire regroupant n pixels. Il est donc difficile de considérer une image donnée comme une source aléatoire dès que n croît, et la notion d'entropie de Shannon n'a pas grand sens pratique. Par contre, si, dans le schéma de compression, on décide de coder, par exemple, chaque pixel indépendamment des autres, alors l'entropie d'ordre 1, H_1, donne effectivement une borne inférieure du débit binaire du codeur. Dans ce cas, son utilisation est justifiée, et on montre effectivement que l'entropie «pixel par pixel» cumulée pour les sous-images quantifiées, est inférieure à celle de l'image initiale. Néanmoins ces mesures supposent un choix préalable des pas de quantification pour chaque sous-image, et dépendent donc fortement de la distorsion produite par la quantification. (La réduction d'entropie est relativement faible si l'on garantit une distorsion nulle.) Un simple calcul d'entropie ne donne donc qu'une indication sommaire de l'utilité relative de telle ou telle transformée pour la compression.

Une autre mesure, très souvent utilisée, est celle du du gain G_{TC} en rapport signal à bruit apportée par une transformée par rapport au codage MIC, pour un même débit binaire alloué aux quantificateurs :

$$G_{TC} = \frac{\frac{1}{n}\sum_i \sigma_i^2}{\sqrt[n]{\prod_i \sigma_i^2}}$$

où σ_i^2 est la variance de chacune des n sous-images. L'utilisation de cette mesure suppose, d'une part, que la transformée est orthogonale (pour que l'énergie totale d'une image soit la somme des énergies des sous-images), et d'autre part, que la variance de l'erreur de quantification sur une image est proportionnelle à $\sigma^2 2^{-2R}$, où σ^2 est la variance (énergie) du signal image, et R est le nombre de bits alloués. Cette supposition n'est justifiable que sous des hypothèses de stationnarité, par exemple si le bruit de quantification est blanc et uniforme. Ces hypothèses sont contestables, surtout pour une compression à moyen débit où la taille des pas de quantification utilisés n'est souvent pas négligeable par rapport à la dynamique des pixels.

Néanmoins, un certain nombre de travaux utilisent G_{TC}, pour mesurer les performances de filtres donnés dans une TOD/banc de filtres [9], ou comme critère à maximiser (ce qui revient à minimiser les variances des sous-images obtenues par transformée) pour déterminer, par exemple, des filtres «optimaux» dans une TOD pour des applications de codage [10].

Dans cette thèse, on préferera comparer différents filtres par l'intermédiaire d'un schéma simple, mais réaliste, de compression d'images plutôt que de se baser sur des mesures générales, qui sont sujettes à caution. On va donc maintenant décrire les opérations de quantification et de codage choisis pour l'étude expérimentale de cette thèse.

7.2 Quantification et codage

Répétons-le, il ne s'agit pas ici d'optimiser particulièrement les opérations de quantification et de codage par rapport à la transformée utilisée, mais plutôt de définir un ensemble de choix réalistes,

qui permetteront de comparer différents paramètres (filtres) de la transformée entre eux. Il n'est donc pas question, pour l'étude expérimentale qui va suivre, de comparer différentes techniques de quantification (scalaire/vectorielle), ni de comparer différents types de transformées (bancs de filtres, TCD), qui sont adaptées à différents méthodes de quantification/codage.

Pour les choix que l'on va décrire maintenant, on quantifie séparément chaque sous-image, sans tenir compte des corrélations inter-sous-bandes.

7.2.1 Quantification scalaire uniforme

La quantification scalaire uniforme consiste à quantifier séparément chaque pixel en 2^R niveaux, où R est le nombre (éventuellement non entier) de bits par pixel (bpp) alloué au quantificateur. Le pas de quantification q est choisi uniforme. Un pixel p est tout d'abord ramené dans l'intervalle de $[0, 2^R - 1]$ par transformation linéaire $ap + b$, où a et b sont calculés en fonction de R et des valeurs minimales et maximales que peut prendre p. La valeur quantifiée correspondante x est l'entier le plus proche de $ap+b$. La quantification inverse consiste à remettre la valeur quantifiée dans l'intervalle d'amplitude initial, par la transformation $(x - b)/a$.

Bien que cela n'ait pas d'influence sur la distorsion apportée par la quantification, il est important, pour obtenir une bonne qualité visuelle de l'image reconstruite, que les pixels de valeur zéro des sous-images "passe-bande" soient quantifiés à la valeur zéro (en d'autres termes, b doit être entier).

7.2.2 Quantification vectorielle sur réseau

La quantification vectorielle sur réseau[1] est l'extension multi-dimensionnelle de la quantification scalaire uniforme. Elle est associée à un réseau régulier Λ à n dimensions (sous-groupe de $\mathbf{Z}^n$), qui permet de quantifier un vecteur de n pixels en un des points du réseau.

On détermine d'abord le «volume» dans lequel les vecteurs de pixels de l'image se trouvent. Cette détermination s'effectue à l'aide d'une norme (généralement $\mathbf{L}^1$ ou $\mathbf{L}^2$). Par exemple, la norme $\mathbf{L}^2$ maximale pour les vecteurs de pixels d'une sous-image définit une hyper-sphère dans laquelle se trouvent les vecteurs de données. Pour la norme $\mathbf{L}^1$, c'est une «hyper-pyramide» [11]. La détermination de ce volume est importante, car on obtient un gain appréciable si celle-ci est bien adapté à la distribution statistique des vecteurs de pixels d'une sous-image. Il est également possible de définir un volume hors duquel on admet un pourcentage faible de vecteurs, qui seront codés séparément [11].

On se ramène ensuite, par un facteur d'échelle a appliqué à chaque pixel, à un volume contenant à peu près 2^{nR} points du réseau Λ, ce qui va nous permettre de quantifier les pixels (sur R bits par pixel) en calculant pour chaque vecteur, le point du réseau le plus proche. On a donc besoin, dans cette étape, de connaître pour un volume donné, la nombre de points du réseau qui s'y trouvent. Cela peut se calculer, une fois pour toutes, grâce à l'utilisation de séries entières, suivant que le volume est une hyper-sphère (norme $\mathbf{L}^2$) [12] ou une hyper-pyramide (norme $\mathbf{L}^1$) [11].

On a également besoin de disposer d'un algorithme de détermination du point du réseau le plus proche d'un vecteur donné. Un certain nombre d'algorithmes, le plus souvent très efficaces (qui ne demandent, par exemple, aucune multiplication) sont décrits dans [12].

[1] Aussi appelée quantification vectorielle en treillis, à cause du terme anglais *lattice quantization*. Le terme "lattice" signifie généralement treillis (comme pour des implantation de filtrage en treillis), mais ici, s'identifie aux "réseaux" en cristallographie.

Enfin, la quantification «inverse» consiste à appliquer l'inverse du facteur d'échelle aux valeurs quantifiées pour se ramener au volume initial.

Un quantification vectorielle sur réseau est *a priori* plus avantageuse qu'une quantification scalaire uniforme, parce qu'elle tient compte des corrélations inter-pixels dans un vecteur, en particulier dans la définition du «volume» dans lequel on quantifie et dans le codage entropique qui suit. Certains réseaux sont également plus intéressants que d'autres, pour ce qui est de la distorsion quadratique moyenne produite (voir paragraphe suivant).

Pour les résultats présentés dans cette thèse, on a implanté une quantification vectorielle en considérant des «vecteurs» constitués d'un bloc de 2×2 pixels, et le réseau D_4, ensemble des vecteurs de $\mathbf{Z}^4$ dont la somme des coordonnées est paire. Les définitions de volume ont été faits à l'aide de la norme $\mathbf{L}^1$ ou $\mathbf{L}^2$.

7.2.3 Justification du choix de la quantification "uniforme"

Il est souvent considéré comme utile, pour des applications de compression, d'optimiser les quantificateurs afin de minimiser la distorsion quadratique pour un débit binaire donné. Ces types de quantificateurs «optimaux» sont non-uniformes, alors que note choix s'est porté sur des quantificateurs uniformes (scalaires ou vectoriels sur «réseau»). Nous allons maintenant tenter de justifier ce choix.

L'expression générale de la distorsion quadratique D par pixel d'un quantificateur uniforme en dimension n (la cas de la quantification scalaire s'en déduit en posant $n = 1$), est souvent mentionné dans la litérature [12]. Je redérive ici cette expression, afin de montrer qu'il est possible de l'obtenir simplement sous certaines hypothèses classiques, et qu'elle constitue une simple extension du cas scalaire (quantificateur de Lloyd-Max). On fait les hypothèses traditionnelles (bien que souvent non justifiables en pratique) suivantes : le vecteur de pixels x est une source aléatoire défini par une densité de probablité $p(x)$, qui est constante à l'intérieur d'une cellule de Voronoï V_i donnée. Rappelons que V_i est l'ensemble des points $x \in \mathbf{R}^n$ les plus proches d'un point du réseau donné x_i (c'est l'intervalle de quantification dans le cas scalaire) ; l'ensemble des V_i forment une partition du volume total V_T. On suppose également que tous les V_i ont même forme, et sont toutes semblables à un protoype V de volume 1. En notant q_i le volume de V_i (pas de quantification dans le cas scalaire), on identifie donc $(V_i - x_i)/q_i$ à V. La distorsion quadratique moyenne par pixel est donc

$$\begin{aligned} D &= \frac{1}{n} \int_{V_T} \|x - x_i\|^2 p(x)\, dx \\ &= \frac{1}{n} \sum_{i=1}^{N} p_i\, q_i^{n+2} \int_V \|x\|^2\, dx, \end{aligned}$$

où $\|x\|$ est la norme $\mathbf{L}^2$ du vecteur x, $N = 2^{nR}$ est le nombre de points de quantification, et $p_i = p(x_i)$.

On veut maintenant minimiser D, c'est à dire minimiser $\sum_{i=1}^{N} p_i q_i^{n+2}$ avec la contrainte $\sum_i p_i^{n/(n+2)} q_i^n = \int p(x)^{n/(n+2)}\, dx$ (constante). On obtient (par la méthode des multiplicateurs de Lagrange) que tous les termes $p_i^{n/(n+2)} q_i^n$ sont égaux (par exemple à leur moyenne arithmétique). On obtient finalement la formule de Zador [13]

$$D = G 2^{-2R} \left(\int p(x)^{n/(n+2)}\, dx \right)^{(n+2)/n} \tag{7.1}$$

où $G = \frac{1}{n}\int_V \|x\|^2\,dx$ (où V est de volume 1) est un terme purement géométrique, qui vaut la fameuse constante $\frac{1}{12}$ dans le cas scalaire. La cas de la quantification uniforme (ou sur réseau) correspond à la même formule dérivée en supposant $p(x) = 1/V_T$ uniforme.

La dérivation de cette formule appelle plusieurs remarques. Tout d'abord, il est difficile de construire, en pratique, un réseau non uniforme optimal, permettant de réaliser la contrainte sur les q_i. Ceci fait appel à des algorithmes sophistiqués de type Linde-Buzo-Gray [14] qui nécessitent une étape d'apprentissage longue et coûteuse. Rien de tel n'est nécessaire pour la quantification vectorielle sur réseau, pour laquelle des algorithmes de quantification efficaces existent.

De plus, la formule (7.1) ne tient pas compte du taux binaire obtenu après codage entropique des vecteurs quantifiés. On a pu constater [2] que le gain obtenu par codage entropique après quantification vectorielle optimale est relativement faible : tout se passe comme si la quantification vectorielle «optimale» tentait de supplanter la partie «codage entropique» par une distribution adéquate des cellules de quantification. Néanmoins, cette façon de faire n'est pas forcément optimale si l'on impose, par exemple, des contraintes sur l'entropie après quantification. Il a même été dit [15] que, pour ce problème, la quantification «uniforme» sur réseau était en fait plus proche de l'optimalité. On peut donc penser que l'association quantification vectorielle sur réseau + codage entropique permet d'obtenir un taux de compression au moins égal à celui obtenu par des quantificateurs vectoriels minimisant la distorsion. La première solution est en tout cas un choix réaliste, bien plus facile à mettre en œuvre.

Remarquons, pour terminer ce paragraphe, que le choix du réseau est souvent déterminé par la valeur de la constante G dans (7.1). Pour $\mathbf{Z}^n$, et notamment pour la quantification scalaire, $G = \frac{1}{12} = 0.0833\ldots$. D'autres réseaux permettent d'obtenir des valeurs plus faibles, et apportent donc un gain purement géométrique. Une borne inférieure de G correspond au cas où les cellules de Voronoï sont des sphères : $G = \frac{1}{2\pi e} = 0.0585\ldots$, qu'il est théoriquement possible d'approcher lorsque $n \to \infty$. Cependant, la plupart des réseaux que l'on peut utiliser en pratique fournissent des valeurs de G intermédiaires [12] : nous donnons ici ces valeurs pour les réseaux D_4, E_8, et Λ_{16}, qui sont ceux pour lesquels G est minimal par rapport à tous les réseaux connus dans leurs dimensions respectives. Pour le réseau D_4, $G = \frac{1}{\sqrt{2}}(\frac{1}{12} + \frac{1}{40}) = 0.0766\ldots$. Pour le réseau E_8, $G = \frac{929}{12960} = 0.0716\ldots$. Enfin, pour le réseau Λ_{16}, $G = 0.0585\ldots$. Ce dernier n'apporte qu'un gain théorique de 0.8 dB en rapport signal à bruit. On peut donc penser que le choix précis du réseau n'est pas vraiment crucial.

Par contre, quantifier vectoriellement apporte potentiellement, par rapport au cas scalaire, un gain de compression plus important, dû au codage entropique qui tient compte des corrélations entre pixels d'un même vecteur. Le paragraphe suivant décrit un certain nombre de critères permettant au bout du compte, de mesurer ce gain de compression.

7.2.4 Codage entropique

Commençons par le cas scalaire, où chaque pixel est codé indépendamment des autres. Les entropies d'ordre 1, calculées pour chaque sous-image, fournissent, dans ce cas, une borne inférieure du débit binaire réalisable par le codeur. Elle s'estime simplement, à partir de l'histogramme d'une sous-image, par la formule

$$H_1 = -\sum_i p_i \log_2 p_i$$

où les p_i sont des estimations des probablités d'occurence des pixels ayant pour valeur i, calculées comme le nombre de pixels ayant la valeur i rapporté au nombre total de pixels dans la sous-image.

L'entropie reste néanmoins une borne supérieure sur le débit binaire et donne, par conséquent, des taux de compression sur-estimés. Nous avons donc choisi de mesurer également le débit binaire obtenu par codage de Huffmann. Par rapport à d'autres méthodes de codage plus sophistiquées (Huffmann adaptatif, Lempel-Ziv, codage arithmétique), nous l'avons retenu surtout parce qu'il est possible de calculer le débit binaire résultant sans implanter le codeur lui-même, simplement à partir d'un histogramme. Il suffit de «singer» l'algorithme de création d'arbre binaire de Huffmann en ne tenant compte que des longueurs de mots de code qui en résultent.

Le débit binaire R_H (en bpp) obtenu après codage de Huffmann satisfait aux inégalités

$$H_1 \leq R_H \leq H_1 + 1.$$

En général, le débit binaire obtenu est plus proche de la borne inférieure (entropie), sauf si le nombre de niveaux de quantification est très réduit (de 0 à 3), auquel cas R_H s'approche plutôt de $H_1 + 1$. Dans ce dernier cas, il serait plus approprié de raffiner les résultats en utilisant une méthode de *run-length coding*, ce qui n'a pas été fait ici.

Dans le cas vectoriel, où l'on code des vecteurs de n pixels, on a utilisé les mêmes mesures : entropie d'ordre n, H_n et débit binaire après codage de Huffmann (pour lequel on a un encadrement plus fin qu'en une dimension : $H_n \leq R_H \leq H_n + \frac{1}{n}$). Comme ces mesures ne nécessitent pas d'implanter un codeur, on ne s'attardera pas sur les problèmes pratiques qui apparaissent, comme la façon de numéroter chaque point du réseau pour coder les vecteurs quantifiés. La seule difficulté, ici, est de pouvoir calculer un histogramme pour ces vecteurs de pixels. En effet, dans la cas de blocs de 2×2 pixels par exemple, le nombre de valeurs possibles pour un vecteur donné est 2^{4R}, où R est le nombre de bits par pixel sur lequel la sous-image a été quantifiée. Cette valeur peut être particulièrement grande, surtout si la quantification est fine. Néanmoins, le nombre total de tels vecteurs est limité par la taille de la sous-image, ce qui rend le calcul de l'histogramme possible par un algorithme adapté.

Dans le cas vectoriel, cependant, ces critères posent problème. En effet, l'application d'une méthode pratique de codage telle que l'algorithme de Huffmann nécessite le stockage ou la transmission de données supplémentaires (*overhead*), fournissant la table de correspondance entre symboles et mots de code. Si cet *overhead* reste négligeable en dimension 1, il n'en est pas forcément de même en plusieurs dimensions, surtout si la sous-image codée est de taille importante ou a été quantifiée finement. C'est pourquoi on s'est imposé certaines restrictions quant aux sous-images susceptibles d'être quantifiées et codées vectoriellement (les autres étant quantifiées et codées pixel par pixel) et on a tenu compte, dans le cas vectoriel, d'une estimation de l'*overhead* à rajouter au débit binaire obtenu par codage de Huffmann.

Toutes ces difficultés font que l'on ne s'attardera pas, dans cette thèse, à comparer directement les résultats obtenus par quantification/codage vectoriel par rapport au cas scalaire. Ceci nécessiterait une analyse des performances du codeur plus poussé, en tenant compte de contraintes algorithmiques.

En résumé, on a choisi ici de disposer de trois critères pour le débit binaire obtenu après transformée et quantification : le débit binaire obtenu sans codage entropique (qui correspond au nombre de bits alloué dans les différents quantificateurs), l'entropie, et le débit binaire obtenu après codage de Huffmann. Il reste, pour ces trois critères, à regarder la façon dont on peut allouer les débits binaires pour les différentes sous-images, et trouver le nombre optimal de

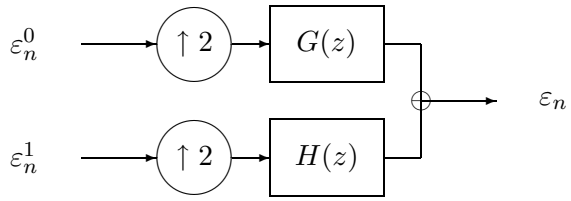


Figure 7.2: Additivité de la distorsion. Sur une étape de décomposition orthogonale, on a $\sum_n \varepsilon_n^2 = \sum_n (\varepsilon_n^0)^2 + \sum_n (\varepsilon_n^1)^2$, où ε_n, ε_n^0, et ε_n^1 désignent les erreurs commises par quantification avant et après décomposition.

décompositions dans la TOD, de façon à minimiser la distorsion obtenue à la reconstruction. C'est ce que nous allons voir maintenant.

7.3 Allocation optimale de débits binaires dans les sous-images [18]

L'idéal serait de pouvoir, par une procédure automatique, maximiser la qualité visuelle de l'image reconstruite pour un débit binaire total donné. Mais, bien entendu, cette qualité visuelle est essentiellement un critère subjectif, et on se contentera de minimiser un critère de distorsion quadratique D, qui permet effectivement de trouver des algorithmes d'allocation optimale de débit binaire dans les sous-images. La qualité visuelle obtenue ne peut alors qu'être observée *a posteriori* sur écran.

7.3.1 Préliminaires

Afin de fixer les notations, désignons par D la distorsion quadratique totale, égale à $\|x - \hat{x}\|^2$, la norme $\mathbf{L}^2$ de la différence entre l'image initiale x et l'image reconstruite $\hat{x}$. Appelons R le nombre de bits total nécessaire pour coder l'image $\hat{x}$. Suivant le critère choisi, R peut se rapporter au nombre de bits alloué aux quantificateurs, à l'entropie ou au nombre de bits nécessaire au codage de Huffmann. On note également R_i le nombre de bits total nécessaire au codage de la sous-image n$^\circ i$, et D_i la distorsion apporté par quantification de cette sous-image. On a toujours $R = \sum_i R_i$.

Dans la suite, on supposera également que $D = \sum_i D_i$: ceci n'est exact que pour une TOD orthogonale, comme montré dans la figure 7.2. Cette relation n'est bien sûr qu'approchée dans le cas de filtres biorthogonaux, que l'on peut néanmoins choisir proches du cas orthogonal (chapitre 5). On néglige également ici la (faible) réduction finale de distorsion apportée à la fin de la reconstruction, lorsqu'on re-quantifie l'image reconstruite sur le même nombre de bits par pixel (en général 8) que l'image originelle.

Cette condition d'additivité des distorsions est essentielle pour la suite : elle permettera de simplifier grandement l'algorithme d'allocation optimale des débits binaires. En effet, elle permet de ne s'intéresser qu'à la partie «analyse» de la transformée, en s'affranchissant totalement de l'opération de reconstruction.

Le problème posé est donc de trouver les meilleurs quantificateurs (définis par type – scalaire/vectoriel – ou par nombre de bits alloués) qui minimise le critère

$$\min D = \sum_i D_i \tag{7.2}$$

sous la contrainte d'obtenir, en final, un nombre total de bits ne dépassant pas un certain nombre donné R_d (budget en bits).

$$R = \sum_i R_i \leq R_d \tag{7.3}$$

7.3.2 Choix de l'algorithme

Il existe plusieurs façons de traiter ce problème. Une façon classique d'opérer est de faire l'hypothèse que la distorsion D_i s'exprime en fonction de R_i sous la forme $D_i = c_i 2^{-2R_i}$, où c_i est une constante. Nous avons vu que ceci s'obtient sous les hypothèses, non justifiables en général, que le signal est stationnaire, décrit par une densité de probabilité uniforme à l'intérieur d'une cellule de Voronoï (ou d'un intervalle de quantification). De plus, cela ne s'applique qu'au cas où les R_i représentent les taux binaires alloués aux quantificateurs, sans tenir compte d'un éventuel codage entropique. Avec de telles hypothèses, cependant, il est facile de résoudre le problème (7.2) sous la contrainte (7.3) par la méthode des multiplicateurs de Lagrange. On trouve que l'optimum est atteint lorsque toutes les distorsions D_i sont égales. Le résultat est décevant, car il donne en pratique, des taux de compression faibles qu'il est facile d'améliorer en faisant varier les D_i. Ceci a amené Michel Barlaud et son équipe [2] à introduire artificiellement les coefficients de pondération sur les D_i dans le problème (7.2). Le problème devient alors que l'on pert l'automatisation de la procédure, étant obligé d'optimiser ces poids «manuellement», par une technique d'apprentissage ou à l'aide d'un modèle.

On a donc préféré mettre en œuvre une autre technique, proposée par plusieurs auteurs [16, 17, 18] et appliquée au cas des paquets d'ondelettes par Martin Vetterli et K. Ramchandran [18]. Cette technique est attractive car elle est entièrement automatique et très flexible : elle permet d'optimiser l'allocation en débits binaires directement sur une image donnée, sans nécessairement faire un apprentissage afin de fixer définitivement tous les paramètres de quantification à employer sur un ensemble d'images. Il est également possible de l'employer à des fins d'apprentissage. Néanmoins, on a choisi ici d'optimiser chaque image séparément afin de pouvoir comparer de façon équitable les propriétés des filtres utilisées dans la transformée : le résultat de l'optimisation dépend directement de l'image traitée et des filtres utilisés.

7.3.3 Description graphique

Le principe de cet algorithme consiste à choisir, parmi un ensemble de quantificateurs q_i à déterminer au préalable pour chaque sous-image, ceux qui résolvent le problème (7.2) suivant la contrainte (7.3). En se basant sur les travaux de Vetterli et Ramchandran [18], je propose de décrire ici d'une façon très simple, (parce que graphique) la résolution de ce problème.

On commence par déterminer toutes les valeurs de R_i et de D_i pour tous les quantificateurs possibles q_i pour chaque sous-image n$^\circ$ i. Ce pré-calcul n'est pas forcément très coûteux, surtout si on se limite à des choix restreints de quantificateurs. En effet, il ne nécessite que le calcul d'une TOD (banc de filtres d'analyse) et se borne à considérer chaque sous-image indépendamment des autres.

Si l'on effectuait toutes les combinaisons possibles des $(R_i(q_i), D_i(q_i))$, on obtiendrait un nuage dense de points dans le plan (R, D). On va rechercher, pour cet algorithme, le point optimum (R^*, D^*) uniquement sur *l'enveloppe convexe* de ce nuage, qui est une ligne brisée représentée à la figure 7.3. Alors, le point optimum sur l'enveloppe convexe est celui qui se

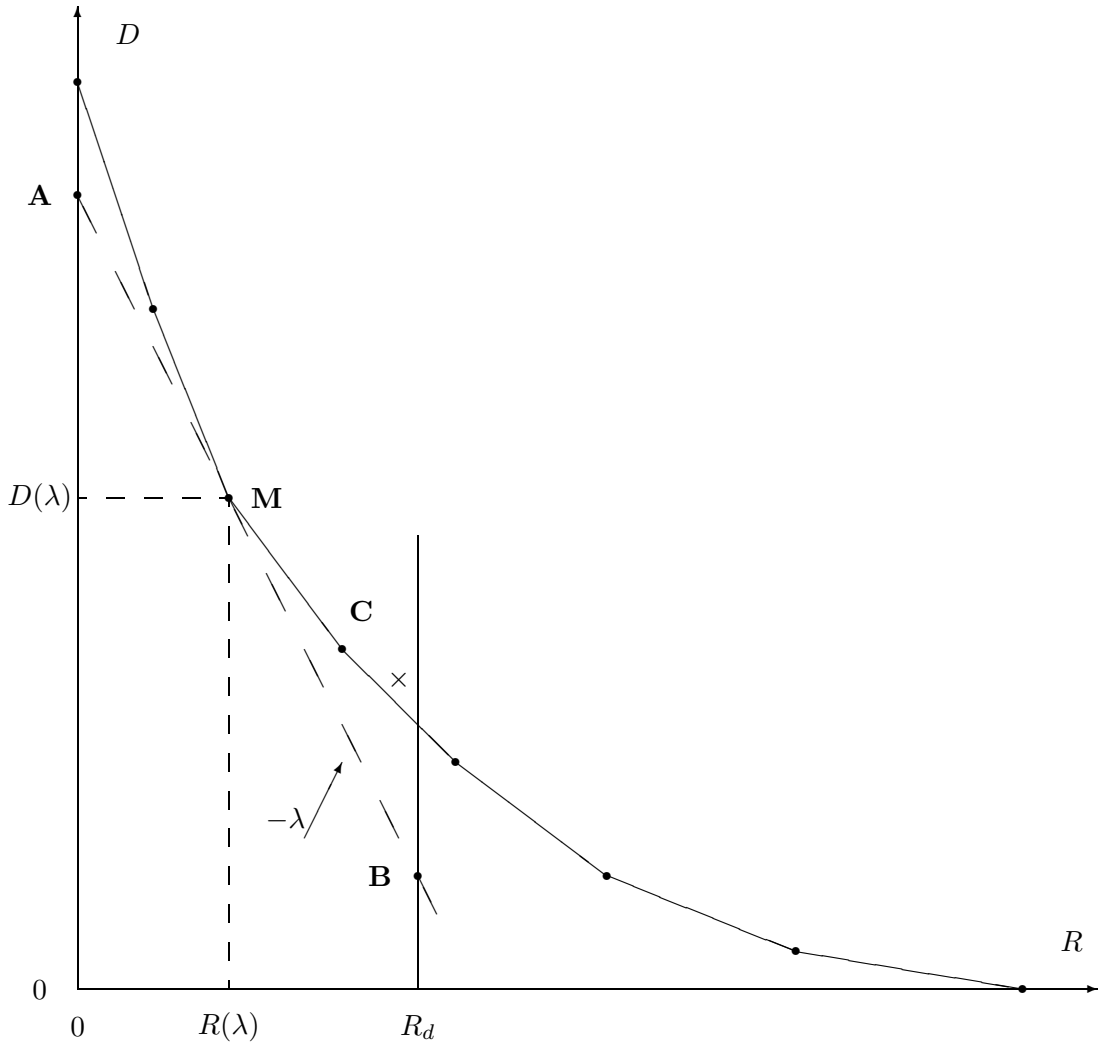


Figure 7.3: Illustration graphique de l'algorithme d'allocation optimale de bits.

trouve immédiatement à gauche de la barre $R = R_d$ (point **C** sur la figure 7.3)[2].

L'algorithme d'allocation optimale de bits prend alors la forme de deux optimisations imbriquées :

1. **Recherche d'un point sur l'enveloppe convexe en fonction d'une pente $-\lambda$:** Pour une valeur donnée de $\lambda > 0$, on recherche l'unique point de l'enveloppe convexe qui lui est associé : il est défini comme l'intersection d'une droite de pente $-\lambda$ (en pointillé sur la figure 7.3) avec l'enveloppe convexe, lorsque celle-ci est réduite à un point. Dans la figure, on a choisi $\lambda = 2$, et le point associé est alors **M**, de coordonnées $(R(\lambda), D(\lambda))$. Si $-\lambda$ se trouve être la pente d'un des segments constituant l'enveloppe convexe, l'intersection est en fait tout ce segment : on choisit alors le point situé à sa gauche.

   Cette détermination revient à trouver le point $(R(\lambda), D(\lambda))$, parmi tous les points du nuage, qui minimise l'ordonnée de **A** sur la figure. Il est facile de voir que cette quantité vaut $D + \lambda R$. (En cas d'égalité, on choisit le point dont la valeur de R est la plus faible.) Autrement dit, on résout le problème :

$$\min_{q_i} D + \lambda R = \sum_i (D_i + \lambda R_i)$$

   Cette minimisation s'effectue facilement en minimisant $D_i + \lambda R_i$, pour un λ donné, indépendamment pour chaque sous-image.

2. **Recherche de la pente donnant le point optimal :** Il reste à déterminer une valeur de λ «optimale» pour laquelle l'étape 1 ci-dessus fournit le point optimum (**C** sur la figure). Il est facile de voir, en considérant toutes les droites de pente variable passant par leurs point associés sur l'enveloppe convexe, que cela revient à maximiser l'ordonnée du point **B** représenté sur la figure, qui est une fonction concave de λ. Cette ordonnée vaut en fait $D + \lambda(R - R_d)$. La pente optimale λ^* est donc la solution du problème

$$\max_{\lambda} D(\lambda) + \lambda(R(\lambda) - R_d)$$

   où $R(\lambda)$ et $D(\lambda)$ sont fournis par l'étape 1. Cette maximisation est effectuée par un algorithme de type Newton sur les pentes [18], qui, en pratique, converge très rapidemment vers la solution optimale $R^* = R(\lambda^*), D^* = D(\lambda^*)$.

Une analyse plus poussé de cet algorithme, basée sur des fonctionnelles lagrangiennes (qu'on reconnait dans les optimisations ci-dessus), est donnée dans [18]. Notons que cet algorithme peut tout aussi bien déterminer le nombre total de bits minimal R sous une contrainte sur la distorsion $D \leq D_d$: il suffit d'échanger les rôles de R et D.

7.3.4 Nombre de décompositions optimal

Il est important de noter que cet algorithme a également été utilisé pour déterminer, dans la TOD, le nombre optimal de niveaux de décomposition J. Ceci peut se faire simplement [18] grâce à l'additivité de R et D (*cf.* figure 7.2). Il suffit, dans l'étape 1, de considérer la valeur de $\sum_{i=1}^{4}(D_i + \lambda R_i)$ pour les quatre sous-images (numérotées $i = 1$, 2, 3 et 4) de taille (et résolution)

[2] Il est possible, néanmoins, que le vrai optimum n'apparaisse pas sur l'enveloppe convexe («point caché», représenté par une croix dans la figure 7.3). Mais l'erreur commise est généralement très faible si le nuage de points est suffisamment dense, c'est à dire si le choix de quantificateurs est suffisamment large.

la plus petite, correspondant au dernier niveau $j = J$. Si cette valeur est plus grande que la valeur $D + \lambda R$, correspondant à la sous-image passe-bas au niveau $j = J - 1$ (dont les quatres sous-images proviennent par décomposition), on «élague» l'arbre de décomposition en réduisant la valeur de J d'une unité, et on recommence le test. Sinon, J est optimal pour la valeur donnée de λ. La valeur de J optimale est alors obtenue à la fin de l'algorithme.

7.4 Conclusion

Nous avons passée en revue les différents éléments constitutifs et algorithmes du schéma de compression d'images fixes, utilisé pour obtenir les résultats montrés au chapitre suivant. En résumé, le schéma choisi utilise une TOD bidimensionnelle séparable, où le nombre total de niveaux de décomposition (octaves) J est déterminé optimalement au sens du § 7.3. Les sous-images sont quantifiées de façon «uniforme», que ce soit scalairement ou vectoriellement sur réseau. Les quantificateurs sont, là encore, déterminés de façon «optimale». La taux de compression est finalement déterminé par trois critères, au choix : débit binaire après quantification (sans codage entropique), débit binaire après codage de Huffmann, ou entropie. Ces mesures sont déterminées à partir d'histogrammes, sur les pixels ou sur les vecteurs de pixels. La quantification vectorielle apporte un gain de distorsion «géométrique», faible par rapport au cas scalaire, mais permet surtout d'obtenir un gain potentiel important au niveau du codeur, qui code des vecteurs de pixels en tenant compte en partie des corrélations entre pixels. (Notons que l'on peut également appliquer un codage «vectoriel» sur des blocs de pixels quantifiés scalairement). L'utilisation de tels codeurs sur des vecteurs de pixels, avec les difficultées qui leur sont propres (*overhead*, etc.) posent un certain nombre de problèmes ouverts qui demanderaient un travail supplémentaire.

Nous avons également justifié les choix qu'on vient de mentionner, pour l'utilisation qu'on compte faire du schéma de compression, à savoir, comparer les performances de différents filtres dans la transformée en ondelettes discrète. On a ainsi justifié l'utilisation d'un schéma de compression donné, par rapport à celle de mesures générales, mais sujettes à caution. On a également insisté sur la nécessité de disposer d'un bon algorithme d'allocation optimale de débits binaires. L'algorithme choisi ici est suffisamment rapide pour être utilisé, dans nos simulations, indépendamment sur chaque image initiale codée.

Il reste à utiliser le schéma qu'on vient de décrire, afin d'obtenir les principaux résultats pratiques de cette thèse, qui sont discutés dans le chapitre suivant.

Bibliographie

[1] O. Rioul, "On the choice of "wavelet" filters for still image compression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Minneapolis, MN), Apr. 1993. To appear.

[2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Acoust., Speech, Signal Processing*. To appear.

[3] S. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 12, pp. 2091–2110, Dec. 1989.

[4] G. Karlsson and M. Vetterli, "Extensions of finite length signals for subband coding," *Signal processing*, vol. 17, pp. 161–168, 1989.

[5] L. Karray, "Compression d'images par ondelettes," Mémoire de D.E.A., 1992.

[6] M. Antonini, *Transformée en ondelettes et compression numérique des images*, PhD thesis, Université de Sophia-Antipolis, 1991.

[7] A. Cohen, I. Daubechies, and P. Vial, "Wavelets on the interval," in *Proc. Int. Colloqueum "Wavelets and Applications"*, (Toulouse, France), 1992. To appear.

[8] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli, "Time-varying orthonormal tilings of the time-frequency plane," *IEEE Trans. Signal Processing*. Special issue on wavelets, to appear.

[9] A. N. Akansu and R. A. Haddad, *Multiresolution signal decomposition*, New York : Academic Press, 1992.

[10] P. Desarte, B. Macq, and D. T. M. Slock, "Signal-adapted multiresolution transform for image coding," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 897–904, Mar. 1992.

[11] M. Barlaud, P. Solé, M. Antonini, P. Mathieu, and T. Gaidon, "Pyramidal lattice vector quantization for multiscale image coding," *IEEE Trans. Image Processing*. To appear.

[12] J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices and groups*, New York : Springer, 1988.

[13] P. Zador, "Asymptotic quantization error of continuous signals and their quantization dimension," *IEEE Trans. Inform. Theory*, vol. 28, pp. 139–149, 1982.

[14] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Comm.*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[15] A. Gersho, "Asymptotically optimal block quantization," *IEEE Trans. Inform. Theory*, vol. 25, no. 4, July 1979.

[16] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1445–1453, Sept. 1988.

[17] G. J. Sullivan and R. L. Baker, "Efficient quadree coding of images and video," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, May 1991, pp. 2661–2664.

[18] K. Ramachandran and M. Vetterli, "Best wavelet packet bases in the rate distortion sense," *IEEE Trans. Image Processing*, Apr. 1993.

Chapitre 8

Résultats

Nothing ever comes out as planned.
— SEAY'S LAW
In case of doubt, make it sound convincing.
— FINAGLE'S 4TH RULE

Ce dernier chapitre présente les résultats détaillés quant à l'influence de propriétés des filtres de la Transformée en Ondelettes Discrètes sur les taux de compression obtenus dans le cadre du schéma décrit au chapitre précédent. Insistons tout de suite sur le fait que les résultats présentés ici ne sont valables que pour ce simple schéma de compression d'images fixes. Le problème de l'influence des propriétés de filtres pour une TOD bidimensionnelle non-séparable, par exemple, reste ouvert.

La plus grande partie du contenu de ce chapitre a également été exposé dans [1], reproduit à l'appendice A 6.

8.1 Préliminaires

8.1.1 Filtres utilisés

Les filtres utilisés comme paramètres de la TOD sont ceux calculés par les méthodes exposées au chapitre 5. Rappelons que l'on avait déterminé ces filtres à partir de quelques variables. Pour le cas orthogonal avec critère de Tchebichev ou le critère «quadratique», les filtres maximalement sélectifs en fréquence avec contraintes de régularité ont été déterminés par la donnée de leur longueur (paire) L, du nombre K de zéros à $z = -1$ dans le filtre passe-bas (où l'on s'est restreint à $L/2 - K$ pair), et de la largeur de la bande de transition $\Delta\omega$. Dans les résultats présentés ici, la longueur L varie de 2 à 18, toutes les valeurs de K pour une longueur donnée ont été considérées, et les largeurs de bande de transition sont, en fréquence normalisé, $\Delta\omega = 0.0625$, 0.1, et 0.14 (avec les mêmes conventions que pour les filtres classiques de Johnston [2]). On pourra se référer aux chapitres 4 et 5 pour les valeurs correspondantes de l'ordre de régularité de Sobolev ou de Hölder, et des atténuations en bande passante obtenues. On présentera également des résultats obtenus pour les bancs de filtres à phase linéaire déterminés par la procédure décrite au chapitre 5.

8.1.2 Images testées

Les images que l'on a utilisé pour les simulations sont les images standard J-PEG : BARBARA, ZELDA, et BOATS de taille 576×720, BRIDGE de taille 512×512, et LENA de taille 256×256. Toutes ces images test sont initialement codées à 8 bits par pixel (bpp). Dans la suite, on focalisera surtout sur LENA et BARBARA, par manque de place, et afin de comparer plus précisément les différents tests. Il est important de signaler que l'on a constaté que *les comportements expérimentaux* (allures des courbes, etc.) *sont les mêmes pour toutes les images testées.*

8.1.3 Visualisation des résultats

Pour évaluer numériquement les performances du schéma de compression avec allocation optimale de débits binaires dans les sous-images (*cf.* chapitre précédent), on a choisi de représenter les courbes donnant le rapport signal à bruit «crête» (*Peak Signal over Noise Ratio (PSNR)*) en fonction du nombre de bits par pixel nécessaire, après codage de Huffmann, pour coder entièrement l'image. Rappelons que le rapport signal à bruit crête, souvent utilisée comme mesure de distorsion, est défini en fonction de la distorsion quadratique totale D (définie au chapitre précédent) par

$$\mathrm{PSNR(dB)} = 10 \log_{10} \frac{S_c}{D}$$

où S_c est l'énergie totale d'une image constante dont tous les pixels sont égaux à la valeur maximale des pixels de l'image initiale. Pour une image donnée, cette mesure ne diffère du rapport signal à bruit traditionnel que par une constante additive.

Nous avons également testé d'autres critères pour le débit binaire (entropie, et débit binaire sans codage entropique). On s'est aperçu, de façon générale, que les comportements expérimentaux, là encore, ne dépendent pas de façon cruciale du critère choisi. Les courbes obtenues ci-dessous sont reproduites pour des valeurs du débit binaire entre 0 et 1.5 bpp, mais on observe des comportements similaires pour des valeurs plus élevées [1].

8.1.4 Mise en œuvre de l'allocation optimale de bits

Les courbes de visualisation des résultats ont été obtenues grâce à l'algorithme d'allocation optimale de bits, décrit en détail au chapitre 7, de la façon suivante : avec les notations de ce chapitre, une courbe donnée correspond à tous les points de l'enveloppe convexe dans le plan (R, D), afin de trouver toutes les distorsions optimales pour chaque valeur de R_d, c'est à dire pour chaque valeur du taux de compression visé. Il suffit, pour l'obtenir, de ne considérer que l'étape 1 de l'algorithme d'allocation de bits, où l'on fait varier λ de 0 à une valeur très grande.

Il est important de noter que, en conséquence, les quantificateurs ne restent pas fixes d'un choix de filtres à l'autre, mais s'adaptent optimalement en fonction des filtres utilisés et de l'image.

Le choix des débits binaires alloués aux différents quantificateurs possibles associés aux sous-images a été pris assez large : de 0 à 8 bpp pour chaque sous-image, avec un pas de 0.2 bpp, ce qui permet d'obtenir un nombre de points optimaux suffisamment dense sur les courbes donnant les résultats. L'allure de l'allocation de bits obtenue à l'optimum, dans les différentes sous-images, est très variable suivant le débit binaire total désiré.

Pour un débit binaire autour de 1 bpp, par exemple, on observe que l'allocation optimale de bits, obtenue automatiquement ici, est en accord avec celle généralement imposée pour de tels

débits [3]. Le nombre optimal de décompositions effectuées dans la TOD est, dans ce cas, souvent égal à 4. L'algorithme d'allocation optimale décide de coder la sous-image passe-bas sur un grand nombre de bits (en général 8 bpp, qui correspond au nombre de bpp pour l'image initiale). Il aurait été peut être préférable de rendre possible, pour cette sous-image, une quantification sur un nombre de bits supérieur à 8 bpp (mais les programmes de simulations utilisés auraient dues être partiellement ré-écrites pour ce cas). Les autres sous-images sont codées sur un nombre de bpp décroissant au fur et à mesure que le niveau de décomposition j décroît (ou que la taille des sous-images croît). En général, les sous-images «passe-haut/passe-haut» sont codées sur moins de bits que les autres sous-images au même niveau de décomposition. Enfin, la sous-image «passe-haut/passe-haut» correspondant au premier niveau de décomposition est souvent ignorée par l'algorithme (codée sur 0 bpp).

8.1.5 Qualité visuelle et critère quadratique

On a observé les résultats visuellement sur l'image reconstruite à l'aide d'un moniteur/processeur TRIDYN (de la société INFOROP) spécialisé pour le traitement d'images. En ce qui concerne la qualité visuelle de l'image reconstruite, on considère généralement que le critère d'erreur quadratique, que l'on utilise ici pour donner les résultats, est mal adapté au critère visuel subjectif.

Cependant, les comparaisons effectuées ici sont faites sur des distorsions minimisées par rapport à l'image et à chacun des filtres. On a, dans ce cadre, constaté que le critère quadratique choisi est finalement un «bon» critère visuel. En effet, lorsqu'on observe les images reconstruites pour un débit binaire permettant une *distorsion tout juste perceptible* pour l'image présentant le meilleur rapport signal à bruit, on constate que, d'une façon générale, le critère quadratique est la qualité visuelle *vont dans le même sens*, même pour des différences en rapport signal à bruit de 1 à 2 dB, ou des différences d'artéfacts sont tout juste visibles en effectuant un zoom sur des portions de l'image. En deçà de 1 dB de différence, il est très difficile de percevoir une amélioration sur une image ou sur l'autre si les filtres sont caractéristiques comparables. (Par contre, des différences de 5 dB «sautent aux yeux».)

Comme la plupart des résultats décrits ci-dessous diffèrent de quelques dB seulement en rapport signal à bruit (pour un taux de compression donné), il est difficile de montrer, par des reproductions photographiques, des différences notables dans le cas où l'on s'intéresse à des distorsions tout juste perceptibles sur écran (différences de 1 à 2 dB autour de 30 à 35 dB). On pourrait montrer les images pour des débits plus faibles, conduisant à des artéfacts bien visibles sur toutes les images comparées. Mais dans ce cas, on constate qu'il est parfois très hasardeux d'affirmer que telle ou telle image est de plus grande qualité pour des différences en dB aussi faibles. Pour ces raisons, on se limitera, dans ce chapitre, aux courbes qui donnent le rapport signal à bruit en fonction du nombre final de bpp nécessaire pour coder l'image. Néanmoins, il fallait souligner que ces courbes sont en bon accord avec la qualité visuelle observée sur écran.

8.2 Influence de la phase

La figure 8.1 compare les performances de codage pour différents choix de phases dans le cas de filtres orthogonaux ayant la même réponse fréquentielle. L'effet de la phase est presque imperceptible du point de vue de l'énergie de l'erreur à la reconstruction, quel que soit le débit binaire imposé. On a observé des différences de moins de 1 dB à débit binaire donné, indépendamment des autres paramètres des filtres (longueur, bande de transition, etc.), et ceci

(a)

(b)

Figure 8.1: Influence de la phase (cas orthogonal). (a) Courbes obtenues pour l'image BARBARA et pour les 4 solutions "filtres orthogonaux" de longueur $L = 10$, de degré de platitude $K = 3$, et de largeur de bande de transistion $\Delta\omega = 0.1$. Ces filtres ont les mêmes réponses en fréquence mais différentes réponses de phase. (b) Courbes obtenues pour l'image LENA et pour les 8 solutions, ne différant que par la phase, correspondant à $L = 14$, $K = 5$ et $\Delta\omega = 0.14$. Les courbes obtenues sont quasiment indiscernables à l'échelle de la figure.

(a)

Figure 8.2: Influence de la phase (cas biorthogonal). En trait plein, courbe obtenue pour une solution biorthogonale (*cf.* chapitre 5) de paramètres $L = 14$, $K = 5$ (et donc $L' = 22$), $\Delta\omega = 0.14$, d'ordre de régularité de Hölder $r = 1.5$ et $r' = 2.1$. En trait pointillé, courbe obtenue pour la solution orthogonale correspondant aux mêmes paramètres (c'est à dire une des courbes de la figure 8.1 (b)), ayant pour ordre de régularité $r = 1.7$ et quasiment la même atténuation en bande passante (-25 dB).

sur plusieurs images. Selon le taux de compression, ceci correspond visuellement à des différences quasiment imperceptibles, ou pour lesquelles déterminer une préférence serait hasardeux. Bien que ces filtres orthogonaux ne peuvent être à phase linéaire (pour $L > 2$), un des filtres utilisé pour produire la figure 8.1 a une variation maximale de retard de groupe ne dépassant pas 1/2 échantillon, ce qui est très proche de la phase linéaire.

On peut alors se demander si des filtres réellement à phase linéaire peuvent améliorer la situation. Pour obtenir la phase linéaire, il est nécessaire de relâcher la contrainte d'orthogonalité. A l'aide de la méthode décrite au chapitre 5, on a calculé des solutions biorthogonales $(G(z), G'(z))$ à phase linéaire en s'approchant le plus possible du cas orthogonal, où $G(z)$ présente quasiment le même gabarit que le filtre orthogonal calculé à partir des mêmes paramètres. Les résultats sont présentés à la figure 8.2. On remarque, là encore, que le fait de disposer de filtres presque orthogonaux mais à phase linéaire ne produit pas un résultat très différent du cas orthogonal correspondant, aussi bien pour la courbe de la figure 8.2 que pour la qualité visuelle observée.

La conclusion, qui peut paraître surprenante pour une application de codage d'images, et qu'il n'est donc *a priori* pas justifiable, en tout cas vis-à-vis du critère de distorsion, de préférer des filtres biorthogonaux à phase linéaire à des filtres orthogonaux qui ne sont pas à phase linéaire. Dans toutes nos expérimentations, la phase des filtres n'a joué quasiment aucun rôle. On en déduit que, si la phase a effectivement un rôle à jouer, c'est plutôt en liaison avec des critères perceptuels.

Dans la suite, on se bornera donc à présenter des résultats pour les filtres orthogonaux,

pour lesquels les méthodes de calcul développés au chapitre 5 sont bien maitrisés et fournissent un compromis réellement optimal entre régularité et sélectivité en fréquence. (Arbitrairement, on prendra toujours comme solutions celles dont la phase est la plus proche de la phase linéaire.)

De plus, en conséquence de ce qui vient d'être dit, on peut penser que si la propriété de régularité a effectivement une influence pour la suite des expérimentations, sa mesure ne devrait pas dépendre de la phase du filtre : autrement dit, le paramètre important serait alors l'ordre de régularité au sens de Sobolev et non pas au sens de Hölder (*cf.* chapitre 4).

8.3 Influence de la régularité/sélectivité en fréquence

Nous faisons ici un certain nombre d'expérimentations, visant à faire varier un paramètre de filtre à la fois, afin d'étudier l'influence de la régularité par rapport à celle de la sélectivité en fréquence des filtres.

8.3.1 Degré de platitude variable

Nous commençons par faire varier le degré K de «platitude» des filtres (c'est à dire le nombre de zéros à la fréquence de Nyquist dans le filtre passe-bas), les autres paramètres (longueur et bande de transition) étant fixés. Des résultats sont montrés dans la figure 8.3 pour des filtres orthogonaux calculés avec le critère de Tchebichev (chapitre 5) et en restant en quantification scalaire.

Clairement, le compromis régularité/sélectivité en fréquence influe sur les résultats de compression, pour de nombreux débit binaires imposés. On observe en fait que *le choix de filtres plus réguliers, et donc, moins sélectifs en fréquence, est meilleur pour une longueur donnée.* Ceci peut sembler surprenant, mais a été observé sur différentes images, en utilisant différents critères pour le débit binaire final, et en se basant à la fois sur les courbes et sur la qualité visuelle de l'image reconstruite. En fait, ces observations sont en accord avec la remarque faite notament par Kronander [4] qu'une «bonne» sélectivité en fréquence n'est pas essentielle pour les performances de codage, tout au moins dans ce cadre précis de compression d'image fixes.

En regardant de plus près la figure 8.3, on s'aperçoit qu'il apparait deux catégories bien distinctes de filtres :

1. Ceux dont la réponse fréquentielle ne s'annule pas à la fréquence de Nyquist (cas où $K = 0$). Les filtres sont alors très sélectifs en fréquence, mais pas réguliers du tout (le processus d'itération des filtres diverge – *cf.* chapitre 4). La courbe de rapport signal à bruit obtenue se trouve en dessous des autres avec une différence de l'ordre de 5 à 10 dB. Dans ce cas, un artéfact gênant apparaît très visiblement sur l'image reconstruite, même pour des filtres très atténuées dans la bande passante (jusqu'à 40 dB). Ce type d'artéfact a été également observé dans [4]. Il peut s'interpréter en disant qu'une partie de l'information passe-bas, qui est essentielle pour coder l'image, passe dans les coefficients en ondelettes (sous-images passe-bande), qui, eux, sont parfois quantifiés très grossièrement afin de réduire le débit.

2. Les filtres réguliers ($K > 0$). Les performances pour tous ces filtres ne comportent que des différences de l'ordre de 1 à 2 dB. Néanmoins, on a observé que la qualité visuelle augmente effectivement avec K (c'est à dire lorsque la sélectivité en fréquence décroît). Dans la plupart les cas observées, ce sont donc les filtres de Daubechies (valeur de K maximale) qui donnent les meilleures performances.

(a)

(b)

Figure 8.3: Influence de la régularité (platitude). L'image codée est ici BARBARA, avec quantifiaction scalaire uniquement. (a) Courbes obtenues pour les 3 familles de filtres orthogonaux de longueur $L = 10$ et largeur de bande de transition $\Delta\omega = 0.0625$, correspondant aux valeurs $K = 1$, 3 et 5. (b) Courbes obtenues pour les 4 familles de filtres orthogonaux de longueur $L = 12$ et largeur de bande de transition $\Delta\omega = 0.0625$, correspondant aux valeurs $K = 0$, 2, 4 et 6. On observe, dans tous les cas, que le rapport signal à bruit augmente globalement avec K.

(a)

(b)

Figure 8.4: Influence du type de quantification. L'image codée est LENA, et les 4 filtres utilisées sont orthogonaux, de longueur $L = 12$, de largeur de bande de transition $\Delta\omega = 0.0625$, correspondant à $K = 0$, 2, 4 et 6. (a) Quantification scalaire. (b) Quantification vectorielle sur le réseau D_4 (pour certaines sous-images).

Figure 8.5: Influence du critère d'optimisation des filtres. Ces courbes, obtenues sur l'image LENA, correspondent à des filtres calculés grâce au critère «quadratique» du chapitre 5. Les paramètres sont $L = 12$, $\Delta\omega = 0.0625$, $K = 0$, 2, 4 et 6. Les courbes sont donc à comparer avec la figure 8.4 (a), où les filtres sont calculés avec un critère de Tchebichev.

Des comportements similaires ont été observés dans le cas d'une quantification vectorielle sur réseau (figure 8.4). La quantification vectorielle, bien qu'apportant un intérêt potentiel du point de vue du gain apporté par les quantificateurs et surtout par les codeurs entropiques (*cf.* chapitre 7), réagissent de la même façon du point de vue des paramètres de la transformée, qui est celui considéré ici. Dans la suite, on se limitera donc au cas de la quantification scalaire.

Les mêmes comportements ont été observés également pour des filtres calculés avec un autre critère d'optimisation (figure 8.5). Comme on l'a vu au chapitre 5, ces filtres sont globalement plus réguliers, mais moins sélectifs en fréquence, que les filtres (calculés avec le critère de Tchebichev) utilisés dans les autres figures.

Comparons la figure 8.5 avec la figure 8.4 (a) : dans le cas de la figure 8.5, les courbes, pour $K < 6$, sont globalement plus élevées que dans la figure 8.4 (a), et correspondent à des régularités plus élevées et une sélectivité en fréquence plus faible (Les courbes les plus élevées dans chacune des figures (correspondant à $K = 6 = L/2$) sont les mêmes, car les filtres coïncident avec celui de Daubechies de longueur 12, qui ne dépend pas du critère d'optimisation considéré pour le calcul des filtres.) De plus, l'influence des valeurs de K strictement positives les unes par rapport aux autres est moins nette, ce qui correspond à des valeurs de régularité plus proches que dans le cas de la figure 8.4 (a). Là encore, les observations faites plus haut se reflètent sur les courbes obtenues. Dans la suite, on se contentera de considérer des filtres calculés avec le critère de Tchebichev.

8.3.2 Longueur variable

La figure 8.6 illustre la dépendence de la longueur des filtres utilisés sur les performances de compression, lorsqu'on a fixé les autres paramètres, et notamment l'allure de la réponse fréquentielle du filtre en fixant $L/2 - K$, qui correspond aux nombres d'oscillations dans cette

(a)

(b)

Figure 8.6: Influence de la longueur. (a) Courbes obtenues pour l'image BARBARA, et pour des filtres satisfaisant à $K = L/2-2$ de différentes longueurs ($L = 4$ à 18), et même largeur de bande de transition $\Delta\omega = 0.0625$. Les courbes sont globalement plus élevées au fur et à mesure que L croît. (b) Courbes obtenues pour l'image LENA, et pour des filtres de Daubechies ($K = L/2$) de différentes longueurs ($L = 2$ à 18). Ici, le rapport signal à bruit commence par augmenter globalement pour les premières longueurs (jusqu'à $L \leq 12$) jusqu'à ce qu'une "asymptote" soit atteinte (les courbes sont même légèrement plus basses ensuite).

réponse (*cf.* chapitre 5). De façon générale, les performances (ainsi que la qualité visuelle à débit binaire total donné) sont meilleures lorsque cette longueur croît (ce qui fait également croître la régularité). Néanmoins une asymptote est rapidemment atteinte : on n'obtient pas d'amélioration sensible dès que la longueur dépasse $L = 10$ ou 12, ce qui correspond à des ordres de régularité inférieurs à 2. Utiliser donc des filtres très réguliers, et donc longs, ne semble pas être très utile pour ce schéma de compression.

8.3.3 Bande de transition variable

On compare ici les performances du codage pour différentes bandes de transition imposées sur les filtres. La figure 8.7 montre qu'il se produit une certaine amélioration lorsque l'on augmente la largeur de bande de transition $\Delta\omega$, en supposant qu'il y ait suffisamment de degrés de liberté dans le calcul des filtres pour que ce paramètre ait une influence (nous avons vu au chapitre 5 que les filtres de Daubechies, par exemple, ne dépendent pas de $\Delta\omega$). La dépendance de $\Delta\omega$ sur les performances est en fait d'autant plus importante que le nombre de degré de liberté, $L/2 - K$, est grand.

Or, faire croître $\Delta\omega$ améliore l'atténuation dans la bande passante, mais provoque également un «écrasement» plus important de la réponse fréquentielle du filtre passe-bas autour de la fréquence de Nyquist si $K > 0$, ce qui fait en général croître l'ordre de régularité. Il semble donc qu'ici, la propriété utile est l'aptitude de la réponse fréquentielle du filtre passe-bas à venir s'«écraser» à cette fréquence de Nyquist. Une bonne mesure pour quantifier ce phénomène est précisément la régularité (au sens de Sobolev).

La sélectivité en fréquence, quant à elle, demande que la réponse fréquentielle soit atténuée sur toute la bande passante, sans se préoccuper particulièrement du comportement à la fréquence de Nyquist, et demande également que la bande de transition soit la plus étroite possible. La discussion qui précède montre que cette propriété n'est pas particulièrement utile pour notre étude.

8.4 Conclusion

Les résultats obtenus soulignent, par des mesures objectives, l'importance de la propriété de régularité des filtres pour la compression d'images «par ondelettes», en comparaison avec la sélectivité en fréquence et la phase, tout du moins pour des filtres relativement courts ($L \leq 12$). Dans nos expérimentations, la régularité joue une rôle surtout pour des ordres de régularité ≤ 1, et par conséquent, il est probablement inutile d'augmenter cet ordre de régularité et du même coup, la longueur des filtres.

L'origine de l'importance de la régularité était auparavant attribué à des considérations de type psychovisuel (*cf.* les *a priori* sur l'utilité de la régularité présentés au chapitre 4). Les résultats présentés confirment les observations déjà faites, mais ont été établis de façon plus rigoureuse, en se basant sur une notion mathématique et des critères objectifs.

Bien entendu, les résultats présentés ici ne sont valables que sous les hypothèses faites sur le calcul de filtres et pour le schéma simple de compression d'images fixes utilisé. Nous n'en sommes encore qu'aux préliminaires, et il est possible d'étendre l'approche effectuée ici pour d'autres types de transformées en ondelettes, de quantification et de codage. Néanmoins, dans le cadre restreint choisi, l'étude est suffisamment exhaustive pour servir de la valeur de référence.

(a)

(b)

Figure 8.7: Influence des largeurs de bandes de transition. L'image codée est BARBARA, les filtres orthogonaux utilisés ont pour largeur de bande de transition $\Delta\omega = 0.0625$, 0.1, et 0.14. (a) Filtres de longueurs $L = 8$ et $K = 2$. (b) Filtres de longueurs $L = 12$ et $K = 2$. Les courbes s'élèvent globalement au fur et à mesure que $\Delta\omega$ croît.

Bibliographie

[1] O. Rioul, "On the choice of "wavelet" filters for still image compression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Minneapolis, MN), Apr. 1993. To appear.

[2] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 1980, pp. 291–294.

[3] M. Antonini, *Transformée en ondelettes et compression numérique des images*, PhD thesis, Université de Sophia-Antipolis, 1991.

[4] T. Kronander, *Some aspects of perception-based image coding*, PhD thesis, Linköping University, Sweden, 1989.

Chapitre 9

Conclusion générale et perspectives

A conclusion is a place where you got tired of thinking.
— MATZ'S MAXIM
You always have to give up something you want for something you want more.
— CHRIS' COMMENT
When there's a will, there's a won't.
— GUALTIERI'S LAW OF INERTIA

Le mot-clé de ce mémoire de thèse est «régularité». La plupart des études que nous avons menées porte sur ce concept. Ce nouveau critère sur les filtres utilisés dans un système de compression par ondelettes/banc des filtres a connu récemment un intérêt grandissant. Nous avons relié, par une approche théorique de la décomposition multi-résolution, la théorie des bancs de filtres et celle des ondelettes, afin de se convaincre que ce critère consitue *a priori* l'apport pratique essentiel des ondelettes dans le domaine du codage. La régularité est pressentie importante, parce que reliée à des problèmes perceptuels par des arguments heuristiques.

Afin d'étudier de façon la plus objective possible cette notion de régularité en rapport avec d'autres propriétés de filtres, nous avons cherché tout d'abord à la quantifier précisément, en se basant sur des définitions mathématiques. Nous avons trouvé des estimations de régularité optimales et insisté sur leur application pratique, en proposant un certain nombre de méthodes de calcul de filtres réguliers, qui permettent d'exploiter des bons compromis entre régularité et sélectivité en fréquence.

L'étude s'est orientée ensuite vers la réalisation pratique d'un schéma de codage, où l'on a tenté de mesurer, à l'aide d'un critère objectif et bien défini (distorsion quadratique), l'influence de la régularité sur le gain de compression. A cette fin, nous avons cherché à effectuer une étude aussi exhaustive que possible, en faisant varier chaque propriété des filtres de la transformée plus ou moins indépendamment les unes des autres. Nous avons ainsi pu montré, d'une façon la plus objective possible, l'utilité de la régularité dans certains cas de figure.

Cette étude a soulevé de nombreuses orientations et extensions possibles pour chacun des points étudiés. Elles ont été regroupées à la fin de chaque chapitre, et l'on ne s'étendra pas à nouveau sur celles-ci.

En ce qui concerne l'application du concept de régularité et d'autres propriétés de filtres à la compression d'images fixes, l'approche effectuée dans cette thèse s'est donné un cadre relativement restreint, mais donne les outils de base permettant d'étendre cette étude à des systèmes de compression plus généraux. Par conséquent, comme perspective, l'utilisation de

transformées offrant plus de flexibilité d'analyse que la TOD séparable nous paraît intéressante à étudier par une approche similaire.

On pense en particulier aux décompositions multi-résolution liée aux bancs de filtres/ ondelettes non-séparables, mais aussi aux décompositions en bancs de filtres à changements d'échantillonnage rationnels et aux paquets d'ondelettes, qui offrent suffisamment de souplesse pour s'adapter éventuellement à des critères psycho-visuels. Ceux-ci pourraient, grâce à la théorie des ondelettes, être reliés à des notions mathématiques du type «régularité».

Partie IV

Appendices

Appendice A

Articles

Nothing ever goes away.
— COMMONER'S LAW OF ECOLOGY
If it's imcomprehensible, it's mathematics.
— CERF'S 4TH EXTENSION TO THE HANDY GUIDE TO MODERN SCIENCE

On reproduit ici quelques articles écrits ou co-écrits par l'auteur de cette thèse. Ces articles sont publiés ou soumis pour publication, et ont directement trait à cette thèse : ils fournissent des éléments essentiels à son développement, auxquels les chapitres antérieurs se réfèrent.

A.1 Wavelets and Signal Processing

par Olivier Rioul et Martin Vetterli.
Publié dans IEEE Signal Processing Magazine, Octobre 1991.

A.2 A Discrete-Time Multiresolution Theory

par Olivier Rioul.
A paraître dans IEEE Transactions on Signal Processing, Août 1993.

A.3 Simple Regularity Criteria for Subdivision Schemes

par Olivier Rioul.
Publié dans SIAM Journal on Mathematical Analysis, Novembre 1992.

A.4 A Remez Exchange Algorithm for Orthonormal Wavelets

par Olivier Rioul et Pierre Duhamel.
Soumis pour publication dans IEEE Transactions on Circuits and Systems–II. Analog and Digital Signal Processing en Février 1993.

A.5 Fast Algorithms for Discrete and Continuous Wavelet Transforms

par Olivier Rioul et Pierre Duhamel.
Publié dans IEEE Transactions on Information Theory, Mars 1992.

A.6 On the Choice of "Wavelet" Filters for Still Image Compression

par Olivier Rioul.
A paraître dans les *proceedings* d'ICASSP'93.

Appendice B

Bibliographie de l'auteur

You never find an article until you replace it.
— HARPER'S MAGAZINE LAW
The progress of science varies inversely with the number of journals published.
— PARKINSON'S LAW

Cet appendice regroupe les articles écrits ou co-écrits par l'auteur. La plupart rendent compte de travaux effectués pour cette thèse. (Parmi eux, les articles B.1 [2], B.2 [2, 3, 4, 6], et B.3 [10], fournissent des éléments essentiels à son développement et sont reproduits dans l'appendice A.) Cependant, un certain nombre se situent, à strictement parler, en dehors du sujet présenté dans ce manuscript : Il s'agit de travaux parallèles (analyse temps-fréquence, Wigner-Ville) ou d'extensions (régularité pour les bancs de filtres rationnels). D'autres extensions font l'objet de travaux en cours.

B.1 Articles "tutoriaux"

[1] Y. Meyer, S. Jaffard, et O. Rioul, "L'analyse par ondelettes," *Pour La Science*, no. 119, pp. 28–37, Sept. 1987.

[2] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, vol. 8, no. 4, pp. 14–38, Oct. 1991.

B.2 Articles publiés dans des revues à comité de lecture

[1] O. Rioul and P. Flandrin, "Time-scale energy distributions—A new class extending wavelet transforms," *IEEE Trans. Signal Processing*, vol. 40, no. 7, pp. 1746–1757, July 1992.

[2] O. Rioul, "A discrete-time multiresolution theory," *IEEE Trans. Signal Processing*, vol. 41, no. 8, Aug. 1993. To appear.

[3] O. Rioul and P. Duhamel, "Fast algorithms for discrete and continuous wavelet transforms," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 569–586, Mar. 1992. (Part II. Special issue on Wavelets and Multiresolution Signal Analysis).

[4] O. Rioul, "Simple regularity criteria for subdivision schemes," *SIAM J. Math. Anal.*, vol. 23, no. 6, pp. 1544–1576, Nov. 1992.

[5] O. Rioul, "Regular wavelets : A discrete-time approach," *IEEE Trans. Signal Processing.* Special issue on Wavelets and Signal Processing, to appear.

[6] O. Rioul and P. Duhamel, "A Remez exchange algorithm for orthonormal wavelets," *IEEE Trans. Circuits Syst. II.* Submitted for publication.

B.3 Articles publiés dans des actes de congrès

[1] C. Dorize, O. Rioul, and A. Chaigne, "Analysis and synthesis of sound signals using a discrete wavelet transform (DWT)," in *13th International Congress on Acoustics*, (Belgrade, Yugoslavia), 1989.

[2] P. Flandrin, B. Vidalie, and O. Rioul, "Fourier and wavelet spectrograms seen as smoothed Wigner-Ville distributions," in *Proc. Int. Colloqueum "Wavelets and Applications"*, Y. Meyer, Ed., (Paris), Masson, Springer-Verlag, 1992, pp. 93–103. Marseille, France, May 1989.

[3] P. Flandrin and O. Rioul, "Wavelets and affine smoothing of the Wigner-Ville distribution," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Albuquerque, NM), Apr. 1990, pp. 2455–2458.

[4] O. Rioul, "Structures and algorithms for the orthonormal discrete wavelet transform," in *Proc. 1990 Digital Signal Processing Workshop*, (New Paltz, NY), Sept. 1990, pp. 3.3.1–2.

[5] O. Rioul, "Fast algorithms for the continuous wavelet transform," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Toronto, Ontario, Canada), May 1991, pp. 2213–2216.

[6] O. Rioul et P. Duhamel, "Transformées en ondelettes discrètes et continues—Comparaison et algorithmes rapides," in *Treizième Colloque GRETSI*, (Juan-Les-Pins, France), vol. 1, Sept. 1991, pp. 193–196.

[7] O. Rioul, "Regular wavelets–Theory and algorithms," in *Proc. Int. Colloqueum "Wavelets and Applications,"* (Toulouse, France, June 1992), 1993. To appear.

[8] O. Rioul, "A simple, optimal regularity estimate for wavelets," in *Proc. European Signal Processing Conf. (EUSIPCO)*, (Brussels, Belgium), vol. II, Aug. 1992, pp. 937–940.

[9] O. Rioul, "A discrete approach to regularity of 1D or 2D wavelets," in *Proc. 1992 Digital Signal Processing Workshop*, (Utica, IL), Sept. 1992, pp. 1.9.1–2.

[10] O. Rioul, "On the choice of 'wavelet' filters for still image compression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Minneapolis, MN), Apr. 1993. To appear.

[11] T. Blu and O. Rioul, "Wavelet regularity of iterated filter banks with rational sampling changes," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, (Minneapolis, MN), Apr. 1993. To appear.

[12] H. L. Bihan, P. Siohan, O. Rioul, et P. Duhamel, "Une méthode simple de calcul de bancs de filtres/ondelettes bi-orthogonales," in *Proc. European Signal Processing Conf. (EUSIPCO)*, Sept. 1993. Soumis pour présentation.

B.4 Rapports techniques

[1] O. Rioul, "Ondelettes," mémoire , Centre de Mathématiques, Ecole Polytechnique, 91128 Palaiseau Cedex, France, Juin 1987.

[2] O. Rioul, "Wigner-Ville representations of signals adapted to shifts and dilations," Tech. Rep. TM 11227-880422-03, AT&T Bell Laboratories, Murray Hill, NJ, Apr. 1988.

[3] Collectif, "Articles sur les transformées en ondelettes soumis par le département ETP," Rapport Technique CRPE/192, CRPE (CNET/CNRS), 38–40, rue du Général Leclerc, 92131 Issy-Les-Moulineaux, France, Avril 1991.

Appendice C

Calcul de filtres : boîtes à outils MATLAB

Other people's tools work only in other people's gardens.
— 1ST LAW OF GARDENING
There's always one more bug.
— LUBARSKY'S LAW OF CYBERNETIC ENTOMOLOGY
It won't work.
— JENKINSON'S LAW
When all else fails, read the instructions.
— CAHN'S AXIOM

Cet appendice regroupe les fichiers de description (en anglais) de deux boîtes à outils MATLAB, conçues principalement pour le calcul de filtres RIF réguliers, orthogonaux ou bi-orthogonaux à phase linéaire. Leur utilisation nécessite la version 3.5 de MATLAB, et des boîtes à outils standard MATLAB (*MATLAB Toolbox, Signal Processing Toolbox* et *Optimization Toolbox*).

Le succès de MATLAB atteint outre-atlantique grandit en Europe, et confère à ces outils un intérêt potentiel pour les chercheurs intéressés par le calcul de filtres "ondelettes". Ces outils ont été proposés pour leur intégration dans une bibliothèque de programmes MATLAB réunie par le GDR-134 «Traitement du Signal et des Images».

C.1 DSP Toolbox

Cette boîte à outils contient principalement des outils de visualisation et des algorithmes classiques de calcul de filtres RIF.

IMPULSE

IMPULSE(H) plots impulse response of FIR filter described by vector H.
IMPULSE(H,'Title') also writes 'Title' at the top of the plot.

See MAGNITUDE, PHASE, GROUPDELAY.

MAGNITUDE

MAGNITUDE(H) plots magnitude (frequency) response of FIR filter described by vector H. By default, a linear scale is assumed. MAGNITUDE(H,1) plots magnitude on a dB scale. MAGNITUDE(H,2) plots magnitude on a dB scale, assuming the filter corresponds to a squared magnitude.

For dB plots, the pass-band gain is assumed to be equal to 1. To change its value, enter the gain as a third argument. Adding 'Title' as an argument also writes 'Title' at the top of the plot.

See IMPULSE, PHASE, GROUPDELAY.

PHASE

PHASE(H) plots phase response of FIR filter described by vector H.
PHASE(H,'Title') also writes 'Title' at the top of the plot.

See IMPULSE, MAGNITUDE, GROUPDELAY.

GROUPDELAY

GROUPDELAY(H) plots group delay response of FIR filter described by vector H.
GROUPDELAY(H,'Title') also writes 'Title' at the top of the plot.

See IMPULSE, MAGNITUDE, PHASE.

RESPONSES

RESPONSES(H) plots impulse, magnitude, phase and group delay responses of FIR filter described by vector H. Optional arguments 1, 2 (dB scales) and gain are the same as for MAGNITUDE.

See IMPULSE, MAGNITUDE, PHASE, GROUPDELAY.

ZEROES

ZEROES(R) plots zeroes (roots) described by column vector R. ZEROES(R,'Title') also writes 'Title' at the top of the plot. Do not confuse with MATLAB standard ZEROS.

See RESPONSES, ZEROS.

ATTENUATION

ATTENUATION(H,B) gives stop-band attenuation, in dB, of half-band FIR filter with transition band $(0.25 - B/2, 0.25 + B/2)$, where B is the normalized transition bandwidth.

ATTENUATION(H,B,2) assumes H corresponds to a square magnitude.
[A,δ]=ATTENUATION(...) also gives the tolerance δ in the stop band (assuming gain is 1).

See MAGNITUDE.

PHASEVAR

PHASEVAR(H) gives (positive) total group delay variation of filter described by vector H. This is a measure of phase distortion : If it is close to zero, then H is close to being linear phase. If it is zero, then the filter is linear phase.

CAUTION : This measure is generally spoiled by zeros in the stop band's magnitude response, causing diracs to appear in the group delay. However, phase transitions by π should not be a problem.

[var1,var2]=PHASEVAR(H) gives group delay variations in the first $(0, \pi/2)$ and second $(\pi/2, \pi)$ half band, respectively. This measure is more significant for half-band filters (var1 for low-pass filters, var2 for half-band filters) and avoids the problem of Diracs.

Check the result with GROUPDELAY, PHASE, or ZEROES.

See RESPONSES, PHASE, GROUPDELAY, MAGNITUDE, ZEROES.

FACTOR

Orthogonal analysis/synthesis filter factorization (Riesz lemma).

FACTOR(ZP) plots the complex zeroes ZP of a strictly linear phase low-pass «product» filter $P(z)$ (= odd-tap low-pass zero- phase filter whose zeroes on the unit circle have even multiplicity), then waits for zeroes to be selected by clicking the mouse on the screen, and finally gives the vector describing filter H such that $P(z)$ is the convolutional square of $H(z)$, i.e.,

$$H(z)H(z^{-1}) = P(z) \text{ or } |H(e^{i\omega})|^2 = P(e^{i\omega}).$$

(Zeroes on the unit circle are not selectable : They are retained with twice less multiplicity.)

ZP can be obtained from design routines like REMEZWAV, etc. The resulting filter is normalized such that NORM(H)=1, but this may be inappropriate for e.g. MAGNITUDE (gain problem). (Note that checking the result with ZEROES will add computational round-off errors.)

See MAGNITUDE, ZEROES, REMEZWAV.

FACTORALL

Orthogonal analysis/synthesis filter factorizations (Riesz lemma).

Given the complex zeroes ZP of a strictly linear phase low-pass «product» filter $P(z)$ (= odd-tap low-pass zero-phase filter whose zeroes on the unit circle have even multiplicity), FACTORALL(ZP) gives all line-vectors describing all real filter solutions H such that $P(z)$ is the convolutional square of $H(z)$, i.e.

$$H(z)H(z^{-1}) = P(z) \text{ or } |H(e^{i\omega})|^2 = P(e^{i\omega}).$$

ZP can be obtained from design routines like REMEZWAV, etc.

The resulting filters, normalized such that NORM(H)=1, are output in matrix form [H_0 ; H_1 ; H_2 ; H_3 ; ...]. (CAUTION : indexes range from 0). This routine also plots zeroes for each solution H_n. H_n is constructed by selecting zeros outside (1) or inside (0) the unit disk as absolute angle of zeros increases. For example, $n = 1011$ in base 2 means (in reverse order) : First outside, then outside, then inside, then outside. H_0 is the minimum-phase solution. Last H_n ($n = 2^{(\cdots)} - 1$) is the maximum-phase solution. (Note that the second half of solutions are the time-reversals of the first half.)

See FACTOR, PHASESORT.

PHASESORT

PHASESORT(H) given a matrix whose lines describe filters H_n of the same magnitude response (as given by FACTORALL), returns this matrix, sorted from the closest to the farthest from linear phase. A filter is closer to linear phase if its group delay deviation in the pass-band, as given by PHASEVAR, is smaller.

[HSORTED,INDEX]=PHASESORT(H) also gives the corresponding indexes of filters $n = 0, 1, \ldots$ (=phase codes as described in FACTORALL).

[HSORTED,INDEX,VAR]=PHASESORT(H) also gives the group delay variations.

See FACTORALL, GROUPDELAY, PHASEVAR.

REMEZZ

H=REMEZZ(L,K,B,ω,C) gives optimum low-pass filter described by vector H, of length L, with K zeroes at the Nyquist frequency, normalized transition bandwidth B, normalized frequency offset $\omega = (\omega_p+\omega_s)/2-0.25$, and (optional) weight coefficient $C = \delta_1/\delta_2$ (pass-band attenuation is greater as C is larger).

[H,δ_1]=REMEZZ($\ldots$) also gives the maximum deviation δ_1 in the passband ($\delta_2 = \delta_1/C$, attenuation is $-20\log_{10}(\delta_1/C)$

REMEZZ(L,K,B,ω,C,0) uses global variable glob as initial guess of alternations and set glob for next call (to minimize the number of iterations). To reset the guess, set glob to [].

C.2 Wavelet Toolbox

Cette boîte à outils contient les algorithmes originaux de calcul de filtres réguliers et d'estimation de régularité, ainsi que des outils de visualisation d'ondelettes et dérivées.

WAVELET

WAVELET(H) plots the scaling function (father wavelet) associated to low-pass FIR filter described by vector H. WAVELET(H,G) plots the (mother) wavelet associated to low-pass and high-pass filters described by vectors H and G, respectively.

More generally, WAVELET(H,G) plots the limit function obtained by an iterative "subdivision" procedure whose initial sequence is G. In particular, WAVELET(H) sets $G = H$.

By default, the number of iterations is the maximum permissible on this computer. WAVELET(H,iter) or WAVELET(H,G,iter) forces the number of iterations to iter (for a quicker plot). WAVELET returns the actual number of iterations performed.

[iter,wavlet]=WAVELET(H,$\ldots$) also returns the values of the limit function in wavlet, obtained after iter iterations.

See DERIV.

DERIV

DERIV(H,n) plots the nth derivative of the scaling function (father wavelet) associated to low-pass FIR filter described by vector H. DERIV(H) sets n to 1. DERIV(H,G,n) plots the nth derivative of the (mother) wavelet associated to low-pass and high-pass filters described by vectors H and G, respectively. More generally, DERIV(H,G,n) plots the nth derivative of the limit function of an iterative "subdivision" procedure whose initial sequence is G. DERIV(H,n) sets $G = H$.

By default, the number of iterations is the maximum permissible on this computer. DERIV(H,n,iter) or DERIV(H,G,n,iter) forces the number of iterations to iter (for a quicker plot). DERIV returns the actual number of iterations performed. [iter,wavlet]=DERIV(H,...) also returns the values of the limit function in wavlet, obtained after iter iterations.

See WAVELET.

MOTHER

MOTHER(H) returns G, the orthogonal complement high-pass filter of low-pass FIR filter described by vector H. If H corresponds to the orthogonal father wavelet, then G corresponds to the orthogonal mother wavelet. For example, WAVELET(H,MOTHER(H)) plots the orthogonal wavelet.

See WAVELET, DERIV.

REG

REG(H) gives Hölder Regularity of filter described by vector H, as estimated by a sharp upper bound. (The regularity may be negative.) Removal of zeroes in H at half the sampling frequency is done assuming remainder values $< 10^{-3}$. REG(H,Z) forces the number Z of such zeros.

See SOBREG.

SOBREG

SOBREG(P) gives (optimal) Sobolev regularity order of low-pass FIR filter H of even length L, which is the «square root» of the input filter $P(z)$, a strictly linear phase filter of length $2L - 1$ (see FACTOR). (The regularity may be negative.) Removal of zeroes in $P(z)$ at half the sampling frequency is done assuming remainder values $< 10^{-3}$. SOBREG(P,Z) forces the number Z of such zeros in $H(z)$.

See FACTOR, REG.

REMEZWAV

REMEZWAV(L,K,B) gives impulse response of maximally frequency selective $P(z)$, product filter of paraunitary filter bank solution $H(z)$ of length L satisfying K flatness constraints (wavelet filter). The normalized transition bandwidth is given as B. This argument is optional if $K = L/2$ (Daubechies filter).

[P,R]=REMEZWAV(L,K,B) also gives the roots of $P(z)$, which can be used to determine $H(z)$.

[P,R,δ]=REMEZWAV(L,K,B) also gives maximum deviation δ, if $K < L/2$.

L2WAV

L2WAV(L,K,B) produces convolutional square $P(z)$ of orthogonal low-pass filters $H(z)$ (see FACTOR), of length $2L-1$, with K zeros at half the sampling frequency (vanishing moments), best approximating ideal filter in the L^2 sense for normalized transition bandwidth $B < 0.5$. L must be even and $K \leq L/2$. L2WAV(L,$L/2$,B) is equivalent to REMEZWAV(L,$L/2$).

P is computed using linear programming (see LP), and is normalized such that the sum of coefficients equals 1. [P,PZ]=L2WAV(...) also gives the zeroes PZ of filter P, for use with FACTOR.

See FACTOR, FACTORALL, LP (Optimization Toolbox).

BIORTH

BIORTH(H) gives the biorthogonal complementary filter H' of H, a low-pass symmetrical filter. If L =length(H) is even, L' =length(H')=length(H). If L =length(H) is odd, L' =length(H')=length(H)-2.

BIORTH(H,K) gives H' with K zeroes at $z=-1$: $L-K$ must be odd, otherwise $K=K+1$. $\Rightarrow L' = L + 2(K-1)$ (default is $K=0$ if L odd, $K=1$ if L even)

NOTE : if H has pass-band gain 1, H' will have pass-band gain 2

BIWAV

BIWAV [H,H']=BIWAV(L,K,B) gives frequency selective biorthogonal filters described by vectors H and H', with K zeroes at $z=-1$. length(H)= L, while length(H')= $L+2(K-1)$. H is a Remez filter whose transition bandwidth is B. The tolerance factor $C=\delta_1/\delta_2$ and the bandwidth shift $\delta\omega$ for filter H are printed. H' is the filter closest to H in the quadratic sense.

NOTE : global variable glob is used (see REMEZZ).

See REMEZZ, BIORTH, FMINU or FMIN (Optimization Toolbox).

It's better to have a horrible ending than to have horrors without end.
— MATSCH'S LAW