



HAL
open science

Synthèse de bancs de filtres adaptés, application à la compression des images.

Hocine Bekkouche

► **To cite this version:**

Hocine Bekkouche. Synthèse de bancs de filtres adaptés, application à la compression des images.. Traitement du signal et de l'image [eess.SP]. Université Paris Sud - Paris XI, 2007. Français. NNT : . tel-00345288

HAL Id: tel-00345288

<https://theses.hal.science/tel-00345288>

Submitted on 8 Dec 2008

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.



UNIVERSITÉ PARIS-SUD XI
Faculté des Sciences d'Orsay

THÈSE DE DOCTORAT

SPÉCIALITÉ : AUTOMATIQUE ET TRAITEMENT DU
SIGNAL

École Doctorale Science et Technologie de l'Information
des Télécommunications et des Systèmes

Présentée par :

Hocine BEKKOUCHE

Sujet

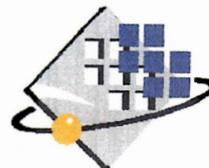
**Synthèse de bancs de filtres adaptés,
application à la compression des images.**

Soutenue le 18 juin 2007 devant les membres du jury

M. Pierre DUHAMEL	Directeur de recherche CNRS	Président
M. Olivier DÉFORGES	Professeur à l'INSA de Rennes	Rapporteur
M. Rémy PROST	Professeur à l'INSA de Lyon	Rapporteur
M. Jacques OKSMAN	Professeur à SUPÉLEC	Directeur de thèse
M. Michel BARRET	Enseignant-chercheur à SUPÉLEC	Co-directeur de thèse



Université Paris Sud XI
U.F.R. Scientifique d'Orsay
DIVISION de la RECHERCHE
Service des Etudes Doctorales
vvvvvvvvvv



N° d'ordre :

8674

DOCTORAT

(arrêté du 25 Avril 2002)

RAPPORT DE SOUTENANCE

établi par le Président du Jury

Thèse de Doctorat : Spécialité : AUTOMATIQUE ET TRAITEMENT DU SIGNAL

de M. BEKKOUCHE Hocine

Sujet : SYNTHÈSE DE BANCS DE FILTRES ADAPTÉS, APPLICATION À LA COMPRESSION DES IMAGES.

Date de soutenance : Le 18 Juin 2007

Mr Bekkouche a effectué un exposé parfaitement clair et didactique, synthétisant bien ses travaux et mettant en avant ses contributions, qui sont réelles, sur un sujet difficile. Son aisance à l'oral a bien mis en avant la qualité de sa démarche, argumentée et rigoureuse.
Il a répondu avec pertinence aux nombreuses questions du jury, sans en éluder les difficultés, démontrant ainsi sa maîtrise du sujet, un recul certain sur le domaine concerné, et la qualité de sa communication.

Date et Signature des
membres du Jury :

le 18/06/07

R. DUHAMME

R. PROST

Fl. BARRET

J. OKSMA

O. DEFORGES

La force consiste à affronter le péril et à supporter
le travail avec mûre réflexion. Elle renferme
la magnificence, la confiance, la patience, la persévérance.
La magnificence consiste à méditer et à exécuter
des choses grandes et sublimes, avec une large et généreuse disposition
de l'âme. Par la confiance, l'esprit place en lui-même
un espoir puissant et assuré pour les choses grandes et honnêtes.
La patience supporte volontairement et longtemps des choses ardues
et difficiles, en vue de l'honnête ou de l'utile.
La persévérance est une constance inébranlable après juste et mûre réflexion.



*Saint-Augustin, Évêque d'Hippone.
Thagaste (Souk-Ahras) 354
Hippone (Annaba) 430*



Remerciements

J'exprime mes sincères remerciements à Monsieur le Professeur Pierre DUHAMEL, Directeur de Recherches au CNRS, Laboratoire de Signaux et Systèmes, responsable de la division Signaux, pour avoir accepté de présider le jury de cette thèse.

Je tiens également à exprimer mes sincères remerciements à Monsieur Olivier DEFORGES, Professeur des Universités à l'INSA de Rennes ainsi qu'à Monsieur, Rémy PROST Professeur des Universités à l'INSA de Lyon pour l'intérêt qu'ils ont porté à ces travaux en acceptant de les rapporter.

Je remercie chaleureusement Monsieur Jacques OKSMAN, Professeur-Directeur de la Recherche et des Relations Industrielles à Supélec, d'avoir accepté la direction de cette thèse et pour ses valeureux conseils.

Cette thèse n'aurait pas vu le jour sans la patience et la générosité de mon co-directeur de recherche, Monsieur Michel BARRET, Enseignant-Chercheur à Supélec-Metz. Je veux vivement le remercier pour le temps et la confiance qu'il m'a accordés au long de ces années et d'avoir cru en mes capacités.

Je suis profondément reconnaissant à Monsieur Mustapha NADI, Directeur du Laboratoire d'Instrumentation Électronique de Nancy (L.I.E.N.) à l'Université Henri Poincaré, Nancy ainsi qu'à Monsieur Amar ROUANE, Professeur au même laboratoire pour leur soutien et leurs conseils qui m'ont été fort utiles.

Mes sincères remerciements vont également à Messieurs Rémy CLAVERIE, Enseignant-Chercheur, Jean-Pierre CHARLES, Professeur et Nicolas FRESSENGEAS, Professeur, du Laboratoire des Matériaux Optiques, Photonique et Systèmes (LMOPS) pour leur généreuse aide.

Messieurs Joël JACQUET, Délégué à la Recherche et aux Relations Industrielles, Olivier PIET-QUIN, Enseignant-Chercheur, Responsable de l'équipe IMS, Modhaffer SAIDI, Doctorant à Supélec-Campus de Metz et Mohamed HARITI, Post-Doctorant CNES, ont accepté de participer aux présentations de présoutenance et leurs conseils judicieux m'ont été utiles. Je les en remercie.

Mes remerciements s'adressent aussi à tous mes amis pour leur sympathie, leur gentillesse et leur soutien constant dans les moments de joie et comme dans les autres. J'inclus tout particulièrement ceux (rangée par ordre alphabétique), qui, ces dernières années, ont beaucoup comptés pour moi :

- Nedjem AYAT, Ingénieur/Docteur, Chef de projet/Consultant Systèmes d'information, Starlog Conseil, Paris
- Samir BENZAHRA, Sous-Directeur, Europe Optical Product Technical Sales Department, Huawei Technologies Co., Ltd.
- Eugen DEDU, Maître de conférences, Département Génie des Télécommunications et des Réseaux, Université de Franche-Comté.
- Badie GHORAB, Ingénieur, PDG, NTIConsulting M&D, Paris
- Malik HOCINE, Ingénieur/Docteur, Chercheur à Institute of Advanced Telecommunications, Swansea University, Pays de Galles.
- Rachid KAOUACHE, Docteur, Enseignant-Chercheur à l'ENSAM, Aix-en-Provence
- Jalel LAHRECH, Ingénieur, Consultant Tests/Assurance Qualité Saoudi Telecom
- Tarek LAMARA, Docteur, Chercheur R&D, ABB Switzerland Ltd. Amine M'SIR Ingénieur/Docteur, Ingénieur R&D, ACAMAS Electronique
- Michel NAROZNY, Ingénieur/Docteur, Ingénieur de développement, TAGATTITUDE, Paris
- Jalil YAHIA, Ingénieur/Docteur, Responsable bureau d'études et industrialisation, BRACER Biotech

Si Marcel Proust ne croyait pas en l'amitié, c'est parce qu'il n'a pas connu les miens.

Sans oublier les membres de ma belle famille DRISSI qui m'ont comblé de gentillesse.

Mes parents ont été ceux qui m'ont appris beaucoup de précieux théorèmes de la vie, introuvables dans les publications et les livres. C'étaient certainement leurs *daâouat el kheir* qui m'ont guidé pendant toutes les phases de ce projet.

Ma profonde gratitude envers la femme de ma vie MESSA-OUUDA qui a su faire preuve d'une patience que peu pourraient accepter. Ses encouragements et son amour m'ont permis de garder mon optimisme et ma volonté dans les moments de stress. Elle est certainement ma meilleure bénédiction, et ce projet serait resté un rêve sans son soutien. Je lui exprime ici mon amour, mon respect et ma reconnaissance. Je lui dédie ce travail, ainsi qu'à ma fille LYNA-AMBRE notre rayon de soleil, née pendant la préparation de cette thèse.



Résumé

Les travaux développés dans cette thèse portent sur les décompositions multirésolution dans un cadre de *lifting scheme*, appliquées à la compression d'images. Pour commencer, une structure de décomposition consistant en un "*lifting scheme* généralisé" est proposée. Ce schéma permet d'exploiter toute l'information disponible au décodage dans l'étape de prédiction. Cela est rendu possible par l'ajout d'un filtre de prédiction supplémentaire par rapport à la structure classique de *lifting scheme*. Le schéma proposé est ensuite associé à deux méthodes d'adaptation. La première, appelée GAE, suppose une stationnarité globale de l'image, tandis que la seconde, LAE ne suppose qu'une stationnarité locale de l'image. Dans ce dernier cas, les filtres prédictifs sont adaptatifs.

Trois applications de ces méthodes en compression d'images sont ensuite proposées. Dans un premier temps, une comparaison des performances en compression sans perte sur des images de textures, synthétiques, gaussiennes, à stationnarités locale et globale (vérifiant les hypothèses plus haut), est réalisée. Sur ces signaux, les mesures d'entropie d'ordre 1 ont montré que les méthodes proposées offrent en moyenne un gain en codage de 0,5 bpp (GAE) et 0,8 bpp (LAE) par rapport à la décomposition en ondelette (9,7), de 0,8 bpp (GAE) et 1,11 bpp (LAE) par rapport à la (5,3) et de 0,41 bpp (GAE) et 0,65 bpp (LAE) par rapport à la méthode de Gerek et Çetin. La deuxième application concerne le codage sans perte d'images réelles de natures variées. Les gains par rapport à l'état de l'art se sont révélés plus faibles que ceux obtenus pour les images synthétiques. Enfin, la dernière application traite les cas du codage progressif et du codage avec perte. Pour la compression avec pertes, nous avons modifié la méthode LAE pour palier aux problèmes de divergence dus à l'impossibilité au niveau du décodeur de reconstruire les filtres prédictifs à partir d'échantillons quantifiés. Elle se révèle plus efficace que lorsque l'on utilise les filtres usuels de longueur fixe (9,7) et (5,3).

Mots clés : Bancs de filtres adaptés, transformée en ondelettes, *lifting scheme* adapté, filtrage adaptatif, compression sans perte, codage progressif.



Abstract

This thesis is concerned with multiresolution decompositions in a lifting scheme framework as well as its application to image compression. First, a decomposition structure consisting of a "generalized lifting scheme" is proposed, using all information available in the decoding stage for prediction. This new scheme relies on an additional predictive filter which does not appear in classical lifting scheme structures. The proposed scheme is then associated to two adaptation methods. The first one, called GAE, assumes a global stationnarity of the image, while the second one, called LAE assumes only a local stationnarity. In the latter, adaptive filters are used.

A study of the performance in lossless coding is done by using a set of Gaussian synthetic images with local and global stationarities (satisfying the assumptions on the images). Estimations of the first order entropy of the transformed image showed that the proposed methods offer on average an improvement of 0.5 bpp (GAE) and 0.8 bpp (LAE) compared to the (9,7) wavelet transform, 0.8 bpp (GAE) and 1.11 bpp (LAE) compared to the (5,3) wavelet transform and 0.41 bpp (GAE) and 0.65 bpp (LAE) compared to the method of Gerek & Çetin (a method using LMS-based adaptive filters). The second part is concerned with the study of performance in lossless coding of still images with varied features. The obtained gains in bit rate compared to the state of the art are lower than those obtained for the synthetic images. Finally, in the last part, the application to progressive coding and lossy coding is studied. In order to prevent the LAE-divergence phenomenon due to the quantization step in lossy compression, we modified its structure. Measurements of rate/distortion showed that, for high bit rates, that the modified LAE outperforms the (9,7) and (5,3) wavelet transforms.

Keywords : Wavelets and signal-adapted filter banks, signal-adapted lifting scheme, adaptive filtering, lossy/lossless image compression, progressive coding.



TABLE DES MATIÈRES

Notations	v
Introduction générale	1
1 Banc de filtres, ondelettes et <i>lifting scheme</i> adaptés	5
1.1 Introduction	5
1.2 Banc de filtres à reconstruction parfaite	6
1.2.1 Sous- et sur-échantillonnage	6
1.2.2 Analyse et synthèse par bancs de filtres	8
1.2.3 Représentation polyphase d'un banc de filtres	10
1.3 Ondelettes et <i>lifting scheme</i>	12
1.3.1 Analyse multi-résolution et ondelettes	13
1.3.2 <i>Lifting scheme</i>	16
1.4 Banc de filtres optimal en codage avec pertes	18
1.4.1 Le banc de filtres à optimiser	19
1.4.2 Le signal à coder	20
1.4.3 Les quantificateurs	20
1.4.4 Position du problème d'optimisation	21
1.4.5 Cas d'un banc de filtres biorthogonal	21
1.4.6 Cas d'un banc de filtres orthogonal	22
1.5 Banc de filtres, ondelettes et <i>lifting scheme</i> 2-D	24
1.6 <i>Lifting schemes</i> adaptés en compression	24
1.6.1 Adaptation par sélection de l'ordre du prédicteur	24
1.6.2 Adaptation sous contrainte	25
1.6.3 Adaptation pour des signaux WSS avec post-traitement non-linéaire	26
1.6.4 Adaptation basée sur l'algorithme du gradient stochastique	28
1.7 Conclusion	28
2 Compression d'images	31
2.1 Introduction	31

2.2	Compression avec pertes	31
2.2.1	Transformation	32
2.2.2	Quantification	33
2.2.2.1	La quantification scalaire	33
2.2.2.2	La quantification vectorielle	36
2.2.3	Allocation optimale	36
2.2.3.1	Allocation de débit	38
2.2.4	Codage entropique	39
2.2.4.1	Le codage de Huffman	39
2.2.4.2	Le codage arithmétique	40
2.2.4.3	Codage universel	40
2.3	Codage progressif	41
2.3.1	Progressivité en résolution	42
2.3.2	Progressivité en qualité	42
2.4	Compression sans perte	43
2.4.1	Schéma général	43
2.4.2	Prédiction	44
2.4.2.1	Prédiction linéaire	45
2.4.2.2	Réseaux de neurones	47
2.4.2.3	<i>Lifting scheme</i> réversible	47
2.5	Principaux systèmes de compression sans perte	49
2.5.1	LOCO-I	49
2.5.2	CALIC	50
2.5.3	JPEG 2000	51
2.5.4	Autres systèmes à base de <i>lifting schemes adaptés</i>	52
2.6	Conclusion	52
3	Synthèse de bancs filtres adaptés proposée	53
3.1	Introduction	53
3.2	Justification du critère de minimisation de la variance	54
3.3	Structure en <i>lifting scheme</i> généralisé	57
3.3.1	Schéma de principe pour des signaux 1D	57
3.3.2	Extension aux signaux 2D	58
3.3.3	Réversibilité de la structure proposée	60
3.3.4	Décomposition multi-résolution	61
3.3.5	Adaptation de la structure 2D proposée	62
3.4	Méthode d'adaptation globale (GAE)	62
3.4.1	Choix du critère	62
3.4.2	Calcul rapide de la matrice du système	64
3.5	Méthode d'adaptation locale (LAE)	66
3.5.1	Choix du critère	66
3.5.2	Algorithme de décomposition	69
3.6	Application à des signaux synthétiques	69
3.6.1	Construction des signaux	69
3.6.2	Résultats	70
3.7	Conclusion	72

4	Application à la compression sans perte des images réelles	75
4.1	Introduction	75
4.2	Ajustement des paramètres par famille d'images	75
4.2.1	La méthode GAE	75
4.2.1.1	Choix du type de sous-échantillonnage	75
4.2.1.2	Choix des ordres des filtres	76
4.2.1.3	Nombre de niveaux de décomposition	77
4.2.1.4	Utilisation de la transformation S (décomposition SGAE)	78
4.2.1.5	La méthode GAE améliorée (GAEa)	80
4.2.1.6	En-tête du flot de bits	81
4.2.2	La méthode LAE	82
4.2.3	Choix de l'algorithme d'adaptation	82
4.2.3.1	Traitement des bords	84
4.2.3.2	Choix du type de sous-échantillonnage	85
4.2.3.3	Choix de l'ordre des filtres prédicteurs	86
4.2.3.4	Choix du facteur d'oubli	88
4.2.3.5	En-tête du flot de bits	92
4.2.4	Complexité	92
4.3	Performances en compression sans perte	92
4.3.1	Avec mesure de débit par entropie d'ordre 1	92
4.3.2	Avec débit réel	95
4.3.3	Codage progressif en résolution	96
4.3.4	Analyse des résultats	96
4.4	Peux-on faire du codage progressif en qualité?	96
4.4.1	Schéma LAE modifié	96
4.4.1.1	Un niveau de décomposition	98
4.4.1.2	Plusieurs niveaux de décompositions	98
4.4.2	Description du codec	99
4.4.3	Estimation des coefficients de pondération	101
4.4.4	Évaluation des performances à hauts débits	103
4.4.4.1	La méthode GAEa	103
4.4.4.2	La méthode LAE modifiée	103
4.5	Conclusion	103
	Conclusion générale	105
	Bibliographie	107
	Annexes	114
	A Courbes débit/distorsion relatives à la section 4.4.4	115
	B Images utilisées dans les tests	125
B.1	Les mages naturelles	125
B.2	Images médicales IRM	128
B.3	Images satellitaires	131
B.4	Images de textures et empreintes digitales	133

C Transformations en ondelettes de \mathbb{Z} dans \mathbb{Z}	135
C.1 La transformation S	135
C.2 La transformation (2, 2) ou 5/3	135
C.3 La transformation (4, 2)	136
C.4 La transformation (2, 4)	136
C.5 La transformation (2+2, 2)	137
C.6 La transformation (4, 4)	137
C.7 La transformation (6, 2)	138
C.8 La transformation 9 – 7	138
C.9 Les transformations S+P et TS	140

Notations

Acronymes

1D : mono-dimensionnel.

2D : bi-dimensionnel.

AR : Auto-Régressif, filtre AR.

ARMA : Auto-Régressif à Moyenne Ajustée, filtre ARMA.

BPP : Bits Par Pixel, nombre moyen de bits par pixel.

CALIC : *Context-based, Adaptive, Lossless Image Coding*, codeur sans perte introduit par [124].

CREW : *Compression with Reversible Embedded Wavelets*, codeur d'image introduit par Zandi et al. [128].

DSP : Densité Spectrale de Puissance d'un signal stationnaire au sens large.

DCT : *Discret Cosinus Transform*.

DPCM : Differential Pulse Code Modulation (type de codage).

DWT : *Discret Wavelet Transform*.

DMT : *Discret MultiTone*, type de communication numérique (voir [18]).

EBCOT : Embedded Block Coding with Optimal Truncation, codeur entropique introduit dans [106].

EQM : Erreur Quadratique Moyenne.

EZW : *Embedded Zerotree Wavelet*, schéma de codage introduit par Shapiro [98].

GAE : *Globally Adapted Estimation*, une des deux méthodes proposées, ayant une adaptation globale.

GAEa : variante de la méthode GAE décrite au paragraphe 4.3 page 80.

GAP : *Gradient-Adjusted Predictor*, utilisé dans CALIC.

GSN : Gradient Stochastique Normalisé (algorithme).

IEC : *International Electrotechnical Commission*.

IRM : Imagerie par Résonance Magnétique.

ISO : *International Organization for Standardization*.

JASPER : implantation en langage C du standard JPEG 2000 pour la compression avec et sans perte (voir la note 6 au bas de la page 51).

JPEG : *Joint Photographic Expert Group*, la norme JPEG établie par ISO et IEC.

- JPEG 2000** : nouveau standard en codage d'image.
- JPEG-LS** : nouveau standard de compression sans perte de la norme JPEG, utilise l'algorithme LOCO-I.
- LAE** : *Locally Adapted Estimation*, une des deux méthodes proposées, ayant une adaptation locale.
- LAR** : *Locally Adaptive Resolution*, méthode de compression d'image introduite dans [42].
- LOCO-I** : codeur sans perte (utilisé dans la norme JPEG-LS) introduit dans [120].
- LOT** : *Lapped Orthogonal Transform*, introduites par Malvar [71].
- MA** : Moyenne Ajustée, filtre MA.
- MCA** : Moindres Carrés Adaptatifs (algorithme).
- MIMO** : *Multi-Input, Multi-Output*, système à plusieurs entrées et plusieurs sorties.
- MMSE** : *Minimum Mean Square Error*, désigne un banc de filtres.
- PC** : *Personal Computer*.
- PCP** : Pulse Coded Modulation, type de codage.
- PSNR** : *Peak Signal to Noise Ratio*, Pic du Rapport Signal à Bruit exprimé en anglais.
- RIF** : Réponse Impulsionnelle Finie (s'applique à un filtre).
- RII** : Réponse Impulsionnelle Infinie (s'applique à un filtre).
- RLE** : *Run Length Encoding*, type de codage.
- S** : La transformation S.
- S+P** : Transformation de \mathbb{Z} dans \mathbb{Z} introduite par Said et Pearlman [92].
- SFQ** : *Space-Frequency Quantization*, codeur entropique introduit dans [125].
- SGAE** : méthode GAE avec la transformation S.
- SHD** : Super Haute Résolution, s'applique à des images.
- SISO** : *Single Input, Single Output*, système à une entrée et une sortie.
- SPA** : transformation S+P avec le prédicteur A.
- SPB** : transformation S+P avec le prédicteur B.
- SPC** : transformation S+P avec le prédicteur C.
- SPIHT** : *Set Partitioning In Hierarchical Trees*, codeur entropique permettant du codage progressif en qualité introduit par Said et Pearlman [91].
- TFD** : Transformée de Fourier Discrète.
- WSS** : *Wide Sens Stationary*, stationnaire au sens large (pour un signal).
- TS** : Transformation TS.

Notations

- Les vecteurs (identifiés à la matrice colonne de leurs coordonnées) sont désignés par des lettres soulignées (ex. \underline{v}). Les matrices sont désignées par des lettres majuscules en caractère gras (ex. \mathbf{A}). Nous allons aussi rencontrer des fonctions matricielles $\mathbf{A}(z)$. Ce sont des matrices dont les éléments sont des fractions rationnelles en z ou z^{-1} .
- \mathbf{A}^T signifie la matrice transposée de \mathbf{A} .
- \mathbf{A}^\dagger signifie la matrice transposée et conjuguée de \mathbf{A} .
- $\delta(n)$ est le symbole de Kronecker, vérifiant $\delta(0) = 1$ et $\delta(n) = 0$ pour $n \neq 0$.
- L'exposant astérisque, ex. $H^*(z)$, indique la conjugaison complexe de $H(z)$, tandis que l'indice astérisque $H_*(z)$ indique la conjugaison des coefficients de $H(z)$ uniquement.
- Pour une fraction rationnelle $H(z)$, la notation $\tilde{H}(z)$ est définie comme suit : conjuguer les coefficients et remplacer z par z^{-1} . Autrement dit, $\tilde{H}(z) = H_*(z^{-1})$. Pour des matrices nous posons $\tilde{\mathbf{A}}(z) = \mathbf{A}_*^T(z^{-1})$.

Grandeurs utilisées

Variance

La variance σ_I^2 d'une image I de taille $N \times M$ est définie par

$$\sigma_I^2 = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M \left[I(i, j) - \left(\frac{1}{NM} \sum_{k=1}^N \sum_{l=1}^M I(k, l) \right) \right]^2.$$

Erreur quadratique moyenne

L'erreur quadratique moyenne (EQM) est utilisée pour mesurer la distorsion entre une image I de taille $N \times M$ et son approximation \hat{I} . Une petite valeur de l'EQM correspond à une petite distorsion. Elle est définie par la relation

$$\text{EQM} = D(I, \hat{I}) = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M \left[I(i, j) - \hat{I}(i, j) \right]^2.$$

Pic du rapport signal sur bruit

Le pic du rapport signal sur bruit ou PSNR (*Peak Signal to Noise Ratio*), est une autre mesure de distorsion exprimée en dB. Des petites distorsions correspondent à des grandes valeurs du PSNR. Il est défini par

$$\text{PSNR} = 10 \log_{10} \frac{P^2}{\text{EQM}}$$

où P est la valeur maximale d'un pixel. Par exemple, pour les images codées sur 8 bits, $P = 255$.

Entropie du premier ordre

L'entropie du premier ordre $H(I)$ d'une image I de taille $N \times M$ et prenant les valeurs discrètes s_i est donnée par

$$H(I) = - \sum_i P(s_i) \log_2 P(s_i),$$

où la probabilité $P(s_i)$ est la proportion des pixels de I ayant la valeur s_i :

$$P(s_i) = \frac{\text{card} \{(n, m) \mid I(n, m) = s_i\}}{NM}.$$

Entropie du premier ordre moyenne

Après une décomposition multi-résolution réversible, les sous-bandes de l'image transformée ont des statistiques différentes. L'entropie du premier ordre moyenne de l'image décomposée est calculée comme la somme des entropies du premier ordre de ces images de sous-bandes pondérées par leurs tailles relatives. Par exemple, l'image transformée de la figure 1, a son entropie du premier ordre moyenne, ou plus brièvement son entropie moyenne, qui vaut

$$H_{moy}(I) = \frac{1}{16} [H(I_{\ell\ell}) + H(I_{\ell h1}) + H(I_{h\ell1}) + H(I_{hh1})] + \frac{1}{4} [H(I_{\ell h}) + H(I_{h\ell}) + H(I_{hh})].$$

Plus généralement, pour L niveaux de décomposition, l'entropie moyenne vaut

$I_{\ell\ell}$	$I_{h\ell1}$	$I_{h\ell}$
$I_{\ell h1}$	I_{hh1}	
$I_{\ell h}$		I_{hh}

FIG. 1 – Deux niveaux de décomposition.

$$H_{moy}(I) = \frac{H(I_{\ell\ell})}{4^L} + \frac{1}{4} \sum_{i=0}^{L-1} \frac{H(I_{\ell hi}) + H(I_{h hi}) + H(I_{h li})}{4^i},$$

où l'indice i dans la sous-image $I_{h li}$ (par exemple) indique le $(i + 1)^{\text{ème}}$ niveau de décomposition.

Taille du flot de bits ou débit réel

L'entropie du premier ordre et l'entropie moyenne permettent d'évaluer les performances de différentes transformations multi-résolutions réversibles sans avoir besoin de coder l'image transformée et indépendamment du codeur entropique. Pour connaître le nombre exact de bits nécessaires au codage de l'image transformée, il

faut associer à la transformation un codeur qui est généralement entropique. Un système de compression complet (figure 2.1) est alors obtenu. Le débit réel est mesuré à partir du nombre moyen de bits par pixel (*bpp*) de l'image codée, il est défini par

$$R = \frac{\text{taille du flot de bits (en bits)}}{\text{nombre de pixels de l'image originale}},$$

et s'exprime en bits par pixel.

Dans le cas de la compression avec pertes, la distorsion varie avec le débit. Pour évaluer les performances de la transformation en codage progressif, des courbes donnant la distorsion en fonction du débit, c'est-à-dire $\text{PSNR} = f(R)$ sont tracées. Dans le cas de la compression sans perte, la distorsion est nulle.

Introduction générale

Les décompositions multi-résolutions par ondelettes et bancs de filtres ont montré leur efficacité en plusieurs domaines et en particulier en compressions d'images dès le début des années 90 (voir l'article d'Antonini et *al.* [14]). Une nouvelle famille de codeurs d'images à base d'ondelettes a émergé après la publication des travaux de Shapiro en 1993 [98], où les codeurs par arbres de zéros, qui exploitent la représentation bi-dimensionnelle de la transformée en ondelettes d'une image, ont été introduits. Les codecs résultants, améliorés tout au long des années 90 ont montré leur supériorité par rapport au standard JPEG. De plus, ils offraient une grande souplesse aux utilisateurs, comme le codage progressif en résolution ou en qualité et aujourd'hui le nouveau standard JPEG 2000 utilise des décompositions en ondelettes, aussi bien en codage sans pertes qu'avec pertes.

Il faut noter que l'une des propriétés sur lesquelles les codeurs d'images à base d'ondelettes sont basés n'est pas satisfaite. En effet, les images réelles sont fondamentalement non stationnaires et non régulières. Le plus basique des modèles d'images doit inclure des régions, séparées par des contours. En général les régions sont constituées de textures irrégulières. En deux mots, les images naturelles (comme les signaux de parole) n'ont pas cette régularité qui permet aux ondelettes d'être des bases de représentation optimales. À partir de là, plusieurs chercheurs ont proposé des schémas qui modifient la base d'ondelettes appliquée en fonction des caractéristiques locales du signal. Plus généralement, des schémas de compression associés à des bancs de filtres à reconstruction parfaite qui s'adaptent au signal à coder ont été étudiés. Citons par exemple le *lifting scheme* adaptatif de Gerek et Çetin [46] qui peut être considéré comme du codage par prédiction linéaire multi-résolution appliqué aux images.

Le codage par prédiction linéaire a montré son efficacité sur des signaux mono-dimensionnels, par exemple en codage de signaux de parole des téléphones portables, où il est appliqué à l'échelle industrielle. Appliqué aux images, il a été intégré partiellement dans différents schémas, voir par exemple l'article de Egger et *al.* [44] et sa bibliographie, puis (comme nous venons de le dire) plus récemment dans des schémas de compression multi-résolutions, mais il n'a jamais été appliqué à l'échelle industrielle (les normes JPEG et JPEG 2000 ne l'utilisent pas). Or, en étudiant les schémas de compression multi-résolutions qui utilisent le codage par prédiction linéaire, il apparaît que toute l'information disponible au décodeur pour prédire la valeur du pixel courant n'est pas exploitée et, par conséquent, que les performances limites des codeurs par prédiction linéaire multi-résolutions ne sont pas connues et qu'il est théoriquement possible d'améliorer celles obtenues aujourd'hui par de tels systèmes.

C'est l'objet de cette étude qui porte d'une part sur la synthèse de décompositions multi-résolutions permettant une reconstruction parfaite basées sur l'estimation linéaire en moyenne quadratique et d'autre part sur l'application de ces structures en codage d'images sans perte. Toute l'information disponible au niveau du décodeur pour prédire la valeur du pixel courant est exploitée. Nous nous sommes focalisés sur la compression sans perte d'images 2-D avec la possibilité d'un codage progressif en résolution pour trois raisons principales. La première est qu'à l'époque du début de ces travaux, il existait différents utilisateurs potentiels de ce type de codage (en imagerie satellitaire ou médicale, par exemple). Aujourd'hui, même dans ces domaines les utilisateurs s'orientent vers du codage avec pertes. La seconde est que le nouveau standard de compression d'images JPEG 2000, Part I, venait de sortir, figeant les techniques de codage avec pertes pour plusieurs années, et enfin comme nous le verrons dans ce rapport, l'ajout d'arrondis à l'entier le plus proche pour rendre réversibles les décompositions que nous avons étudiées (et donc pour les appliquer en codage sans perte) simplifie significativement le problème de l'instabilité éventuelle du filtre de reconstruction.

Ce rapport se divise en deux parties principales plus les annexes. La première est constituée des chapitres 1 et 2 et donne un aperçu de l'état de l'art d'une part sur les décompositions par bancs de filtres à reconstruction parfaite optimales (sous certaines conditions) en codage. Les différents critères qui interviennent dans l'adaptation sont détaillés ainsi que les hypothèses nécessaires pour que cette dernière soit optimale en codage. C'est l'objet du chapitre 1. D'autre part, le chapitre 2 rappelle les notions de base en compression d'images, il trace un bref historique de l'évolution des systèmes de compression et présente les principaux systèmes de compression sans perte.

La seconde partie, constituée des chapitres 3 et 4, présente notre contribution à l'étude du codage par prédiction linéaire multi-résolution. Le chapitre 3 commence par une justification du critère que nous avons retenu pour adapter les filtres, puis introduit la nouvelle structure en *lifting scheme* généralisée, qui permet d'exploiter toute l'information disponible au décodeur pour estimer la valeur du pixel courant, associée à deux techniques d'adaptation. La première, appelée GAE (*Globally Adapted Estimation*), est basée sur les moindres carrés et suppose une stationnarité globale de l'image, tandis que la seconde, dite LAE (*Locally Adapted Estimation*), utilise les moindres carrés adaptatifs et ne suppose qu'une stationnarité locale des images. Le chapitre 3 termine par la comparaison des performances en compression sans perte de cette nouvelle structure sur des images de textures synthétiques. Le début du chapitre 4 est consacré à l'application de la structure introduite en codage sans perte d'images réelles, avec la possibilité d'un codage progressif en résolution, quatre familles d'images sont testées (naturelles, satellitaires, IRM médicales, et des textures et empreintes), les performances obtenus sont comparées à l'état de l'art. La fin du chapitre est consacrée au codage avec pertes.

Les résultats présentés dans ce rapport ont fait l'objet de cinq publications, dont quatre dans des conférences internationales avec comité de lecture :

1. H. Bekkouche et M. Barret, "Décomposition multi-résolution adaptative. Application à la compression sans perte des images", *Actes de la Réunion des Théoriciens des Circuits de Langue Française*, Paris, 2001.

2. Barret M. et Bekkouche H., “Adapted nonlinear multiresolution decomposition with applications in progressive lossless image coding”, *Proc. of Int. Symposium on Signal and Image Processing and Analysis*, pp. 609–612, Pula (Croatie) juin 2001.
3. Bekkouche H. et Barret M., “Adaptive multiresolution decomposition : application to lossless image compression”, *Proc. of IEEE ICASSP-02*, pp. 3533–3536, Orlando (USA), mai 2002.
4. Bekkouche H. et Barret M., “Comparison of lossless codecs for satellite and MRI images”, *Proc. of EUSIPCO-02*, pp. 475–478, Toulouse (France), sept. 2002.
5. M. Barret, H. Bekkouche, J.-L. Collette et J. Oksman, “Adapted lifting schemes for lossless image coding”, *Workshop Transmitting, processing and watermarking multimedia contents*, Bordeaux, pp. 49–54, avril 2003.

CHAPITRE 1

Banc de filtres, ondelettes et *lifting* *scheme* adaptés

1.1 Introduction

Aujourd'hui, les ondelettes et les bancs de filtres ont prouvé leur utilité dans de nombreux domaines d'application du traitement du signal et des images (analyse, débruitage, restauration, etc...) et notamment en codage d'images [68]. Dès la fin des années 1980, début des années 1990, plusieurs codeurs basés sur des décompositions en ondelettes ont été introduits. Citons les premiers travaux d'Antonini *et al.* [14] et ceux de Mallat [69].

Il existe une grande variété d'ondelettes bi-orthogonales et très tôt s'est posée la question du choix de l'ondelette la mieux adaptée à un signal pour une application donnée. Or, en codage, il n'existe pas une ondelette qui soit optimale quel que soit le signal à traiter. De plus, pour un signal donné, ses propriétés moyennes varient dans le temps (ou dans l'espace), la décomposition en ondelettes optimale peut varier localement [41]. C'est pourquoi, des schémas de codage utilisant une sélection adaptative de l'ondelette en fonction des statistiques du signal à coder, ou des schémas à base de paquets d'ondelettes introduits par Coifman *et al.* [75] et dont l'ondelette varie suivant le niveau de décomposition, ont été proposés et étudiés [85, 56, 86].

Par ailleurs, la recherche de nouvelles ondelettes et l'étude de leurs performances dans diverses applications ont fait l'objet de recherches intensives pendant les années 90. En 1995, Sweldens a proposé le *lifting scheme* [103]. C'est une technique de construction d'ondelettes ayant des moments nuls pour des ordres de plus en plus élevés¹. Le *lifting scheme* permet une implantation très simple des décompositions en ondelettes et de leurs opérations inverses. Sweldens a publié, avec parfois d'autres auteurs, plusieurs articles sur le sujet (par exemple [39] [28] [104]). En 1996, Calderbank *et al.* [28] ont proposé une technique à base de *lifting scheme* pour rendre réversibles, c'est-à-dire d'entiers en entiers, les décompositions en ondelettes dya-

¹L'origine de l'expression *lifting scheme* est associée à un ascenseur permettant d'accéder à des moments nuls d'ordres de plus en plus élevés.

diques associées à des filtres RIF. Cette technique est utilisée aujourd'hui dans la norme JPEG 2000 pour la compression sans perte des images. De la même façon que précédemment, dans le cadre du codage sans perte, il n'existe pas de décomposition en ondelettes réversible qui soit optimale quel que soit le signal à coder [9]. Et si les performances des décompositions sont médiocres, l'emploi de codeurs entropiques sophistiqués ne permet pas d'atteindre des gains de codages significatifs [90].

Nous nous intéressons aux bancs de filtres à reconstruction parfaite qui s'adaptent au signal (en particulier à l'image) dans un système de compression sans perte et dans ce cadre un certain nombre de travaux récents proposent des décompositions multi-résolutions réversibles qui s'adaptent aux images. Par exemple dans [46], Gerek et Çetin utilisent un *lifting scheme* avec des filtres de prédiction adaptatifs. Les coefficients des filtres sont mis à jour pour chaque pixel de l'image par l'algorithme du gradient stochastique, dans le but de minimiser la variance de l'erreur de prédiction. Dans [27], Boulgouris *et al.* expriment le prédicteur optimal d'une structure en *lifting scheme* en fonction de la densité spectrale de puissance du signal multi-dimensionnel d'entrée. Dans [34], Claypoole *et al.* utilisent une adaptation de l'ordre du filtre prédicteur d'une structure en *lifting scheme* suivant la position du pixel courant, distinguant s'il est dans une zone de texture ou une zone de contour. A. Gouze *et al.* [51] proposent, dans un cadre de compression avec pertes, un *lifting scheme* optimisé sous contrainte.

Dans ce chapitre, nous commençons par présenter des éléments bien connus de la théorie des bancs de filtres à reconstruction parfaite. Puis, nous rappelons sous quelles hypothèses, en codage avec pertes, il est possible aujourd'hui de poser le problème de la recherche du banc de filtres optimal pour un signal stationnaire au sens large.

1.2 Banc de filtres à reconstruction parfaite

1.2.1 Sous- et sur-échantillonnage

L'opération de *sous-échantillonnage* d'un facteur M appliquée à un signal $x(n)$ consiste à en prélever un échantillon sur M . Elle est représentée par le schéma bloc de la figure 1.1. Le signal $y(n)$ est défini par l'égalité $y(n) = x(Mn)$. Par transformée



FIG. 1.1 – Opération de sous-échantillonnage.

de Fourier il vient

$$Y(e^{j\omega}) = \frac{1}{M} \sum_{k=0}^{M-1} X(e^{j(\omega-2\pi k)/M}) \quad (1.1)$$

Le sous-échantillonnage a une interprétation simple dans le domaine fréquentiel : le spectre du signal sous-échantillonné est la superposition de M versions décalées et étirées du spectre du signal d'entrée. D'après l'équation (1.1), l'opération de sous-échantillonnage cause un repliement de spectre, par conséquent il est impossible de

restituer le signal original à partir de sa version sous-échantillonnée, sauf dans le cas particulier où le signal $x(n)$ est à bande limitée $|\omega| < 2\pi/M$ [113]. Ce repliement de spectre est illustré par l'exemple de la figure 1.2 avec $M = 2$.

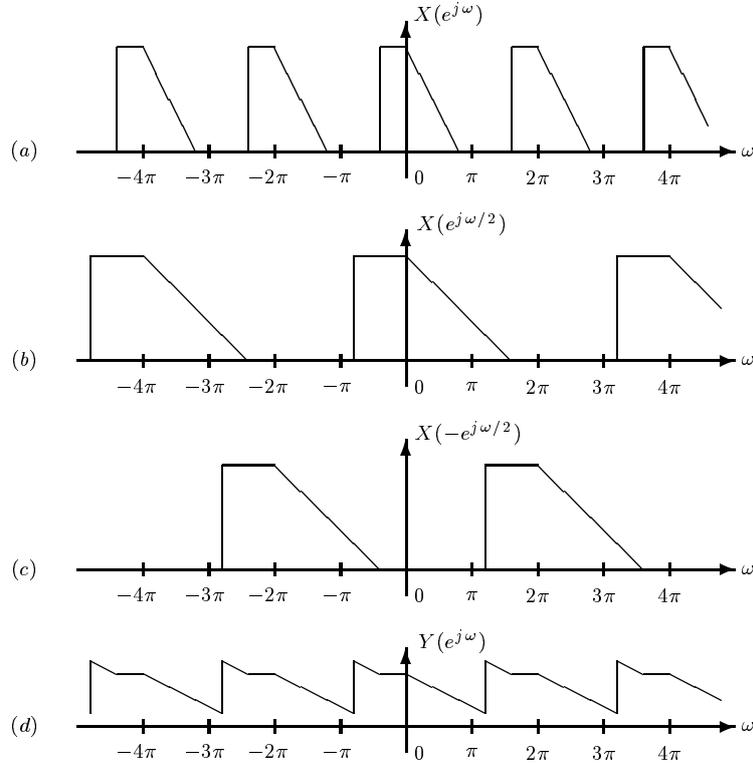


FIG. 1.2 – Effet du sous-échantillonnage avec un facteur $M = 2$ dans le domaine fréquentiel. (a) Spectre du signal d'entrée. (b) et (c) Deux versions étirées et décalées du spectre du signal original. (d) Spectre du signal sous-échantillonné.

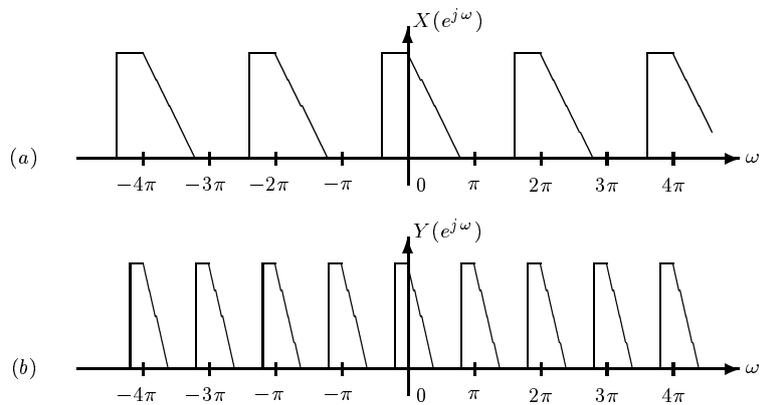


FIG. 1.3 – Effet du sur-échantillonnage avec un facteur $M = 2$ dans le domaine fréquentiel. (a) Spectre du signal d'entrée. (b) Spectre du signal sur-échantillonné.

Le signal $x(n)$ est *sur-échantillonné* d'un facteur M quand, entre deux échan-

tillons consécutifs, sont insérés $M - 1$ zéros. Le signal obtenu $y(n)$ est donc défini par la relation

$$y(n) = \begin{cases} x(n/M) & \text{si } n \text{ est multiple de } M \\ 0 & \text{sinon} \end{cases}$$

et l'opération qui transforme $x(n)$ en $y(n)$ est dite de *sur-échantillonnage*. Elle est représentée par le schéma bloc de la figure 1.4. La relation entre les transformées de



FIG. 1.4 – Opération de sur-échantillonnage

Fourier de l'entrée et de la sortie d'un sur-échantillonneur est donnée par

$$Y(e^{j\omega}) = X(e^{jM\omega}). \quad (1.2)$$

Puisque le spectre n'est pas altéré par le sur-échantillonnage : en effet, il ne subit aucun repliement, le signal initial peut être restitué à partir de sa version sur-échantillonnée. La figure 1.3 illustre les effets du sur-échantillonnage dans le domaine fréquentiel quand $M = 2$.

Souvent un sous-échantillonneur ou un sur-échantillonneur est mis en série avec un filtre. Ces associations apparaissent par exemple dans les bancs de filtres (cf. figure 1.6). Bien qu'en général, l'ordre des opérations de sous-échantillonnage (sur-échantillonnage) et de filtrage ne puisse pas être inversé sans changer le comportement du système, il est parfois possible de trouver deux systèmes équivalents avec l'ordre de ces opérations inversé. Formellement, cela n'est possible que sous certaines conditions, qui sont représentées par les schémas blocs de la figure 1.5 et que l'on appelle les *identités nobles*. En plus de leur utilité théorique, ces identités ont une utilité pratique non négligeable : il est généralement préférable d'effectuer les opérations de filtrage du côté du sous-échantillonneur (sur-échantillonneur) correspondant à la cadence la plus petite.

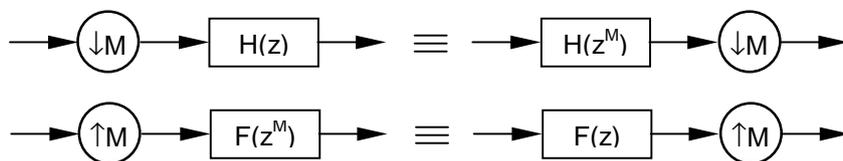


FIG. 1.5 – Les identités nobles.

1.2.2 Analyse et synthèse par bancs de filtres

Un banc de filtres est un ensemble de filtres, avec une entrée ou une sortie commune. Ces deux cas sont représentés sur la figure 1.6. Le système de la figure 1.6(a) est appelé *banc de filtres d'analyse*, et les filtres $H_k(z)$ sont les *filtres d'analyse*. Ce

banc décompose le signal $x(n)$ en M signaux $v_i(n)$ appelés *signaux de sous-bandes*. Le système de la figure 1.6(b) est appelé *banc de filtres de synthèse* et les filtres $F_k(z)$ sont les *filtres de synthèse*. Il combine les M signaux $w_k(n)$ en un seul signal $y(n)$. Les bancs de filtres d'analyse et de synthèse sont généralement associés, le premier décompose un signal pour appliquer un traitement à chaque signal de sous-bande, et le second re-combine les signaux de sous-bandes traités pour construire le signal modifié. Un tel système d'analyse/synthèse est appelé banc de filtres à reconstruc-

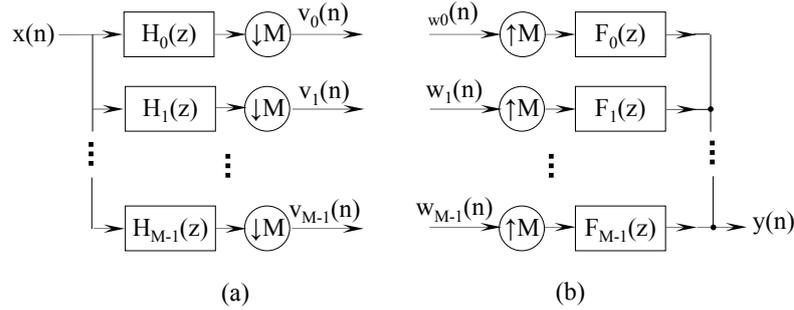


FIG. 1.6 – Banc de filtres à M canaux (a) banc d'analyse (b) banc de synthèse.

tion parfaite quand en l'absence de tout traitement dans les sous-bandes, c'est-à-dire quand $w_k(n) = v_k(n)$, $k = 0, \dots, M - 1$, le signal de sortie $y(n)$ vaut $x(n - \tau)$, où τ est un retard pur.

Pour obtenir des conditions nécessaires et suffisantes assurant la reconstruction parfaite, commençons par donner l'expression du $k^{\text{ème}}$ signal de sous-bande :

$$v_k(n) = \sum_{i=-\infty}^{\infty} x(i)h_k(nM - i) \quad (1.3)$$

où $h_k(n)$ est la réponse impulsionnelle du $k^{\text{ème}}$ filtre d'analyse. Le signal reconstruit s'exprime en fonction des signaux de sous-bandes traités comme suit :

$$y(n) = \sum_{k=0}^{M-1} \sum_{i=-\infty}^{\infty} w_k(i)f_k(n - Mi) \quad (1.4)$$

où $f_k(n)$ est la réponse impulsionnelle du $k^{\text{ème}}$ filtre de synthèse.

Lorsque les signaux de sous-bandes ne sont pas modifiés, $w_k(n) = v_k(n)$ quel que soit k , (1.3) peut être injectée dans (1.4), pour obtenir

$$y(n) = \sum_{l=-\infty}^{\infty} x(l)S(n, l), \quad (1.5)$$

où le noyau S du système global est donné par

$$S(n, l) = \sum_{k=0}^{M-1} \sum_{i=-\infty}^{\infty} f_k(n - Mi)h_k(Mi - l). \quad (1.6)$$

Nous pouvons obtenir la reconstruction parfaite avec un retard τ (entier) si et seulement si $S(n, l) = \delta(n - l - \tau)$, c'est-à-dire si et seulement si

$$\sum_{k=0}^{M-1} \sum_{i=-\infty}^{\infty} f_k(n - Mi)h_k(Mi - l) = \delta(n - l - \tau), \quad (\forall l, n \in \mathbb{Z}). \quad (1.7)$$

Pour la suite, nous supposons sans perte de généralité que $\tau = 0$.

1.2.3 Représentation polyphase d'un banc de filtres

L'invention de la représentation polyphase d'un filtre par Bellanger *et al.* [24] a permis la simplification de plusieurs résultats théoriques en traitement du signal et a conduit à des implantations plus simples des bancs de filtres. Pour expliquer l'idée de base, considérons un filtre de fonction de transfert² $H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$. La séparation des coefficients d'indices pairs de ceux d'indices impairs, donne

$$H(z) = \sum_{n=-\infty}^{\infty} h(2n)z^{-2n} + z^{-1} \sum_{n=-\infty}^{\infty} h(2n+1)z^{-2n}. \quad (1.8)$$

En définissant³

$$E_0(z) = \sum_{n=-\infty}^{\infty} h(2n)z^{-n} \quad \text{et} \quad E_1(z) = \sum_{n=-\infty}^{\infty} h(2n+1)z^{-n} \quad (1.9)$$

nous pouvons écrire

$$H(z) = E_0(z^2) + z^{-1}E_1(z^2). \quad (1.10)$$

D'une manière plus générale, et pour un entier M donné, $H(z)$ peut être décomposé en

$$H(z) = \sum_{\ell=0}^{M-1} z^{-\ell} E_{\ell}(z^M) \quad (1.11)$$

avec

$$E_{\ell}(z) = \sum_{n=-\infty}^{\infty} e_{\ell}(n)z^{-n} \quad (1.12)$$

et

$$e_{\ell}(n) \triangleq h(Mn + \ell) \quad , \quad 0 \leq \ell \leq M - 1. \quad (1.13)$$

Le membre de droite de l'égalité (1.11) est appelé la *représentation polyphase de type 1 ou 2* [113] du filtre $H(z)$ et les filtres $E_{\ell}(z)$ sont ses composantes polyphases.

Regardons maintenant comment utiliser les représentations polyphases pour décrire un banc de filtres [117]. En utilisant l'équation (1.11), la représentation des filtres d'analyse $H_k(n)$ de la figure 1.6(a) en fonction de leurs composantes polyphases donne

$$H_k(z) = \sum_{\ell=0}^{M-1} z^{-\ell} E_{k\ell}(z^M) \quad (1.14)$$

²Nous supposons donc que la série de Laurent

$$\sum_{n=-\infty}^{+\infty} h(n)z^{-n}$$

converge dans la couronne ouverte non vide $\mathcal{C} = \{z \in \mathbb{C} \mid 0 < r < |z| < R\}$ ou dans un disque ouvert.

³Remarquons que les séries de Laurent des relations (1.9) convergent sur la couronne (ou le disque) non vide $\mathcal{C}' = \{z \in \mathbb{C} \mid \sqrt{r} < |z| < \sqrt{R}\}$.

ou sous forme matricielle

$$\begin{bmatrix} H_0(z) \\ \vdots \\ H_{M-1}(z) \end{bmatrix} = \begin{bmatrix} E_{00}(z^M) & E_{01}(z^M) & \cdots & E_{0,M-1}(z^M) \\ \vdots & \vdots & \ddots & \vdots \\ E_{M-1,0}(z^M) & E_{M-1,1}(z^M) & \cdots & E_{M-1,M-1}(z^M) \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(M-1)} \end{bmatrix}$$

et d'une manière plus compacte

$$\underline{\mathbf{h}}(z) = \mathbf{E}(z^M)\underline{\mathbf{e}}(z) \quad (1.15)$$

avec

$$\mathbf{E}(z) = \begin{bmatrix} E_{00}(z) & E_{01}(z) & \cdots & E_{0,M-1}(z) \\ \vdots & \vdots & \ddots & \vdots \\ E_{M-1,0}(z) & E_{M-1,1}(z) & \cdots & E_{M-1,M-1}(z) \end{bmatrix}, \quad (1.16)$$

$\underline{\mathbf{h}}(z) = [H_0(z) \ \cdots \ H_{M-1}(z)]^T$ et $\underline{\mathbf{e}}(z) = [1 \ z^{-1} \ \cdots \ z^{-(M-1)}]^T$. La matrice $\mathbf{E}(z)$ est la *matrice polyphase* de type 1 du banc d'analyse.

D'une manière identique, en utilisant les représentations polyphases⁴ de type 3 [8],

$$F_k(z) = \sum_{\ell=0}^{M-1} z^\ell R_{\ell k}(z^M) \quad (1.17)$$

le banc de filtres de synthèse admet une représentation matricielle. Nous avons

$$\begin{bmatrix} F_0(z) & \cdots & F_{M-1}(z) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & z^{M-1} \end{bmatrix} \begin{bmatrix} R_{00}(z^M) & \cdots & R_{0,M-1}(z^M) \\ R_{10}(z^M) & \cdots & R_{1,M-1}(z^M) \\ \vdots & \ddots & \vdots \\ R_{M-1,0}(z^M) & \cdots & R_{M-1,M-1}(z^M) \end{bmatrix}$$

ou encore

$$\underline{\mathbf{f}}^T(z) = \underline{\tilde{\mathbf{e}}}(z)\mathbf{R}(z^M) \quad (1.18)$$

avec

$$\mathbf{R}(z) = \begin{bmatrix} R_{00}(z) & \cdots & R_{0,M-1}(z) \\ R_{10}(z) & \cdots & R_{1,M-1}(z) \\ \vdots & \ddots & \vdots \\ R_{M-1,0}(z) & \cdots & R_{M-1,M-1}(z) \end{bmatrix} \quad (1.19)$$

et $\underline{\mathbf{f}}(z) = [F_0(z), \dots, F_{M-1}(z)]^T$. La matrice $\mathbf{R}(z)$ est appelée la *matrice polyphase du banc de synthèse*. Les deux représentations polyphases décrites ci-dessus donnent le schéma bloc de la figure 1.7 (a). Grâce aux nobles identités nous obtenons la représentation polyphase (voir la figure 1.7 (b)) du banc de filtres à M sous-bandes de la figure 1.6.

Les bancs de filtres à reconstruction parfaite sont également appelés bancs de filtres bi-orthogonaux. Il sont étroitement liés à la théorie des ondelettes comme nous allons le voir au paragraphe suivant.

⁴Pour un filtre $F(z) = \sum_n f(n)z^{-n}$, sa représentation polyphase de type 3 ou 4 s'écrit $F(z) = \sum_{\ell=0}^{M-1} z^\ell R_\ell(z^M)$ avec $R_\ell(z) = \sum_u f(uM - \ell)z^{-u}$.

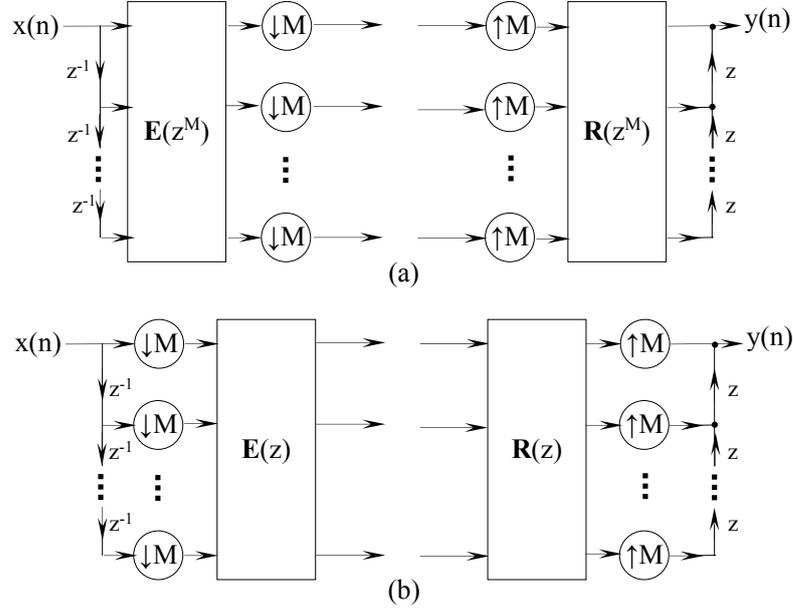


FIG. 1.7 – (a) Représentation polyphase d'un banc de filtres à M sous-bandes. (b) Réarrangement en utilisant les nobles identités.

1.3 Ondelettes et *lifting scheme*

Les ondelettes ont été introduites par Grossman et Morlet [70] et se définissent comme une famille de fonction $\psi_{a,b}$ de $L^2(\mathbb{R})$. Elle sont générées par dilatations et translations d'une fonction de base d'énergie, finie ($\in \mathbf{L}^2(\mathbb{R})$) : l'ondelette mère ψ .

$$\psi_{a,b}(x) = |a|^{-\frac{1}{2}} \psi\left(\frac{x-b}{a}\right) \quad \forall (a,b) \in \mathbb{R}^2, a \neq 0.$$

Les coefficients a et b désignent respectivement le facteur d'échelle pour la dilatation de ψ et le coefficient de translation. Le coefficient d'ondelette d'une fonction $f \in L^2(\mathbb{R})$ au temps b et à l'échelle a est défini par le produit scalaire entre f et l'ondelette $\psi_{a,b}$:

$$Wf(a,b) = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{+\infty} f(x) |a|^{-\frac{1}{2}} \psi^*\left(\frac{x-b}{a}\right) dx \quad (1.20)$$

Si l'ondelette mère ψ vérifie la condition d'admissibilité suivante

$$C_\psi = \int_{-\infty}^{+\infty} \frac{|\hat{\psi}(\omega)|^2}{\omega} d\omega < +\infty \quad (1.21)$$

alors la transformée inverse est donnée par

$$f(x) = \frac{1}{C_\psi} \int_a \int_b Wf(a,b) \psi_{a,b}(x) \frac{dadb}{a^2} \quad (1.22)$$

La fonction f peut aussi être représentée par une famille dénombrable de coefficients d'ondelette. Le facteur d'échelle a et le coefficient de translation b prennent alors un ensemble discret de valeurs. La discrétisation de a et b est réalisée en choisissant

$a_0 > 1$ et $b_0 > 0$ et en posant $a = a_0^m$ et $b = nb_0 a_0^m$ avec $m, n \in \mathbb{Z}$. Les bases d'ondelettes sont alors définies par les fonctions [70]

$$\psi_{m,n}(x) = a_0^{-\frac{m}{2}} \psi(a_0^{-m}x - nb_0)$$

Par suite, la décomposition en ondelettes et la transformée inverse sont déterminées par les équations suivantes :

$$Wf(m, n) = \langle f, \psi_{m,n} \rangle = \int_{\mathbb{R}} f(x) \psi_{m,n}^*(x) dx \quad (1.23)$$

$$f = \sum_{(m,n) \in \mathbb{Z}^2} Wf(m, n) \psi_{m,n} \quad \text{convergence dans } L^2(\mathbb{R}). \quad (1.24)$$

Ainsi, la transformée en ondelette associée à la fonction $f \in L^2(\mathbb{R})$ un ensemble discret de coefficients $Wf(m, n)$. Yves Meyer [74] a construit une *transformée en ondelettes dyadique* pour laquelle $a_0 = 2$ et $b_0 = 1$ et a montré que pour des fonctions particulières ψ de $L^2(\mathbb{R})$, la famille

$$\psi_{m,n}(x) = 2^{-\frac{m}{2}} \psi(2^{-m}x - n), \quad \forall (m, n) \in \mathbb{Z}^2$$

constitue une base orthonormée de $L^2(\mathbb{R})$.

1.3.1 Analyse multi-résolution et ondelettes

Le concept d'analyse multi-résolution a été introduit par S. Mallat [68, 69, 70]. Il s'agit de construire l'approximation d'un signal f à diverses résolutions. Ces approximations sont obtenues au moyen de projections orthogonales sur une famille de sous-espaces emboîtés $\{V_j\}_{j \in \mathbb{Z}}$ vérifiant quelques conditions.

DÉFINITION 1 (Mallat). *Une suite $\{V_j\}_{j \in \mathbb{Z}}$ de sous-espaces fermés de $L^2(\mathbb{R})$ est une approximation multi-résolution quand elle vérifie les cinq propriétés suivantes :*

$$\forall j \in \mathbb{Z}, \quad V_{j+1} \subset V_j, \quad (1.25)$$

$$\forall j \in \mathbb{Z}, \quad f(t) \in V_j \Leftrightarrow f\left(\frac{t}{2}\right) \in V_{j+1}, \quad (1.26)$$

$$\lim_{j \rightarrow +\infty} V_j = \bigcap_{j=-\infty}^{+\infty} V_j = \{0\}, \quad (1.27)$$

$$\lim_{j \rightarrow -\infty} V_j = \overline{\bigcup_{j=-\infty}^{+\infty} V_j} = L^2(\mathbb{R}), \quad (1.28)$$

$$\exists \theta \in V_0 \text{ tel que } \{\theta(t-k)\}_{k \in \mathbb{Z}} \text{ soit une base de Riesz de } V_0 \quad (1.29)$$

Rappelons qu'une famille $(e_j)_j$ de vecteurs est une base de Riesz d'un espace de Hilbert \mathcal{H} quand elle vérifie les deux conditions

- les combinaisons linéaires finies $\sum \beta_i e_j$ sont denses dans \mathcal{H}
- il existe deux constantes strictement positives C_1, C_2 telles que pour toute suite finie :

$$C_1 \left(\sum_j |B_j|^2 \right) \leq \left\| \sum_j B_j e_j \right\|^2 \leq C_2 \left(\sum_j |B_j|^2 \right).$$

L'approximation de f à la résolution 2^{-j} est définie par la projection orthogonale $P_{V_j}f$ de f sur V_j . Pour calculer cette projection, une base orthonormée $(\phi_{j,n})_{n \in \mathbb{Z}}$ de V_j est construite par translations et dilatations d'une seule fonction (dédiuite de la fonction θ), la fonction échelle ϕ :

$$\phi_{j,n}(x) = 2^{-j/2} \phi(2^{-j}x - n), \quad \forall (j, n) \in \mathbb{Z}^2. \quad (1.30)$$

La projection $P_{V_j}f$ est alors donnée par :

$$P_{V_j}f = \sum_{n \in \mathbb{Z}} \langle f, \phi_{j,n} \rangle \phi_{j,n}$$

Il résulte des propriétés (1.25) et (1.26) que $2^{-1/2}\phi(t/2) \in V_0$ et comme $\{\phi(t-n)\}_{n \in \mathbb{Z}}$ est une base orthonormée de V_0 , il existe des scalaires h_n tels que

$$\frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{+\infty} h_n \phi(t-n), \quad (1.31)$$

avec

$$h_n = \left\langle \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right), \phi(t-n) \right\rangle \quad (1.32)$$

La suite $(h_n)_n$ peut être considérée comme la réponse impulsionnelle d'un filtre numérique. La relation (1.31) associée au fait que $(\phi(t-k))_{k \in \mathbb{Z}}$ est une base de V_0 entraîne des conditions sur ce filtre : sa réponse fréquentielle $H(e^{j\omega})$ doit vérifier

$$|H(e^{j\omega})|^2 + |H(e^{j(\omega+\pi)})|^2 = 2 \quad (1.33)$$

Mallat donne une interprétation en termes d'analyse multi-résolution à la théorie des ondelettes, il considère ces dernières comme des fonctions contenant les détails nécessaires pour passer d'une résolution grossière à une résolution plus fine. Rappelons que les approximations de f aux résolutions 2^{-j} et 2^{-j+1} sont données par les projections orthogonales respectives de f sur V_j et sur V_{j-1} . Avec la propriété d'inclusion entre les sous-espaces d'approximation : $V_j \subset V_{j-1}$, le sous-espace supplémentaire orthogonal de V_j dans V_{j-1} est noté W_j :

$$V_{j-1} = V_j \oplus^\perp W_j.$$

Les ondelettes $(\psi_{j,n})_{n \in \mathbb{Z}}$ appartiennent au sous-espace W_j , elles constituent une base orthonormée de ce dernier et sont obtenues par dilatation de la fonction mère ψ :

$$\psi_{j,n}(x) = 2^{-j/2} \psi(2^{-j}x - n), \quad \forall (j, n) \in \mathbb{Z}^2. \quad (1.34)$$

La projection orthogonale de f sur V_{j-1} peut s'écrire comme la somme des projections orthogonales de f sur V_j et W_j . L'ondelette mère ψ associée à cette analyse multi-résolution est déterminée par la suite $(g_n)_n \in \ell^2(\mathbb{Z})$ de ses coordonnées :

$$\frac{1}{\sqrt{2}} \psi\left(\frac{x}{2}\right) = \sum_{n \in \mathbb{Z}} g_n \phi(x-n). \quad (1.35)$$

La relation

$$g_n = (-1)^n h_{-n+1}^* \quad (1.36)$$

permet de construire une ondelette mère [70]. Une transformée en ondelettes rapide décompose successivement chaque approximation $P_{V_j}f$ en une autre plus grossière $P_{V_{j+1}}f$ et en détails correspondant à $P_{W_{j+1}}f$. À la reconstruction, chaque projection $P_{V_j}f$ est calculée à partir de $P_{V_{j+1}}f$ et $P_{W_{j+1}}f$.

Comme $(\phi_{j,n})_{n \in \mathbb{Z}}$ et $(\psi_{j,n})_{n \in \mathbb{Z}}$ sont des bases orthonormées respectives de V_j et W_j , la projection de f sur ces sous-espaces est caractérisée par les coefficients

$$a_j(n) = \langle f, \phi_{j,n} \rangle \text{ et } d_j(n) = \langle f, \psi_{j,n} \rangle.$$

Le théorème de Mallat [70]-(Th. 7.7, p. 251) donne les relations de décomposition

$$a_{j+1}(p) = \sum_{n=-\infty}^{+\infty} \bar{h}(2p-n)a_j(n), \quad (1.37)$$

$$d_{j+1}(p) = \sum_{n=-\infty}^{+\infty} \bar{g}(2p-n)a_j(n) \quad (1.38)$$

avec, pour $x(n)$ donné, $\bar{x}(n) = x^*(-n)$ et celle de reconstruction :

$$\begin{aligned} a_j(p) &= \sum_{n=-\infty}^{+\infty} h(p-2n)a_{j+1}(n) + \sum_{n=-\infty}^{+\infty} g(p-2n)d_{j+1}(n) \\ &= \check{a}_{j+1} \star h(n) + \check{d}_{j+1} \star g(n) \end{aligned} \quad (1.39)$$

avec

$$\check{x}(n) = \begin{cases} x(n/2) & \text{si } n \text{ est pair} \\ 0 & \text{sinon.} \end{cases}$$

Les coefficients a_{j+1} et d_{j+1} se calculent respectivement en prenant un échantillon sur deux de la convolution de a_j avec \bar{h} et \bar{g} , comme indiqué sur la figure 1.8 (a). Le filtre \bar{h} atténue les hautes fréquences du signal $a_j(n)$, $n \in \mathbb{Z}$, alors que \bar{g} est un filtre passe-haut. La reconstruction (1.39) est une interpolation obtenue en insérant des zéros dans les suites a_{j+1} et d_{j+1} pour doubler leur longueur puis en les filtrant comme indiqué sur la figure 1.8 (b). Nous retrouvons les schémas d'un banc de filtres d'analyse et d'un banc de filtres de synthèse.

Nous venons de voir que la théorie des ondelettes et l'analyse multi-résolution, décrites ci-dessus pour des bases orthogonales d'ondelettes, donnent des bancs de filtres à reconstruction parfaite pour lesquels les filtres d'analyse satisfont les relations (1.36) et (1.33). De tels bancs de filtres admettent alors une matrice polyphase, dont la réponse fréquentielle est une matrice orthogonale ou, de façon équivalente, dont la fonction de transfert est para-unitaire. Ces bancs de filtres sont dits orthogonaux. Ainsi, à toute analyse multi-résolution correspond un banc de filtres orthogonal, mais la réciproque est fautive : il existe des bancs de filtres orthogonaux qui ne sont pas associés à des analyses multi-résolutions [35].

Imposer à un banc de filtres à reconstruction parfaite d'être orthogonal est très restrictif. En particulier il est bien connu qu'il n'existe qu'une seule analyse multi-résolution associée à des filtres RIF, c'est celle associée à l'ondelette de Haar. En affaiblissant la condition (1.29) pour que la fonction $\theta \in V_0$ vérifie simplement

$$\forall f \in V_0 \exists (a_k)_k \in \ell^2(\mathbb{Z}) : f(x) = \sum_{k \in \mathbb{Z}} a_k \theta(n-k) \text{ converge dans } L^2(\mathbb{R}),$$

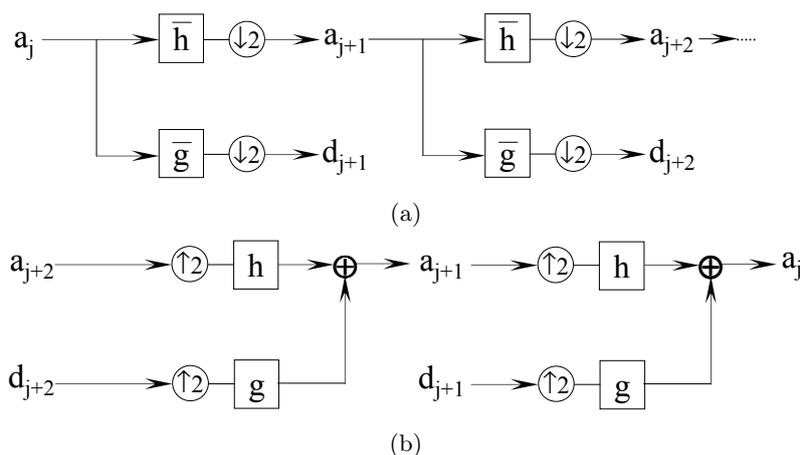


FIG. 1.8 – Utilisation d’un banc de filtre pour le calcul de la transformée en ondelette orthogonale : (a) décomposition, (b) recombinaison.

il est possible de construire des ondelettes, dites biorthogonales, et de leur associer des bancs de filtres bi-orthogonaux. Dans la suite, nous appellerons indifféremment ces bancs de filtres, bi-orthogonaux ou à reconstruction parfaite. Il existe des bancs de filtres bi-orthogonaux constitués uniquement de filtres RIF [38]. Un exemple de filtres de transformation en ondelettes biorthogonales est donné dans le tableau 1.1. Ce sont respectivement les filtres d’analyse des ondelettes de Daubechies [28] (9,7) et (5,3), qui sont utilisées dans le standard JPEG 2000, respectivement pour les modes de compression avec et sans pertes.

n	Coefficients (9,7)		Coefficients (5,3)	
	h	g	h	g
0	0,6029490182363579	1,115087052456994	6/8	1
± 1	0,2668641184428723	-0,5912717631142470	2/8	-1/2
± 2	-0,07822326652898785	-0,05754352622849957	-1/8	
± 3	-0,01686411844287495	-0,09127176311424948		
± 4	0,02674875741080976			

TAB. 1.1 – Coefficients des filtres d’analyse des ondelettes de Daubechies (9,7) et (5,3)[32].

Nous allons voir maintenant la représentation en *lifting scheme* de bancs de filtres bi-orthogonaux.

1.3.2 *Lifting scheme*

À l’origine, le but du *lifting scheme*, introduit par Sweldens [103], était de proposer un procédé de construction d’ondelettes biorthogonales dont les moments s’anulent pour des ordres de plus en plus élevés : “un ascenseur” (*lift*) vers de hauts moments nuls. C’est une alternative intéressante au schéma de filtrage convolutif classique de la transformée, car beaucoup moins complexe. En effet, le nombre d’opérations est divisé par un rapport allant jusqu’à deux en comparaison avec un schéma

classique. De plus, une implantation par *lifting scheme* est très économique en mémoire, qui ne dépend pas du nombre d'échantillons du signal à décomposer. Enfin, Sweldens et Daubechies ont montré que toute décomposition en ondelettes biorthogonale associée à des filtres RIF admet une représentation en *lifting scheme* [28]. Nous verrons par la suite que cette dernière permet de nouvelles libertés d'adaptation pour construire des transformations réversibles ou non.

Le *lifting scheme* consiste en plusieurs étapes (cf. fig. 1.9). En premier lieu, le

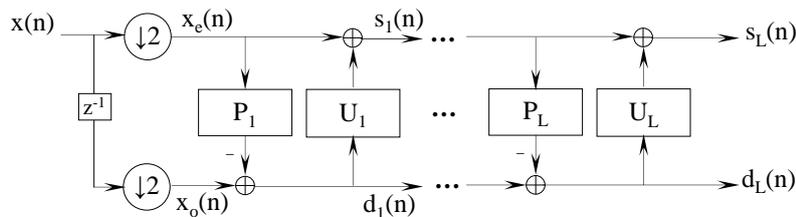


FIG. 1.9 – *Lifting scheme*.

signal d'entrée x est partitionné en plusieurs composantes par une décomposition polyphase. Généralement cette dernière est dyadique et les échantillons d'indices pairs x_e et d'indices impairs x_o sont séparés. Puis, un opérateur de prédiction est appliqué aux échantillons pairs et le résultat est soustrait aux échantillons impairs. Cette opération élémentaire, appelée étape de prédiction (*predict*) ou pas primal, donne une erreur de prédiction :

$$d_1 = x_o - P_1(x_e). \quad (1.40)$$

L'étape de mise à jour (*update*), ou pas dual, du *lifting scheme*, modifie les échantillons pairs avec l'erreur de prédiction. Le signal mis à jour s'exprime ainsi :

$$s_1 = x_e + U_1(d_1). \quad (1.41)$$

Dans le cas de filtres d'analyse à supports longs, plusieurs enchaînements de pas primaires et duaux P_i et U_i sont nécessaires à leur représentation en *lifting scheme*. Apparaissent alors des signaux intermédiaires d_i et s_i , $i = 1, \dots, L$. Au final, et après une éventuelle normalisation, un signal filtré passe-bas s_L (*smooth*) et un signal filtré passe-haut d_L (*detail*) sont obtenus.

Un grand intérêt pratique de cette représentation réside dans le fait que l'inversion de la transformée consiste simplement à changer les additions en soustractions et les soustractions en additions sans effectuer le calcul d'opérateurs inverses P_i^{-1} ou U_i^{-1} :

$$x_e = s_1 - U_1(d_1) \quad (1.42)$$

$$x_o = d_1 + P_1(x_e). \quad (1.43)$$

Les éventuelles erreurs d'arrondis commises à l'analyse dans les opérations $U_1(d_1)$ et $P_1(x_e)$ seront commises de la même manière à la synthèse. Toutefois, les erreurs d'arrondis commises lors d'une somme ne sont pas systématiquement compensées lors de la soustraction et vice versa, c'est pourquoi la représentation en *lifting scheme* de la figure 1.9 n'est pas systématiquement réversible.

Cette nouvelle méthode d'implantation d'une transformée en ondelettes trouve de nombreuses applications en traitement d'images. En compression, elle permet d'élaborer des bancs de filtres à reconstruction parfaite (linéaires ou non-linéaires) qui s'adaptent au signal. Par exemple, ces transformées utilisent des ondelettes à petit support autour des singularités et des ondelettes plus longues dans les régions où le signal est régulier [105, 34].

Toutes les structures que nous venons de présenter se généralisent pour s'appliquer aux signaux bi-dimensionnels (2-D).

1.4 Banc de filtres optimal en codage avec pertes

Dans cette section, nous nous intéressons au problème du banc de filtres optimal pour du codage par sous-bandes. Plus précisément, nous considérons un signal $(x(n))_{n \in \mathbb{Z}}$ qui est décomposé en M signaux de sous-bande $(v_i(m))_{m \in \mathbb{Z}}$ ($0 \leq i < M$) par un banc de filtres d'analyse comme sur la figure 1.6(a). Chacun de ces signaux est quantifié avec un quantificateur scalaire optimal (nous distinguerons deux cas). Les signaux quantifiés passent ensuite par un banc de filtres de synthèse (comme sur la figure 1.6(b)) pour donner un signal $y(n) = \hat{x}(n)$ qui est une approximation de $x(n)$. La distorsion entre le signal original et son approximation est mesurée par l'erreur quadratique moyenne

$$D(x, \hat{x}) = E[|x(n) - \hat{x}(n)|^2].$$

Dans la $i^{\text{ème}}$ sous-bande ($0 \leq i < M$), le signal quantifié est codé en moyenne avec b_i bits par échantillon. Quand le nombre moyen total de bits par échantillon $b = \sum_{i=0}^{M-1} b_i$ est fixé, quel est le meilleur choix pour les filtres H_i (ou de façon équivalente pour la représentation polyphase $\mathbf{E}(z)$ de la figure 1.7(b)), c'est-à-dire celui qui minimise l'erreur quadratique moyenne $E[|x(n) - \hat{x}(n)|^2]$? C'est l'énoncé du problème qui nous intéresse dans cette section et qui est le sujet de nombreux articles scientifiques. Nous limitons l'état de l'art sur le sujet en imposant le cadre restrictif suivant.

- Nous ne considérons que des bancs de filtres uniformes, c'est-à-dire dont le facteur M des sous-échantillonneurs est égal au nombre de canaux.
- Nous ne considérons que des bancs de filtres à reconstruction parfaite, c'est-à-dire pour lesquels $\hat{x}(n) = x(n)$ en absence de quantification.
- Nous ne considérons que des quantificateurs scalaires optimaux et indépendants (un par sous-bande) sous l'hypothèse haute résolution (voir le §2.2.2.1 au chapitre suivant).

Il est alors possible d'obtenir explicitement l'expression de l'erreur de quantification $E[(v - v^q)^2] = D(v, v^q)$ (où v^q est le résultat de la quantification appliquée à v) en fonction de la densité de probabilité $f_v(v)$ de v et de la résolution R du quantificateur :

$$D(v, v^q) \approx c \sigma_v^2 2^{-2R} \tag{1.44}$$

avec σ_v^2 la variance de v , $c = \frac{1}{12} \left(\int_{\mathbb{R}} [f_{\tilde{v}}(v)]^{1/3} dv \right)^3$ pour un quantificateur de Lloyd-Max (ou $c = \frac{1}{12} 2^{2h(\tilde{v})}$ pour un quantificateur uniforme suivi d'un codeur entropique d'ordre 1), \tilde{v} désignant la variable centrée réduite associée à v : $\tilde{v} = \frac{v - E(v)}{\sigma_v}$, $f_{\tilde{v}}(v)$ la

densité de probabilité de \tilde{v} et $h(\tilde{v}) = - \int f_{\tilde{v}}(u) \log_2 [f_{\tilde{v}}(u)] du$ son entropie différentielle [48].

Le cas particulier où la représentation polyphase est sans mémoire, *i.e.* $\mathbf{E}(z)$ est une matrice constante \mathbf{T} pose le problème du codeur par transformée optimal, qui a été étudié par Huang et Schultheiss en 1963 [58], puis plus en détails par Segal en 1976 [96]. Les principaux résultats de ces papiers sont les formules d'allocation optimale de bits (voir chapitre suivant) pour les quantificateurs des sous-bandes et la démonstration que la transformée de Karhunen-Loève (TKL) est optimale sous l'hypothèse que le signal d'entrée est gaussien et que les quantificateurs scalaires sont optimaux, c'est-à-dire vérifient une relation du type (1.44).

Dans le cas du codeur par sous-bande, la représentation polyphase $\mathbf{E}(z)$ a de la mémoire : $\mathbf{E}(z) = \sum_{n \in \mathbb{Z}} \mathbf{E}_n z^{-n}$ car les filtres $H_i(z)$ sont d'ordres quelconques. Il existe alors différentes méthodes pratiques qui optimisent numériquement les coefficients des filtres pour minimiser la distorsion finale, Akansu et Liu [12], Delsarte et Macq [41], Kiraç et Vaidyanathan [63], P. Moulin [78], Tuqan et Vaidyanathan [109]. Des résultats théoriques sur l'optimisation de bancs de filtres orthogonaux à deux canaux ont été développés par Unser en 1993 [110], avec une idée voisine du banc de filtres à composantes principales introduit par Tsatsami et Grannakis [108] en 1995 et Xuan et Bamberger [126] en 1996. Un ensemble général de conditions nécessaires et suffisantes d'optimalité, pour des bancs de filtres orthogonaux quelconques (sans contrainte sur les ordres des filtres), a été développé indépendamment par Vaidyanathan [114].

Le fait que les bancs de filtres bi-orthogonaux sont plus performants en pratique que les bancs de filtres orthogonaux est connu depuis longtemps et a été constaté dans de nombreux papiers (Antonini *et. al.* [14], Aase et Ramstad [6]). De même sous l'angle théorique, le fait que le gain de codage de bancs de filtres bi-orthogonaux peut dépasser celui des bancs de filtres orthogonaux est clair par exemple dans Djokovic et Vaidyanathan [43] ou Vaidyanathan [114]. Un papier important sur la théorie des bancs de filtres bi-orthogonaux, malgré des lacunes dans certaines preuves, est celui de Aas et Muller [5] et ceux de Vaidyanathan, écrits seul ou avec d'autres auteurs [114, 63, 115], ont largement inspiré cette section.

Indiquons pour terminer cette introduction, qu'imposer au banc de synthèse de satisfaire les conditions d'une reconstruction parfaite en absence de quantification n'est pas optimal, c'est-à-dire ne minimise pas la distorsion finale, quand le banc d'analyse est donné [76]. Il est préférable alors de rechercher le banc de synthèse qui, sous certaines contraintes de réalisabilité, minimise la distorsion [49]. Toutefois dans la suite de cette section, nous resterons dans le cadre donné ci-dessus.

1.4.1 Le banc de filtres à optimiser

Le banc de filtres (codeur en sous-bandes) à M canaux, utilisé généralement en codage avec pertes des signaux et des images, que l'on cherche à optimiser est représenté sur la figure 1.10. Les quantificateurs sont scalaires et optimaux. La représentation polyphase associée est montrée sur la figure 1.11. Dans chaque sous-bande, la quantification est supposée à haute résolution. Cela entraîne que les bruits de quantification sont blancs et deux à deux décorrélés, ils sont représentés par les sources

$q_i(n)$. Comme indiqué plus haut, ce système est un banc de filtres bi-orthogonal quand $\mathbf{R}(e^{j\omega})\mathbf{E}(e^{j\omega}) = \mathbf{I}$, pour tout ω , autrement dit, quand il est à reconstruction parfaite, c'est-à-dire $y(n) = x(n)$ en l'absence des quantificateurs. Ce banc de filtres est orthogonal, ou paraunitaire, quand la matrice $\mathbf{R}(e^{j\omega})$ est unitaire pour tout ω . Dans ce cas la biorthogonalité est vérifiée en posant $\mathbf{R}(e^{j\omega}) = \mathbf{E}^\dagger(e^{j\omega})$ ou d'une façon équivalente $F_i(e^{j\omega}) = H_i^*(e^{j\omega})$.

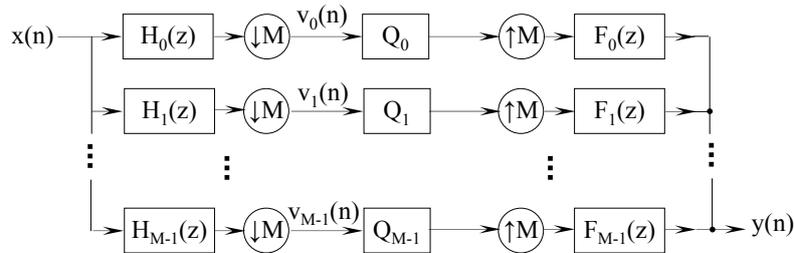


FIG. 1.10 – Schéma bloc d'un codeur en sous-bandes.

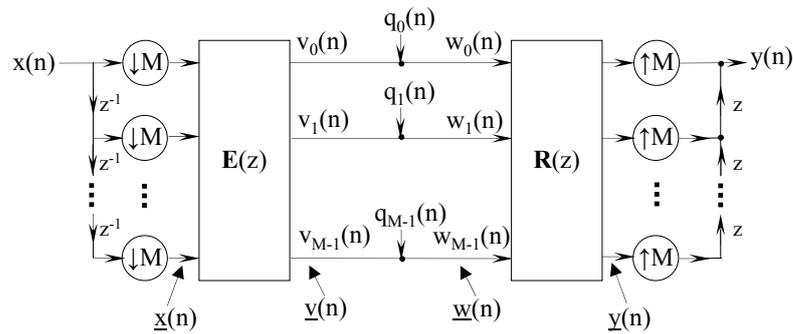


FIG. 1.11 – Représentation polyphase d'un codeur en sous-bandes.

1.4.2 Le signal à coder

Le signal d'entrée $x(n)$ est supposé stationnaire au sens large (WSS) avec une densité spectrale de puissance $S_{xx}(e^{j\omega})$ qui admet une factorisation forte [15]. Le vecteur $\underline{x}(m) = [x(mM), x(mM - 1), \dots, x(mM - M + 1)]^T$ est donc WSS et nous supposons enfin que sa matrice spectrale $\mathbf{S}_{xx}(e^{j\omega})$ est non singulière $\forall \omega \in \mathbb{R}$.

1.4.3 Les quantificateurs

Les bruits de quantification $q_i(n)|_{i=0, \dots, M-1}$ sont supposés blancs, deux à deux décorrélés et centrés, avec des variances $\sigma_{q_i}^2 \approx c_i 2^{-2b_i} \sigma_{v_i}^2$ qui décroissent exponentiellement avec les nombres de bits b_i alloués aux quantificateurs. Cette hypothèse est justifiée quand le débit total $b = \sum_{i=0}^{M-1} b_i / M$ est élevé (haut débit ou haute résolution). La constante c_i dépend de la forme de la densité de probabilité des grandeurs à quantifier. Nous supposons qu'elle est la même $c_i = c$ pour tous les signaux de sous-bandes, c'est le cas par exemple quand $x(n)$ est gaussien. Cette dernière hypothèse

est largement utilisée dans la théorie du codage par transformée ou en sous-bandes. Il est également supposé que les débits b_i sont distribués entre les quantificateurs avec une allocation optimale.

1.4.4 Position du problème d'optimisation

Si les signaux de sous-bandes sont quantifiés avec un budget total moyen b de bits par échantillon fixé, quel est le meilleur choix pour les filtres d'analyse (ou d'une manière équivalente, pour la matrice polyphase), c'est-à-dire celui qui minimise l'erreur quadratique moyenne de reconstruction

$$\varepsilon_{SBC} \triangleq \frac{1}{M} E[\underline{e}^\dagger(n)\underline{e}(n)] \quad (1.45)$$

avec $\underline{e}(n) = \underline{x}(n) - \underline{y}(n)$?

1.4.5 Cas d'un banc de filtres biorthogonal

Dans le cas où le banc de filtres est biorthogonal, l'erreur quadratique moyenne de reconstruction est donnée par [116] :

$$\varepsilon_{SBC} = c2^{-2b}\varphi^{1/M} \quad (1.46)$$

avec

$$\varphi \triangleq \prod_{i=0}^{M-1} \int_0^{2\pi} \left[\mathbf{E}(e^{j\omega}) \mathbf{S}_{xx}(e^{j\omega}) \mathbf{E}^\dagger(e^{j\omega}) \right]_{ii} d\omega/2\pi \times \int_0^{2\pi} \left[\mathbf{R}^\dagger(e^{j\omega}) \mathbf{R}(e^{j\omega}) \right]_{ii} d\omega/2\pi \quad (1.47)$$

où pour une matrice \mathbf{A} , $[A]_{i,j}$ désigne l'élément situé à l'intersection de la $(i+1)$ ^{ème} ligne et de la $(j+1)$ ^{ème} colonne. L'optimisation consiste alors à choisir $\mathbf{E}(e^{j\omega})$ et $\mathbf{R}(e^{j\omega})$, sous la contrainte de bi-orthogonalité $\mathbf{E}(e^{j\omega})\mathbf{R}(e^{j\omega}) = \mathbf{I}$, pour que la fonction φ soit minimale.

Le gain de codage est, par définition, le rapport de l'erreur quadratique moyenne de reconstruction $\varepsilon_{directe}$, lorsque le signal d'entrée est directement quantifié avec en moyenne b bits par échantillon, sur l'erreur quadratique moyenne ε_{SBC} de reconstruction, lorsque la quantification est appliquée sur les signaux de sous-bandes. Il vaut [116]

$$\begin{aligned} G_{SBC} &\triangleq \frac{\varepsilon_{directe}}{\varepsilon_{SBC}} = \frac{\sigma_x^2}{\varphi^{1/M}} \\ &= \frac{\sigma_x^2}{\left[\prod_{i=0}^{M-1} \int_0^{2\pi} S_{xx}(e^{j\omega}) |H_i(e^{j\omega})|^2 d\omega/2\pi \int_0^{2\pi} |F_i(e^{j\omega})|^2 d\omega/2\pi \right]^{1/M}} \quad (1.48) \end{aligned}$$

Sous les hypothèses mentionnées aux paragraphes 1.4.2 et 1.4.3 de cette section, P. Vaidyanathan et A. Kirac conjecturent que le banc de filtres bi-orthogonal optimal associé à $x(n)$ est celui de la figure 1.12, où $P(z)$ est le filtre demi-blanchisseur de $x(n)$, c'est-à-dire le filtre causal tel que

$$\forall \omega \in \mathbb{R} \quad |P(e^{j\omega})|^4 = S_{xx}(e^{j\omega})$$

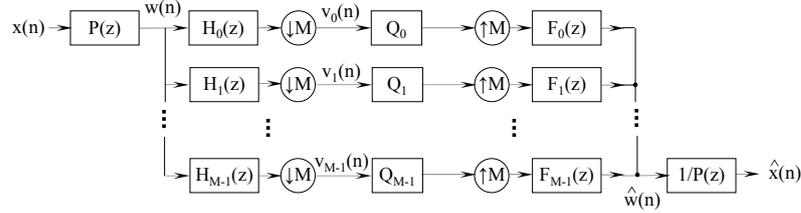


FIG. 1.12 – Cas particulier d'un banc de filtres bi-orthogonal optimal.

et dont le filtre inverse est également causal (son existence est assurée quand le signal $x(n)$ admet une factorisation forte). Enfin les filtres $H_i(z)$ et $F_i(z)$ sont ceux du banc de filtres orthogonal optimal associé à $w(n)$. Ils donnent une preuve de leur conjecture dans des cas restrictifs [116].

Voyons maintenant comment caractériser le banc de filtre orthogonal optimal.

1.4.6 Cas d'un banc de filtres orthogonal

Comme mentionné plus haut, un banc de filtres orthogonal est un banc de filtres bi-orthogonal avec $\mathbf{R}(e^{j\omega}) = \mathbf{E}^\dagger(e^{j\omega})$. En l'insérant dans la relation (1.47), la fonction à minimiser devient :

$$\varphi_{\text{orth}} \triangleq \prod_{i=0}^{M-1} \int_0^{2\pi} \left[\mathbf{E}(e^{j\omega}) \mathbf{S}_{xx}(e^{j\omega}) \mathbf{E}^\dagger(e^{j\omega}) \right]_{ii} d\omega / 2\pi = \prod_{i=0}^{M-1} \sigma_{v_i}^2 \quad (1.49)$$

où $\sigma_{v_i}^2$ est la variance du signal v_i de la $(i+1)$ ème sous-bande. Par conséquent, pour le cas orthogonal, l'optimisation du banc de filtres est équivalente à la minimisation du produit des variances des signaux de sous-bandes. Sous l'hypothèse haute résolution et avec une allocation optimale entre les quantificateurs et pour un signal gaussien (ou plus généralement satisfaisant les hypothèses du § 1.4.3), le gain de codage n'est que le rapport entre les moyennes arithmétique et géométrique des variances des signaux de sous-bandes :

$$G_{SBC} \triangleq \frac{\varepsilon_{\text{directe}}}{\varepsilon_{\text{SBC}}} = \frac{\sigma_x^2}{\left(\prod_{i=0}^{M-1} \sigma_{v_i}^2 \right)^{1/M}} = \frac{\frac{1}{M} \sum \sigma_{v_i}^2}{\left(\prod \sigma_{v_i}^2 \right)^{1/M}} \quad (1.50)$$

Notons que la maximisation du gain de codage, la minimisation de l'erreur quadratique moyenne de reconstruction et la minimisation de la fonction φ_{orth} sont équivalentes.

En codage par transformée orthogonale c'est-à-dire quand la matrice $\mathbf{E}(z)$ est sans mémoire et unitaire, la décorrélation entre les sous-bandes ($E[v_i(n)v_j^*(n)] = 0, i \neq j$) est une condition nécessaire et suffisante pour l'optimalité [71]. Cet optimum est atteint par la transformée de Karhunen-Loève. Pour les codeurs en sous-bandes orthogonaux une condition plus forte est nécessaire, à savoir

$$E[v_i(n)v_j^*(m)] = 0 \quad (1.51)$$

pour tous $i \neq j$ et pour tous $n, m \in \mathbb{Z}$. Cette condition correspond à la **décorrélacion totale** des signaux de sous-bandes. Elle n'est toutefois pas suffisante, dans [114]

il a été montré qu'une condition appelée, **majoration spectrale** est aussi nécessaire. Les signaux de sous-bandes $v_i(n)$ ont cette propriété quand leurs densités spectrales de puissance $\{S_{v_i}(e^{j\omega})\}$ vérifient :

$$S_{v_0}(e^{j\omega}) \geq S_{v_1}(e^{j\omega}) \geq \dots \geq S_{v_{M-1}}(e^{j\omega}), \text{ pour tout } \omega \in \mathbb{R} \quad (1.52)$$

où les sous-bandes sont numérotées pour que $\sigma_{v_i}^2 \geq \sigma_{v_{i+1}}^2$. Les propriétés de décorrélation totale et de majoration spectrale sont chacune nécessaire mais pas suffisante. Toutefois, ensemble elles deviennent suffisantes [114], ce qui donne le théorème suivant.

THÉORÈME 1. *Dans le cas d'un banc de filtres orthogonal, les conditions de décorrélation totale et de majoration spectrale sont nécessaires et suffisantes pour la minimisation de φ_{orth} (c-à-d la minimisation de l'erreur de reconstruction (1.46)).*

Dans [114], P. Vaidyanathan donne une méthode qui permet de construire le banc de filtres orthogonal optimal associé au signal $x(n)$. Les filtres d'analyse $H_i(z)$ solutions sont passe-bandes, leur gain vaut \sqrt{M} dans la bande passante et leurs réponses fréquentielles ont des supports deux à deux disjoints (ce qui assure la décorrélation totale des signaux de sous-bandes). Les supports ayant chacun pour longueur $2\pi/M$ sont choisis pour satisfaire à la condition de majoration spectrale.

Nous venons de voir qu'en codage en sous-bandes avec pertes, sous l'hypothèse d'une quantification scalaire à haute résolution optimale, avec une allocation optimale entre quantificateurs et quand tous les coefficients c_i de l'approximation (1.44) sont identiques (c'est le cas par exemple quand le signal à coder est gaussien), alors la distorsion D s'exprime en fonction du débit b :

$$D(x, \hat{x}) = cM2^{-2b}\phi^{1/M} \quad (1.53)$$

avec

$$\phi = \prod_{i=0}^{M-1} \sigma_{v_i}^2 \prod_{i=0}^{M-1} w_i$$

où

$$w_i = \int_0^{2\pi} \left[\mathbf{R}^\dagger(e^{j\omega}) \mathbf{R}(e^{j\omega}) \right]_{ii} \frac{d\omega}{2\pi} = \int_0^{2\pi} |F_i(e^{j\omega})|^2 \frac{d\omega}{2\pi}.$$

Les coefficients w_i valent 1 quand le banc de filtres est orthogonal et dans ce cas, un banc de filtres orthogonal sera d'autant plus efficace en codage avec pertes à haute résolution que le produit $\prod_{i=0}^{M-1} \sigma_{v_i}^2$ des variances des signaux de sous-bandes sera petit.

Nous avons vu également que, pour un signal stationnaire au sens large (WSS), le banc de filtres orthogonal optimal est constitué de filtres idéaux passe-bandes. Ces derniers étant difficiles à réaliser en pratique, et pouvant nécessiter un grand nombre de coefficients (en théorie ce sont des filtres RII qui ne coïncident pas exactement avec des filtres ARMA), il peut être intéressant de renoncer au calcul de ces filtres et de préférer imposer une structure de filtres (par exemple RIF d'ordres donnés) pour laquelle le problème consiste à rechercher les filtres qui minimisent la distorsion $D(x, \hat{x})$ pour un débit b donné et sous les hypothèses rappelées ci-dessus. Nous voyons

alors que le critère qu'il faudra minimiser en codage avec pertes est le produit des variances des signaux de sous-bandes. Dans le cas de bancs de filtres bi-orthogonaux, le critère qu'il faudra minimiser est le produit des variances des signaux de sous-bandes multiplié par le produit des facteurs w_i qui dépendent des filtres F_i du banc de synthèse.

Il est bien connu qu'en compression sans perte, le critère qu'il faut minimiser pour optimiser le codage est encore le produit des variances des signaux de sous-bandes et ceci même pour un banc de filtres à reconstruction parfaite non orthogonal [82]. Toutefois, n'ayant pas trouvé dans la littérature de démonstration de ce résultat, nous en proposons une au début du chapitre 3 de ce rapport de thèse.

1.5 Banc de filtres, ondelettes et *lifting scheme* 2-D

Les signaux bi-dimensionnels que nous considérons sont des images numériques à support fini. Nous pouvons représenter une image ayant N_ℓ lignes et N_c colonnes par une matrice \mathbf{X} rectangulaire de dimension $N_\ell \times N_c$. Introduisons également l'opérateur *vec* qui, appliqué à une image \mathbf{X} , met toutes ses colonnes les unes au-dessous des autres (la première en haut et la dernière en bas) pour obtenir un vecteur $\text{vec}(\mathbf{X})$ de dimension $N_\ell N_c$.

Les opérations de sous-échantillonnage, sur-échantillonnage, filtrage se généralisent, comme cela est bien connu, pour s'appliquer à des signaux 2-D. Notons \mathbf{Y} l'image obtenue après une décomposition appliquée à \mathbf{X} . Par linéarité, cette opération s'écrit $\text{vec}(\mathbf{Y}) = \mathcal{A} \cdot \text{vec}(\mathbf{X})$, où \mathcal{A} est une matrice. Quand la décomposition consiste à appliquer une transformation \mathbf{A}_ℓ sur les lignes et une transformation \mathbf{A}_c sur les colonnes de \mathbf{X} , c'est-à-dire quand $\mathbf{Y} = \mathbf{A}_c \mathbf{X} \mathbf{A}_\ell^T$ (l'ordre d'application ne modifie pas le résultat), elle est dite séparable.

En général dans les *lifting schemes* qui s'adaptent aux images les filtres prédicteurs ne sont pas séparables.

1.6 *Lifting schemes* adaptés en compression

L'adaptation du *lifting scheme* en fonction du signal d'entrée est principalement utilisée dans un cadre de codage de signaux et principalement des images. Nous recensons dans cette section les structures de ce type qui existent déjà. Certaines ont été développées pour du codage avec pertes, mais il est facile de rendre les structures réversibles (voir le §2.4.2.3 au chapitre suivant) pour les utiliser dans un cadre sans perte.

1.6.1 Adaptation par sélection de l'ordre du prédicteur

La structure en *lifting scheme* proposée par Claypool *et al.* [34] utilise un prédicteur basé sur les propriétés locales de l'image, de sorte que dans la fenêtre de prédiction, les contours qui correspondent aux non stationnarités, soient "évités". Pour chaque fenêtre de prédiction, les données sont analysées. Si ces dernières présentent une certaine régularité, alors l'ordre du filtre prédicteur est grand. Dans le

cas contraire, les pixels responsables de cette “irrégularité” sont déterminés et classifiés comme étant des pixels appartenant à un contour. À proximité de ces contours, l’ordre du filtre prédicteur est réduit, de sorte que le voisinage utilisé pour la prédiction ne chevauche pas le contour. De cette manière, les grandes erreurs de prédiction sont évitées à proximité des contours. La figure 1.13 illustre la procédure de sélection de l’ordre du filtre prédicteur à proximité d’un contour.

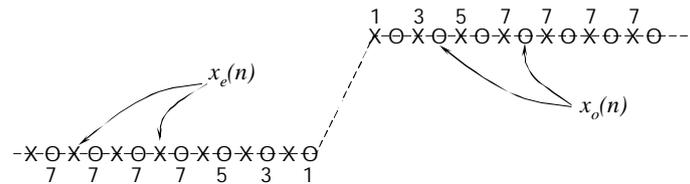


FIG. 1.13 – Sélection de l’ordre du prédicteur. Les nombres indiquent l’ordre du prédicteur utilisé. Il est plus petit à proximité du contour.

1.6.2 Adaptation sous contrainte

Les travaux menés par A. Gouze [50] portent sur la transformée en ondelettes appliquée en compression d’images. L’objectif central était la factorisation d’un banc de filtres bi-dimensionnels non séparables en *lifting scheme*. Ce dernier est par la suite construit en fonction des propriétés statistiques de l’image, dans le but de fournir un algorithme de compression approprié et efficace pour des images directement échantillonnées en quinconce. La méthode proposée se base sur un *lifting scheme* à deux pas.

Le processus d’optimisation définit en premier lieu un opérateur de prédiction. L’auteur part du principe que le codage est plus performant sur un signal de faible variance. Le critère retenu pour définir le pas de prédiction consiste, par conséquent, en la minimisation de la variance de l’image de la sous-bande haute fréquence. Une contrainte est imposée permettant de garder une moyenne nulle aux coefficients de la sous-bande haute fréquence. Cela se traduit par une somme égale à 1 des coefficients du filtre prédicteur. Le critère à minimiser sous contrainte s’exprime de la manière suivante :

$$J(\underline{p}, \lambda) = \sigma_d^2 + \lambda \left(1 - \sum_{k \in S_1} p_k \right) \quad (1.54)$$

avec σ_d^2 la variance de l’image de la sous-bande haute fréquence, p_k les coefficients du filtre prédicteur P regroupés dans le vecteur \underline{p} et λ le poids de la contrainte (multiplicateur de Lagrange). Le calcul des coefficients optimaux est fait en annulant les dérivées partielles de ce critère par rapport à λ et p_k .

L’opérateur de mise à jour U agit sur la composante basse résolution de l’image. Il est efficace quand il retourne une image basse fréquence offrant une représentation fidèle de l’image source à une résolution inférieure. L’image haute fréquence est destinée à subir des dégradations plus ou moins fortes, lorsque le schéma de décomposition est associé à un quantificateur générant des pertes d’information. Il est donc

primordial de pouvoir assurer une reconstruction de l'image la plus fiable possible à partir de la seule image basse fréquence. La synthèse du filtre U détermine l'erreur de reconstruction minimale, en fixant l'image de sous-bande haute fréquence à zéro. Une contrainte supplémentaire est imposée, elle consiste en la conservation de la moyenne dans la sous-bande basse fréquence. Cette contrainte est indispensable, lors de l'implantation, pour éviter tout risque de dépassement de capacité (explosion des coefficients) après l'application du *lifting scheme* sur les approximations successives de l'image originale. Le critère est défini comme l'erreur quadratique moyenne de reconstruction sous contraintes :

$$J(\underline{u}, \lambda) = E[(x - \hat{x})^2] + \lambda \left(\frac{1}{2} - \sum_{k \in S_2} u_k \right) \quad (1.55)$$

où \underline{u} représente le vecteur regroupant tous les coefficients de l'opérateur U , x l'image originale et \hat{x} l'image reconstruite en fixant l'image de sous-bande haute fréquence à zéro. Les coefficients du filtre de mise à jour sont calculés en annulant les dérivées partielles par rapport à u_k et λ .

Il faut noter que l'optimisation des filtres 2-D de décomposition en *lifting scheme* avec un sous-échantillonnage en quinconce, présentée dans cette section et développée par A. Gouze dans [50], s'applique à la compression avec pertes et les filtres calculés sont des filtres 2-D non séparables (voir le § 1.6.4 à la fin de cette section).

1.6.3 Adaptation pour des signaux WSS avec post-traitement non-linéaire

Dans le cas de la compression sans perte des images, une méthode de calcul des filtres prédicteurs optimaux d'un *lifting scheme* est proposée par N.V. Boulgouris *et al.* [27]. Plusieurs cas sont traités suivant le choix de la décomposition polyphase (en quinconce ou non) de l'image à coder. Dans chaque cas, l'efficacité des prédicteurs linéaires est améliorée en utilisant une non-linéarité. Un post-traitement directionnel est employé dans le cas d'un sous-échantillonnage en quinconce. Dans le cas d'un sous-échantillonnage séparable, le post-traitement consiste en une adaptation de la longueur du filtres prédicteur.

Le schéma traité est un *lifting scheme* multidimensionnel généralisé montré à la figure 1.14, cette forme spécifique a été présentée dans [27]. La matrice de sous-échantillonnage est notée \mathbf{M} , avec $M = \det \mathbf{M}$ le nombre de composantes polyphases. Les filtres P_i sont les filtres multi-dimensionnels de prédiction et les filtres U_i sont ceux de mise à jour (*update*), avec $i = 1, \dots, M - 1$. Le signal d'entrée x est n -dimensionnel, supposé stationnaire au sens large, avec une fonction d'autocorrelation

$$R(\underline{s}) = E[x(\underline{m} + \underline{s})x(\underline{m})]$$

où $\underline{s} = [s_1, \dots, s_n]^T$ et $\underline{m} = [m_1, \dots, m_n]^T$, et une densité spectrale de puissance

$$\Phi(e^{j\omega}) = \sum_{\underline{s}} R(\underline{s}) e^{-j\omega^T \underline{s}}$$

où la somme ci-dessus est une somme multiple sur les $(s_i)_i$ et $\omega = [\omega_1, \dots, \omega_n]^T$. Les composantes polyphases x_i , $i = 0, \dots, M - 1$ du signal d'entrée x sont données par

$$x_i(\underline{m}) = x(\mathbf{M}\underline{m} - \underline{r}_i),$$

et les sorties des filtres prédicteurs sont les M signaux de sous-bandes

$$\hat{x}_i(\underline{m}) = \sum_{\underline{k}} p_i(\underline{\mathbf{M}}\underline{k})x[\underline{\mathbf{M}}(\underline{m} - \underline{k})], \quad 0 \leq i < M$$

avec p_i la réponse impulsionnelle du filtre prédicteur P_i . Les filtres prédicteurs optimaux sont définis par les auteurs comme étant ceux qui minimisent les variances des erreurs de prédiction e_i dans les sous-bandes, avec

$$e_i(\underline{m}) = x(\underline{\mathbf{M}}\underline{m} - \underline{r}_i) - \sum_{\underline{k}} p_i(\underline{\mathbf{M}}\underline{k})x[\underline{\mathbf{M}}(\underline{m} - \underline{k})].$$

Ils sont solutions du système d'équations

$$R(\underline{\mathbf{M}}\underline{t} - \underline{r}_i) = \sum_{\underline{k}} p_i(\underline{\mathbf{M}}\underline{k})R[\underline{\mathbf{M}}(\underline{t} - \underline{k})].$$

Moyennant quelques développements (plus de détails dans [27]), une expression analytique permettant de déduire les filtres prédicteurs optimaux est obtenue :

$$P(e^{j\omega}) = 1 + \sum_{i=1}^{M-1} e^{j\underline{r}_i^T \omega} P_i(e^{j\underline{\mathbf{M}}^T \omega}) = \frac{\Phi(e^{j\omega})}{\frac{1}{M} \sum_{i=0}^{M-1} \Phi(e^{j(\omega - 2\pi \underline{\mathbf{M}}^{-T} \underline{q}_i)})}. \quad (1.56)$$

D'où le théorème suivant

THÉORÈME 2. *Les filtres prédicteurs de structure de décomposition en lifting scheme de la figure 1.14 sont optimisés au sens de la minimisation des variances des erreurs de prédictions dans les M sous-bandes si leurs fonctions de transferts $P_i(e^{j\underline{\mathbf{M}}^T \omega})$ sont les composantes polyphases de la fonction $P(e^{j\omega})$ donnée par l'équation (1.56).*

Par conséquent, la connaissance de la densité spectrale de puissance du signal (ou de façon équivalente sa fonction d'autocorrélation) suffit pour déterminer les filtres prédicteurs optimaux.

Dans [27] et pour le cas des images, deux modèles de fonction d'autocorrélation sont supposés (le sous-échantillonnage utilisé est en quinconce) pour la détermination des filtres prédicteurs optimaux. Le premier est un modèle séparable

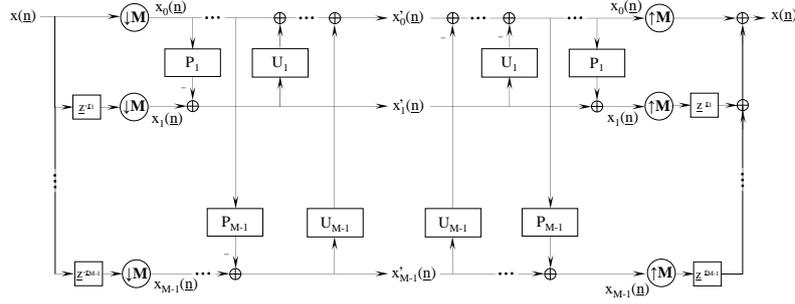
$$R(s_1, s_2) = A\lambda_1^{|s_1|}\lambda_2^{|s_2|}$$

et le second non séparable

$$A\lambda_1^{\sqrt{s_1 s_2}}$$

avec A , λ_1 , λ_2 et λ les paramètres des modèles. Pour permettre au prédicteur d'être également efficace dans un environnement non stationnaire, un post-traitement est appliqué. Il est directionnel dans le cas d'un sous-échantillonnage en quinconce. Dans le cas d'un sous-échantillonnage séparable, le modèle de la fonction d'autocorrélation adopté est unidimensionnel $R(s) = A\lambda^{|s|}$ et la prédiction est améliorée par une non-linéarité consistant à adapter la longueur des filtres.

L'étape de mise à jour (*update*) est fixe et correspond à celle d'une ondelette classique, de telle sorte que l'image d'approximation est basse fréquence (lisse), ce qui est souhaitable pour le codage.

FIG. 1.14 – *Lifting scheme* multidimensionnel à M sous-bandes.

1.6.4 Adaptation basée sur l’algorithme du gradient stochastique

Une structure en *lifting scheme* adaptative basée sur l’algorithme du gradient stochastique a été proposée par Gerek et Çetin [46, 47]. Les coefficients du filtre prédicteur sont mis à jour au fur et à mesure du balayage des pixels de l’image. Le prédicteur adaptatif utilisé pour générer le signal de détails $x_h(n)$ est représenté à la figure 1.15. Il est RIF et obtenu en faisant une prédiction des valeurs du signal $x_2(n)$ à partir d’échantillons du signal $x_1(n)$:

$$\hat{x}_2(n) = \sum_{k=-p}^p a_{n,k} x_1(n-k). \quad (1.57)$$

Les coefficients du filtre $(a_{n,k})_k$ sont mis à jour avec l’algorithme du gradient stochastique [15]

$$\hat{a}(n+1) = \hat{a}(n) + \mu \frac{\tilde{x}_n x_h(n)}{\|\tilde{x}_n\|^2} \quad (1.58)$$

où $\hat{a}(n) = [a_{n,-p}, \dots, a_{n,p}]^T$ est le vecteur contenant les coefficients du filtre à l’instant n et

$$\tilde{x}_n = [x_1(n-p), x_1(n-p+1), \dots, x_1(n+p-1), x_1(n+p)]^T$$

le vecteur observation à l’instant n . Le paramètre μ est le pas d’adaptation, il influence la vitesse de convergence de l’algorithme. Le signal de détails x_h vaut alors

$$x_h(n) = x_2(n) - \sum_{k=-p}^p a_{n,k} x_1(n-k) = x_2(n) - \hat{a}(n)^T \tilde{x}_n \quad (1.59)$$

Bien que les coefficients du filtre prédicteur soient variables en fonction de n , la propriété de reconstruction parfaite reste assurée sans qu’il soit nécessaire de les transmettre car ils seront calculés au décodeur avec le même algorithme d’adaptation.

1.7 Conclusion

Dans ce chapitre nous avons introduit des éléments de la théorie des signaux et systèmes pour comprendre la synthèse des bancs de filtres appliqués au codage. La

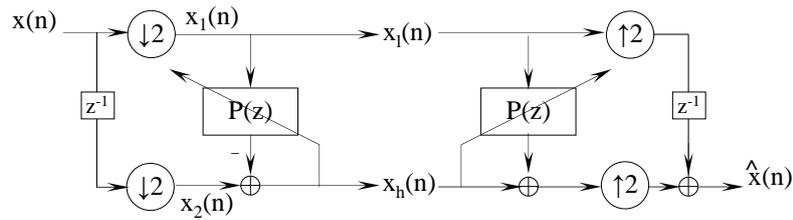


FIG. 1.15 – Structure des bancs de filtres d'analyse et de synthèse adaptatifs proposées par Gerek et Çetin [47].

représentation en *lifting scheme* de bancs de filtres à reconstruction parfaite a permis l'interprétation des signaux de sous-bandes en termes de résidus de prédiction. Il était alors naturel d'associer à la théorie des bancs de filtres, les résultats de l'estimation linéaire en moyenne quadratique, comme cela a été fait par Gerek et Çetin dans [46]. Toutefois, dans leur approche ces derniers n'exploitent pas toute l'information disponible au décodeur, pour adapter leur filtre prédicteur. En effet, les pixels de la sous-bande x_2 précédant le pixel courant (suivant l'ordre de balayage de l'image) sont disponibles pour estimer la valeur courante. L'objectif de notre étude a été d'exploiter (par des outils linéaires) toute l'information disponible au décodeur pour connaître la limite de performance du codage par prédiction linéaire associé à des décompositions hiérarchiques pyramidales. Dans le chapitre suivant nous présentons les éléments de la théorie du codage utiles à notre étude.

CHAPITRE 2

Compression d'images

2.1 Introduction

La compression est l'opération qui consiste à réduire le nombre de bits utiles pour représenter des signaux porteurs d'informations. Il existe deux types de compression : sans perte et avec pertes. Dans ce chapitre nous rappelons les notions élémentaires de la théorie du codage avec et sans pertes.

La section 2.2 présente la structure globale d'un système de compression avec pertes : la transformation, les méthodes classiques de quantifications scalaire et vectorielle, le problème d'allocation optimale et le codage entropique, qui constitue la dernière étape du codeur, sont décrits. Puis nous rappelons les notions de codage progressif. Dans la section 2.4 nous présentons la compression sans perte d'images en nous focalisant sur les transformations utilisant une étape de prédiction. Différents types de prédiction sont détaillés. Enfin, la section 2.5 fait un rappel sur les transformations apparaissant dans les principaux systèmes de compression sans perte.

2.2 Compression avec pertes

La structure d'un codeur d'images avec pertes telle qu'elle sera considérée dans cette thèse est montrée à la figure 2.1. Elle se compose de trois parties étroitement liées : la transformation, la quantification et le codage entropique. La compression est accomplie par l'application d'une transformation à l'image pour réduire la redondance entre pixels, suivie de l'étape de quantification des coefficients de l'image transformée. Le fait d'utiliser un canal de transmission ou un support de stockage limité en débit ou en taille nous oblige à contrôler le débit du codeur en ajustant les pas de quantification. Cette allocation cherche à obtenir un meilleur compromis débit-distorsion, c'est-à-dire une qualité de reconstruction maximale pour un débit donné ou un débit minimal pour une qualité donnée. Elle sera étudiée dans le paragraphe 2.2.3. En fin de chaîne vient l'étape du codage entropique des valeurs quantifiées, elle est complètement réversible (en absence d'erreur de transmission). Chacune de ces étapes va maintenant être développée séparément. Le décodage est

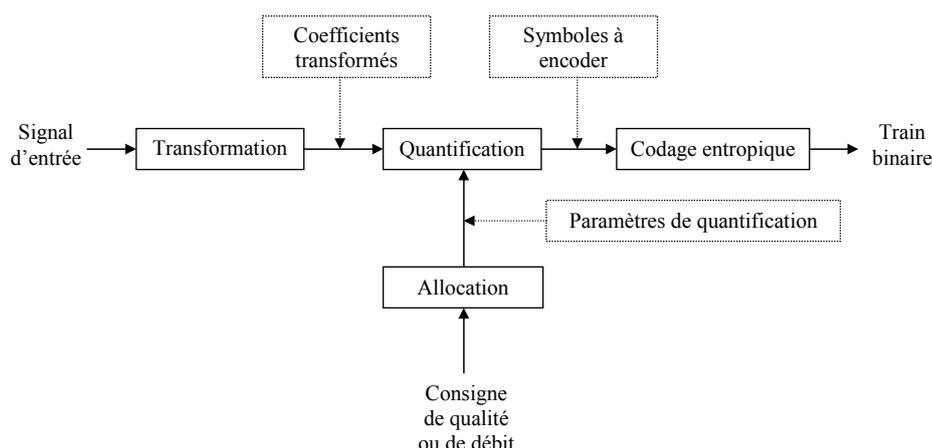


FIG. 2.1 – Schéma type d'un système de compression avec pertes d'images.

tout simplement le cheminement inverse de la compression.

2.2.1 Transformation

Une caractéristique commune à la plupart des images naturelles est la forte corrélation entre les pixels voisins et donc de l'information redondante. Dans un schéma de compression, la transformation permet de réduire ces corrélations, par prédiction dans le cas d'un codage prédictif, par transformée dans le cas d'un codage par transformée ou par banc de filtres dans le cas d'un codeur en sous-bandes. Cette réduction des corrélations augmente considérablement l'efficacité du codage entropique.

Lorsque la transformation est linéaire, cela revient à un changement de base dans l'espace de l'image. Le but d'un codage par transformée est donc de trouver une base dans laquelle, d'une part les échantillons du signal se trouvent naturellement décorrélés, et d'autre part les informations pertinentes sont facilement identifiables en vue d'un codage avec pertes. Parmi les principales transformées utilisées en codage d'images, citons :

- la transformée de Karhunen-Loève, optimale sous certaines conditions mais peu usitée en raison de sa complexité¹,
- la transformée en cosinus discrète (*Discrete Cosine Transform*, DCT), fruit de la théorie de Fourier,
- la transformée en ondelettes discrète (*Discrete Wavelet Transform*, DWT) et les décompositions par bancs de filtres.

Exemples : Transformée en cosinus discrète. L'état de l'art sur le codage d'images ne peut pas se faire sans évoquer la DCT. Notons d'abord qu'elle est très présente dans les standard et normes existants : tous, à l'exception de JPEG2000, l'utilisent comme transformée. La DCT est une transformation orthogonale. Elle est issue de la théorie de Fourier, elle est donc particulièrement adaptée pour décrire des

¹La transformée de Karhunen-Loève consiste à rechercher les composantes principales du signal, en diagonalisant la matrice de sa fonction d'autocovariance en zéro. C'est une transformation orthogonale. Elle correspond à l'analyse en composantes principales (ACP) des cours de statistiques.

signaux stationnaires, ce qui n'est malheureusement pas le cas des images naturelles. C'est la raison pour laquelle, en pratique, l'image est d'abord découpée en blocs (traditionnellement de taille 8x8, mais aussi 4x4, 4x8 ou 8x4), avant de transformer chaque bloc indépendamment.

Transformation en ondelettes (9,7) de Daubechies. Dans le cas de la compression avec pertes où le PSNR et d'autres mesures subjectives de performance sont utilisés, la transformation (9,7) est parmi celles qui donnent les meilleurs résultats. Elle présente de modestes performances en compression sans perte. C'est une transformation quasi-orthogonale dans le sens où les colonnes de la matrice associée sont presque orthogonales deux à deux. Avec les notations du § 1.5 nous avons $\mathbf{A}\mathbf{A}^T \approx \text{diag}(\mathbf{A}\mathbf{A}^T)$ où $\text{diag}(\mathbf{A})$ est la matrice diagonale constituée de la diagonale de \mathbf{A}

2.2.2 Quantification

Les coefficients à la sortie de la transformation (ou des filtres d'analyse dans le cas d'un banc de filtres) prennent en général des valeurs réelles qui doivent être quantifiées pour réaliser la compression. Une étude détaillée de la quantification est réalisée dans [48]. La quantification dans sa forme la plus simple consiste à quantifier chaque coefficient séparément, on parle alors de quantification scalaire. Une autre méthode plus complexe consiste à quantifier plusieurs coefficients à la fois, c'est la quantification vectorielle, qui sous différentes alternatives, fournit théoriquement la performance optimale pour tout codeur [48].

2.2.2.1 La quantification scalaire

Avec une quantification scalaire, la dynamique d'entrée est divisée en une famille finie d'intervalles $P_i = [x_{i-1}, x_i[$ et la valeur de sortie y_i est typiquement choisie dans l'intervalle P_i .

L'opération appelée "codage" consiste à trouver l'intervalle auquel appartient une valeur d'entrée $x(n)$ et à lui associer le numéro de cet intervalle $i(n) \in 1, \dots, L$ qui sera transmis ou stocké. L'opération de "décodage" consiste à associer au numéro $i(n)$ la valeur $y_{i(n)}$ correspondante dans le dictionnaire :

$$\hat{x}(n) = y_{i(n)}.$$

Ainsi, la quantification est un processus à deux étapes :

$$x(n) \xrightarrow{\text{codage}} i(n) \xrightarrow{\text{décodage}} y_i(n) = \hat{x}(n)$$

et l'erreur de quantification est exprimée par

$$q(n) = x(n) - \hat{x}(n)$$

où $x(n)$ et $\hat{x}(n)$ sont respectivement l'entrée et la sortie du quantificateur. La performance d'un quantificateur est mesurée par la distorsion entre les valeurs d'entrée

et de sortie au sens de l'erreur quadratique moyenne (EQM), qu'on cherche souvent à minimiser, ou par le rapport signal sur bruit (SNR) qu'il s'agit de maximiser

$$D = E[|X(n) - \hat{x}(n)|^2]$$

$$SNR = \frac{E[X^2(n)]}{E[|X(n) - \hat{x}(n)|^2]}$$

Sous l'hypothèse haute résolution [48], c'est-à-dire quand

1. le nombre L de niveaux de quantification est élevé,
2. la densité de probabilité de X peut être considérée comme constante dans chaque intervalle P_i , $i = 1, \dots, L$,
3. les représentants sont pris aux milieux des intervalles de quantification.

La quantification scalaire uniforme Soit le signal à temps discret $x(n)$ prenant ses valeurs dans l'intervalle $[a, b]$, nous le considérons comme la réalisation d'un processus aléatoire $X(n)$ centré. La résolution R est fonction du nombre L d'intervalles divisant la dynamique d'entrée $[a, b]$. Pour définir un quantificateur scalaire uniforme, il faut :

1. partitionner l'intervalle $[a, b]$ en $L = 2^R$ intervalles distincts de même longueur $q = (b - a)/2^R$;
2. associer un numéro à chaque intervalle "i";
3. définir un représentant par intervalle "y_i", par exemple le milieu de l'intervalle.

L'erreur de quantification s'écrit alors [48] :

$$\sigma_Q^2 \approx \frac{q^2}{12} = \frac{(b - a)^2}{12} 2^{-2R}.$$

La variance du signal $X(n)$ est proportionnelle à $(b - a)^2$ et le rapport signal sur bruit exprimé en dB vaut

$$SNR = 10 \log_{10} \frac{E[X^2(n)]}{E[q^2(n)]} = 6,02R + C^{te}$$

Le fait d'augmenter la résolution d'un bit revient donc à augmenter le rapport signal à bruit d'environ 6 dB.

La quantification scalaire non-uniforme Soit $f_X(x)$ la densité de probabilité du processus aléatoire $X(n)$, la distorsion moyenne s'écrit

$$D = \sigma_Q^2 = \sum_{i=1}^L \int_{x \in P_i} d(x, y_i) f_X(x) dx \quad (2.1)$$

où P_i est l'élément de la partition associé au numéro i , sachant qu'il y a L cellules P_i , ($i = 1, \dots, L$) distinctes.

Un quantificateur optimal est celui qui minimise D dans (2.1) pour un nombre L donné de cellules. Pour définir ce quantificateur, il s'agit de trouver la partition

$\{P_1, \dots, P_L\}$ et les symboles de sortie $\{y_1, \dots, y_L\}$ qui minimisent D . Cette minimisation conjointe n'admet pas de solution simple. Il existe deux conditions nécessaires d'optimalité [48, 77]. Si les symboles de sortie $\{y_1, \dots, y_L\}$ sont connus, la meilleure partition $\{P_1, \dots, P_L\}$ peut être calculée. Si la partition est donnée, les meilleurs représentants peuvent être déduits. En d'autres termes, la partie "codage" du quantificateur peut être optimisée étant donnée la partie "décodage" et réciproquement. Les deux conditions nécessaires d'optimalité (lorsque l'erreur quadratique moyenne est utilisée comme mesure de distorsion) sont :

- la règle du plus proche voisin,
- la condition du centroïde.

La règle du plus proche voisin. Étant donné un dictionnaire de symboles $\{y_1, \dots, y_L\}$, la meilleure partition est celle qui vérifie

$$P_i = \{x : (x - y_i)^2 \leq (x - y_j)^2 \quad \forall j \in \{1, \dots, L\}\}$$

Et si l'on appelle x_i la valeur définissant la frontière entre les partitions P_i et P_{i+1} , la minimisation de D est obtenue pour

$$x_i = \frac{y_i + y_{i+1}}{2},$$

donc la meilleure partition est celle qui correspond aux points milieux entre deux symboles de sortie adjacents.

La condition du centroïde. Étant donnée une partition $\{P_1, \dots, P_L\}$, les meilleurs symboles $\{y_1, \dots, y_L\}$ sont les centres de gravité de la densité de probabilité dans les régions P_i

$$y_i = \frac{\int_{x \in P_i} x f_X(x) dx}{\int_{x \in P_i} f_X(x) dx} = E[X|X \in P_i]. \quad (2.2)$$

Les deux conditions citées plus haut sont à la base de l'algorithme itératif de Lloyd-Max [67] qui améliore à chaque itération le dictionnaire, ayant la fonction de densité de probabilité et le nombre de symboles L . Avec l'initialisation du dictionnaire à $\{y_i^{(0)}\}_i$, l'algorithme comprend les deux étapes suivantes.

1. Ayant $\{y_i^{(0)}\}_i$, trouver la meilleure partition $(P_i)_i$, satisfaisant à la condition du plus proche voisin.
2. Pour la partition $(P_i)_i$ (donnée ou celle trouvée à l'étape 1), calculer le meilleur dictionnaire $\{y_i^{(n+1)}\}_i$ satisfaisant à la condition du centroïde.

L'algorithme converge quand la distorsion D reste quasiment constante. Il assure généralement la décroissance de la distorsion moyenne mais ne tend pas toujours vers le minimum global, il peut converger vers un minimum local.

Quand les hypothèses d'une quantification scalaire à haute résolution sont satisfaites, on peut montrer [48] que la distorsion minimale vérifie :

$$D \approx \frac{1}{12L^2} \left(\int_{\mathbb{R}} [f_X(u)]^{1/3} du \right)^3. \quad (2.3)$$

2.2.2.2 La quantification vectorielle

La quantification vectorielle n'est qu'une généralisation de la quantification scalaire à plusieurs dimensions [48]. Elle permet de prendre en compte directement la corrélation entre échantillons voisins du signal, plutôt que de chercher d'abord à réduire la redondance au moyen d'une transformation avant de quantifier, comme le fait la quantification scalaire prédictive. La quantification vectorielle est appliquée à chaque vecteur $\underline{x}(m)$ de dimension N constitué de N échantillons consécutifs du signal. Les vecteurs

$$\underline{\hat{y}}^i = [\hat{y}_0^i, \dots, \hat{y}_{N-1}^i]^T$$

représentent les vecteurs symboles à la sortie du quantificateur. L'ensemble des L vecteurs $\underline{\hat{y}}^i$, assimilable à une matrice est le dictionnaire. Ce dictionnaire C associé à une partition $(P_i)_i$ de l'espace \mathbb{R}^N en L régions définit le quantificateur vectoriel $Q : \mathbb{R}^N \rightarrow C$ par $Q(\underline{x}) = \hat{y}^i$ si et seulement si $\underline{x} \in P_i$.

Un quantificateur vectoriel optimal au sens des moindres carrés satisfait aux conditions décrites pour les quantificateurs scalaires, c-à-d la règle du plus proche voisin et la condition du centroïde. Notons que ces deux conditions d'optimalité ne signifient pas une optimalité globale, mais locale. La construction d'un dictionnaire de quantification vectorielle est une technique très sophistiquée qui consiste en une bonne initialisation suivie d'une procédure itérative comme l'algorithme de Lloyd-Max détaillé dans [48].

L'inconvénient de la quantification vectorielle est sa complexité, qui limite la taille des vecteurs $\underline{\hat{y}}^i$ utilisés. Pour contourner ce problème le dictionnaire est structuré de façon à simplifier la recherche du représentant, étant donné le vecteur d'entrée. Ceci est réalisable avec la quantification vectorielle arborescente. Une autre approche est d'utiliser des transformations linéaires (telle que la décomposition en sous-bandes et les transformées en ondelettes) et d'appliquer la quantification vectorielle sur les coefficients. Moreau, dans [77], décrit d'une façon détaillée les différentes alternatives pour la quantification vectorielle ainsi que leurs utilisations. Il reste à dire que nous privilégions dans la suite l'utilisation des quantificateurs scalaires uniformes pour leur simplicité de mise en œuvre. Employés conjointement avec une transformée performante et un codage entropique ces quantificateurs sont presque aussi performants que les quantificateurs vectoriels.

2.2.3 Allocation optimale

Nous considérons ici des décompositions multi-résolutions par bancs de filtres, suivies de quantificateurs scalaires uniformes, avec un quantificateur scalaire par sous-bande. Le problème de base de l'allocation optimale est le suivant : obtenir, pour un débit donné, une erreur de reconstruction d'énergie minimale, ou inversement, minimiser le débit pour une distorsion donnée [77]. Avant d'aborder la méthode d'allocation optimale nous présentons les formules asymptotiques du débit et de la distorsion en fonction du pas de quantification sous l'hypothèse de quantifications à hautes résolutions.

Distorsion asymptotique

Dans ce cas, l'erreur quadratique moyenne D_{sortie} entre le signal d'origine et le signal décodé est égale à la somme pondérée des erreurs quadratiques moyennes de quantification calculées dans chaque sous-bande D_k (*i.e.*, pour chaque quantificateur). Nous avons alors [111, 48]

$$D_{sortie} \approx \sum_{k=1}^{N_{sb}} a_k w_k D_k \quad (2.4)$$

où N_{sb} est le nombre de sous-bandes. Les paramètres a_k apparaissent dans cette relation en raison des tailles différentes des sous-bandes (a_k est le rapport de la taille de la sous-bande k sur la taille du signal, ainsi $\sum_{k=1}^{N_{sb}} a_k = 1$) et les $(w_k)_k$ sont des coefficients de pondération qui dépendent de la transformation. Dans le cas d'une décomposition orthogonale (ou quasi orthogonale comme la décomposition en ondelettes (9,7) de Daubechies), la relation (2.4) est encore valable pour de faibles débits. De plus, quand la décomposition est orthogonale, $w_k = 1$ pour toutes les sous-bandes [111, 112, 123].

Sous l'hypothèse haute résolution, un quantificateur scalaire uniforme de pas de quantification q , admet une erreur quadratique moyenne D qui peut être approximée par [48]

$$D \approx \frac{q^2}{12}. \quad (2.5)$$

Débit asymptotique en fonction du pas de quantification

Pour un quantificateur scalaire appliqué à une source sans mémoire, le débit entropique des valeurs quantifiées vaut :

$$R = - \sum_n Pr(n) \log_2 Pr(n), \quad (2.6)$$

où $Pr(n)$ est la probabilité que la variable X soit dans l'intervalle de quantification P_n . Sous l'hypothèse haute résolution, pour un quantificateur uniforme, l'entropie d'ordre 1 du signal quantifié peut être approximée par la relation suivante [48] :

$$R \approx h(X) - \frac{1}{2} \log_2 12D \quad (2.7)$$

(connue sous le nom de formule de Bennett) où $h(X)$ est l'entropie différentielle du signal à quantifier. L'équation (2.7), s'écrit aussi :

$$D \approx \frac{1}{12} 2^{2h(X)} 2^{-2R}. \quad (2.8)$$

ou encore, en introduisant la variance σ^2 de X et l'entropie différentielle $h(\tilde{X})$ de la variable réduite $\tilde{X} = \frac{X-E[X]}{\sigma}$:

$$D \approx c \sigma^2 2^{-2R} \quad (2.9)$$

avec $c = \frac{2^{2h(\tilde{X})}}{12}$.

2.2.3.1 Allocation de débit

Le débit moyen total, en bits par échantillon du signal, est égal à la somme pondérée suivante :

$$R_{sortie} = \sum_{k=1}^{N_{sb}} a_k R_k, \quad (2.10)$$

où R_k représente le débit moyen en bits par coefficient de la sous-bande k .

Dans nos tests, nous étudierons le codage quasi sans perte, sous l'hypothèse de quantification à haute résolution. Supposons que la relation (2.4) soit valide ainsi que la relation (2.7) pour chaque quantificateur.

Le problème est de répartir les distorsions D_k sous la contrainte

$$D \approx \sum_{k=1}^{N_{sb}} a_k w_k D_k \leq D_{max}$$

pour minimiser

$$R = \sum_{k=1}^{N_{sb}} a_k R_k \approx \sum_{k=1}^{N_{sb}} a_k \left[h(X_k) - \frac{1}{2} \log_2 12D_k \right]$$

ou, de façon équivalente pour maximiser $\prod_{k=1}^{N_{sb}} D_k^{a_k}$. Or l'inégalité de la moyenne permet d'écrire

$$\prod_{k=1}^{N_{sb}} (w_k D_k)^{a_k} \leq \sum_{k=1}^{N_{sb}} a_k w_k D_k$$

avec l'égalité si et seulement si tous les termes $w_k D_k$ sont égaux entre eux². Le débit total sera donc minimal si et seulement si

$$\forall k \quad D_k = \frac{D_{max}}{w_k}. \quad (2.11)$$

Sous l'hypothèse haute résolution, l'allocation optimale est assurée quand la relation (2.11) est satisfaite, de plus

$$D_k \approx \frac{q_k^2}{12}$$

où q_k est le pas de quantification dans la $k^{\text{ème}}$ sous-bande.

Pour éviter de régler le pas de quantification dans chaque sous-bande, l'allocation optimale peut être obtenue en multipliant les coefficients de la sous-bande k par le facteur $\sqrt{w_k}$ avant de quantifier toutes les sous-bandes avec le même pas de quantification $q = \sqrt{12D_{max}}$. Pour décoder, il suffit alors de diviser les valeurs quantifiées de la sous-bande k par $\sqrt{w_k}$ avant d'appliquer la décomposition inverse.

Quand seule l'approximation (2.4) reste valide, les approximations (2.5) et (2.7) ne l'étant plus, l'allocation optimale peut être réalisée par exemple avec l'algorithme de Shoham et Gersho [99].

²En effet, d'après le concavité de la fonction logarithme $\log(\sum a_k X_k) \geq \sum a_k \log X_k$ quand $\sum a_k = 1$, avec égalité si et seulement si $\exists X, \quad \forall k \quad X_k = X$.

2.2.4 Codage entropique

La dernière étape dans un schéma de compression est le codage entropique. C'est une opération parfaitement réversible, c'est-à-dire qu'elle n'introduit aucune distorsion contrairement à la quantification. Il existe deux grandes classes de codeurs entropiques. La première est constituée des codeurs qui n'exploitent pas la redondance entre symboles consécutifs et qui donnent au mieux, comme longueur moyenne du flot de bits, l'entropie d'ordre 1 des données à coder. La deuxième est constituée des codeurs qui exploitent cette redondance pour donner une longueur moyenne qui tend au mieux vers le débit entropique de la source. Nous commençons par présenter des codeurs de la première classe. Après la quantification, les coefficients prennent des valeurs dans un ensemble fini de symboles $\{y_0, y_1, \dots, y_{L-1}\}$. L'idée est de trouver une bijection de l'ensemble $\{y_i : 0 \leq i < L\}$ dans un nouvel ensemble $\{z_i : 0 \leq i < L\}$ de manière à minimiser le nombre moyen de bits par coefficient. Pour établir une telle bijection, les paramètres sont les probabilités de présence $p(y_i)$ de chaque symbole y_i dans l'ensemble des coefficients. Cette bijection transforme les mots de code à longueur fixe en d'autres à longueur variable. Quand les probabilités $p(y_i)$ sont fixes, un codage de Huffman ou arithmétique est utile. En revanche si les probabilités évoluent avec le temps, des méthodes adaptatives, telles que le codage arithmétique adaptatif, sont plus appropriées.

2.2.4.1 Le codage de Huffman

Comme nous l'avons déjà dit, le but du codage entropique est de trouver une bijection $z_i = F(y_i)$ qui minimise la longueur moyenne des mots de code $l(z_i)$ ayant les probabilités d'occurrence $p(y_i)$, $0 \leq i < L$ comme paramètres :

$$E[l(z_i)] = \sum_{i=0}^{L-1} p(y_i) l(z_i). \quad (2.12)$$

De plus, il est exigé que les mots de code $\{z_i\}$ soient uniquement décodables, c'est-à-dire qu'aucun mot du code ne peut être le préfixe d'un autre mot. Rappelons que l'entropie de l'ensemble $\{y_i\}$ est donnée par

$$H = - \sum_{i=0}^{L-1} p(y_i) \log_2 p(y_i), \quad (2.13)$$

et qu'elle correspond, d'après le théorème de Shannon [97], à la limite inférieure de la longueur moyenne des mots de code pour une source sans mémoire. La méthode de construction des mots de code proposée par Huffman satisfait à la condition de préfixe et fournit la longueur moyenne des mots de code la plus proche de l'entropie (à condition que cette longueur moyenne soit supérieure à 1 bit par symbole) [37].

Bien que le codage de Huffman fournisse des mots de code dont la longueur moyenne (2.12) est supérieure au plus d'un bit à l'entropie (2.13), le fait d'avoir un bit de plus peut être inacceptable dans certaines applications, notamment en compression avec pertes d'images où l'entropie n'est qu'une faible fraction de l'unité. Pour contourner ce problème, plusieurs échantillons peuvent être regroupés pour former un nouvel ensemble de symboles dont l'entropie est supérieure à 1 bit. Par

exemple, dans les normes JPEG et MPEG le codage de Huffman est implanté de cette façon avec un codage de type *Run-Length* (longueur de plage) qui le précède pour regrouper des échantillons. Une autre solution consiste à utiliser un codeur arithmétique [118].

2.2.4.2 Le codage arithmétique

Le codage arithmétique attribue un “mot de code” à chaque ensemble de données. Ces “mots de code” correspondent à des intervalles semi-ouverts disjoints inclus dans $[0, 1[$. Les “mots de code” sont construits en spécifiant suffisamment de bits pour distinguer des autres intervalles celui correspondant à l'ensemble actuel de données. Les codes les plus courts sont associés aux intervalles les plus larges, qui représentent les ensembles de données les plus probables. Dans la pratique, l'intervalle est raffiné successivement en utilisant les probabilités d'occurrence $p(y_i)$ des symboles.

Un codage entropique doit être accompagné, aux différents points de codage, d'un modèle d'estimation de probabilité pour chaque symbole possible. Ce modèle de probabilité n'a pas besoin de décrire le processus qui génère les données, il doit tout simplement fournir une distribution de probabilité des données, qui n'est pas nécessairement exacte. Toutefois, le débit sera d'autant plus faible que la distance de Kullback-Leibler entre les distributions de probabilité (réelle et estimée) sera faible [37]. Le modèle peut être adaptatif (estimation dynamique de la probabilité de chaque symbole en fonction des événements qui précèdent), semi-adaptatif (en effectuant un passage préliminaire dans le fichier à coder pour estimer les statistiques) ou non-adaptatif (en utilisant des probabilités fixes pour tous les fichiers). Les modèles non-adaptatifs sont arbitraires et souvent peu performants [122]. Le codage adaptatif est du type “un seul passage” mais a une plus grande complexité. En revanche le codage semi-adaptatif exige un double passage et la transmission du modèle de données comme information supplémentaire. Si la transmission du modèle est bien optimisée il peut fournir une meilleure compression que le codage adaptatif. En général, le coût de transmission du modèle semi-adaptatif est presque le même que celui de l'apprentissage du modèle adaptatif notamment pour des données codées sur 8 à 10 bits [55].

2.2.4.3 Codage universel

Dans ses travaux, Rissanen [87] a introduit la notion de codage universel, qui consiste en l'adaptation des modèles aux données à compresser, dans le but d'atteindre des taux de compression asymptotiquement optimaux. Asymptotique signifie ici, s'approcher le plus possible du débit entropique. Le codage universel s'applique dans des situations contraires à celles des systèmes de compression (par exemple dans les communications numériques) où les propriétés statistiques de la source à coder sont connus à l'avance et où le modèle est alors connu *a priori* par le codeur et le décodeur. Il a également proposé un algorithme de compression asymptotiquement optimal appelé *CONTEXT*, qui vise à compresser une séquence de données par construction adaptative d'un modèle d'arbre. Très complexe et donc non utilisable en pratique, cet algorithme peut néanmoins servir de référence pour d'autres systèmes de compression privilégiant une complexité réduite au dépend de l'efficacité

de codage. Ces méthodes de codage sont connues sous le nom de méthode de codage universel.

Un autre exemple plus connu de codeur universel est l'algorithme LZW de Lempel-Ziv [130, 131] qui exploite l'apparition de motifs répétitifs dans une séquence. Le codage universel s'est avéré très efficace pour la compression de textes [23]. Malheureusement, les méthodes de codage universel ne fonctionnent pas bien quand elle sont appliquées aux images en niveaux de gris pour les raisons suivantes.

- Les images sont caractérisées par des zones de textures séparées par des contours. Le prix à payer pour la modélisation de ces caractéristiques en utilisant un schéma de codage universel s'avère très élevé [119].
- La majorité des schémas de codage universel traitent les données comme une suite mono-dimensionnelle de symboles à coder, mais les images ont une structure intrinsèquement bi-dimensionnelle.
- La taille de l'alphabet dans le cas des images est très importante, typiquement 256 lettres ou plus. Par conséquent, la valeur d'un pixel sera conditionnée sur un nombre très élevé de contexte et quand le nombre est grand par rapport à la taille de l'image, il n'y a pas assez d'échantillons pour estimer correctement les probabilités conditionnelles, conduisant ainsi à de mauvaises performances de codage. Ce problème est connu sous le nom de *context dilution* [119] ou *sparse context* [64]. Il a été abordé par Rissanen dans le cadre théorique de la complexité stochastique en tant que "coût du modèle" [88]. D'une manière intuitive, le "coût du modèle" consiste soit en la quantité d'information nécessaire à la description du modèle quand celui-ci est envoyé entièrement au décodeur, soit à la diminution d'efficacité de codage liée à la mise à jour du modèle au fur et à mesure du codage (et du décodage) des symboles. Dans la littérature relative à la théorie de l'information, ce principe porte le nom de *modélisation universelle de source* [88]. Par exemple, le codeur d'images CALIC a été conçu dans ce cadre théorique.
- Les images acquises par l'intermédiaire de capteurs sont généralement bruitées et c'est pourquoi les schémas de codage universel qui exploitent la répétition fréquente de motifs ne fonctionnent pas bien dans le cas des images.

2.3 Codage progressif

Une fonctionnalité importante en compression d'images est le codage progressif (en qualité ou en résolution). Elle donne à l'utilisateur un contrôle total de la précision avec laquelle l'image est représentée et permet la reconstruction d'une approximation de l'image originale à partir d'une version tronquée du flot de bits. Au fur et à mesure que les données sont disponibles, la qualité de l'image reconstruite, ou sa résolution, s'améliore jusqu'à atteindre, éventuellement, celle de l'image originale.

En 1992, Antonini *et al.* [14] ont proposé un système de codage d'image fixe basé sur la transformation en ondelettes et la quantification vectorielle qui permet un codage progressif en résolution. Un an plus tard, Shapiro [98] a introduit le codage progressif en qualité (*embedded coding*). Son schéma de codage, appelé *Embedded Zerotree Wavelet (EZW) coding* est très rapidement devenu populaire pour ses

multiples avantages, notamment parce qu'il a permis la construction des systèmes unifiés de compression avec et sans pertes. En 1995, Zandi *et al.* [128] ont proposé le système CREW (*Compression with Reversible Embedded Wavelets*) de compression d'images, qui permet le codage progressif en qualité et qui n'est pas très différent du EZW de Shapiro. Un an après, Said et Pearlman [91] ont proposé le système SPIHT (*Set Partitioning In Hierarchical Trees*) similaire à EZW, mais plus performant. Les décompositions multi-résolutions permettent les deux types de progressivité (en résolution et en qualité), mais l'étape de codage qui vient après la décomposition dépend du type de progressivité voulu : les approches par arbres de zéros n'offrent plus, généralement, la progressivité en résolution, du fait des relations inter-bandes qu'elles exploitent. Le nouveau standard JPEG 2000 [93] emploie des techniques issues de l'état de l'art en compression et basées sur les décompositions en ondelettes. Il offre une grande souplesse à l'utilisation et permet en particulier le codage progressif en qualité et en résolution.

2.3.1 Progressivité en résolution

L'objectif principal du codage progressif en résolution est de permettre au récepteur de reconnaître l'image rapidement et à bas coût. Pour cela les bits du train binaire sont ordonnés pour qu'en premier apparaisse l'information des images des sous-bandes de plus basses résolutions (cf. fig. 2.2). L'utilisateur décidera ensuite



FIG. 2.2 – Illustration de la progressivité en résolution.

d'arrêter la transmission, ou de recevoir des détails pour reconstruire l'image à une plus grande résolution, pouvant aller jusqu'à celle de l'image originale. Pour une image de taille $N \times M$, avec N et M divisibles par 2^m , sa taille à la résolution $-m$ est définie par la relation

$$N_m = \frac{N}{2^m} \text{ et } M_m = \frac{M}{2^m}. \quad (2.14)$$

L'image originale correspond à la résolution $m = 0$.

2.3.2 Progressivité en qualité

Contrairement à la progressivité en résolution, l'image est toujours reconstruite à plein résolution, c'est-à-dire à la résolution $m = 0$. C'est la qualité de l'image reconstruite qui s'améliore au fur et à mesure que les données arrivent, jusqu'à atteindre

une qualité suffisante. Ces techniques trouvent par exemple leur utilité dans la transmission sur un réseau limité en bande passante (ou à bande passante variable) et dans le contrôle du nombre de bits avec lequel l'image veut être stockée. Un exemple de progressivité en qualité est illustré à la figure 2.3.



FIG. 2.3 – Illustration de la progressivité en qualité.

2.4 Compression sans perte

2.4.1 Schéma général

L'objectif de la compression sans perte d'une source est d'assigner un code binaire à chaque séquence d'échantillons (constituant par exemple une image) de sorte que le nombre de bits total nécessaire pour coder la séquence soit le plus petit possible.

Si la séquence est constituée de n symboles (x_1, \dots, x_n) le théorème de Shannon donne comme longueur de code moyenne minimale l'entropie jointe $H(x_1, \dots, x_n)$. Le problème réside alors dans l'estimation de la probabilité jointe

$$p(x_1, \dots, x_n) = \prod_{i=0}^{n-1} p(x_{i+1}|x_i, \dots, x_1) \quad (2.15)$$

où $p(x_{i+1}|x_i, \dots, x_1)$ est la probabilité conditionnelle de x_{i+1} sachant x_i, x_{i-1}, \dots, x_1 avec pour convention $p(x_1|x_0) = p(x_1)$. Dans le cas général, même pour une source stationnaire, appliquer un estimateur non paramétrique de distribution de probabilités jointes ou conditionnelles pour obtenir la distribution (2.15) donne de mauvaises estimations à cause du manque de données. Pour contourner ce problème il existe deux approches qui sont souvent couplées. La première consiste à appliquer des estimateurs paramétriques de distributions de probabilités jointes ou plus souvent conditionnelles : l'ensemble des valeurs possibles du passé $(x_i, x_{i-1}, \dots, x_1)$ est partitionné en quelques sous-ensembles bien choisis (c'est la modélisation de contexte) réduisant ainsi de façon drastique le nombre de cas possibles. La seconde cherche à réduire les dépendances longues (c'est-à-dire celles existantes entre des pixels éloignés) de la source au moyen d'une transformation. Il s'agit en général d'une prédiction de x_{i+1} en fonction du passé

$$\hat{x}_{i+1} = P(x_i, \dots, x_1)$$

et le codage porte sur la séquence $(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$ des résidus de prédiction $\tilde{x}_{i+1} = x_{i+1} - \hat{x}_{i+1}$ pour laquelle les distributions de probabilités conditionnelles sont approximativement indépendantes du passé lointain :

$$p(\tilde{x}_{i+1}|\tilde{x}_i, \dots, \tilde{x}_1) \approx p(\tilde{x}_{i+1}|\tilde{x}_i, \dots, \tilde{x}_{i-j})$$

avec $j \ll i$ et qui peuvent être estimées quand la source est stationnaire (ou localement stationnaire).

La plupart des systèmes de compression sans perte des images comportent trois étapes : prédiction, modélisation de contexte et codage entropique [121, 73]. Un schéma est donné à la figure 2.4. La prédiction tend à réduire la redondance entre pixels voisins en exploitant la corrélation spatiale qui existe. Le coût que nécessite le modèle du contexte est ainsi considérablement réduit. Comme la décorrélation complète ne peut pas être atteinte par la prédiction seule, la modélisation de contexte est introduite pour exploiter la redondance restante. Les étapes de prédiction et de modélisation de contexte permettent ainsi une meilleure estimation des distributions des probabilités conditionnelles apparaissant dans (2.15). Enfin, un codeur entropique peut exploiter ces distributions de probabilités pour donner une longueur moyenne de code approximativement égale au débit entropique ou à l'entropie d'ordre 2, 3 ou plus de la source.

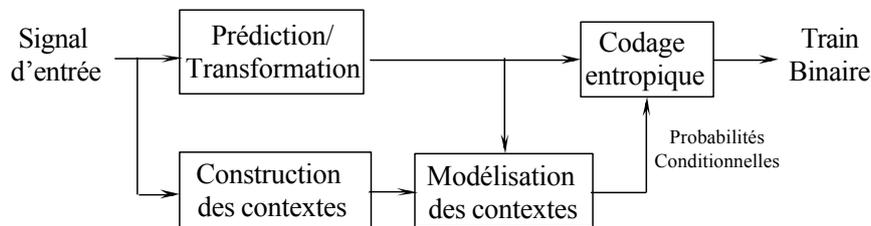


FIG. 2.4 – Schéma général d'un codeur sans perte.

Au paragraphe suivant nous recensons différents systèmes de prédiction rencontrés en compression sans perte. L'étape de prédiction peut être intégrée dans une structure en *lifting scheme*, permettant ainsi un codage progressif en résolution (et éventuellement en qualité allant jusqu'au sans perte), ou non, auquel cas le codeur ne permet pas la progressivité en résolution.

Nous pouvons supposer sans perte de généralité que les symboles sont des nombres entiers ($\in \mathbb{Z}$) et l'étape de prédiction se découpe en deux parties : la première est la prédiction proprement dite à valeurs dans l'ensemble des nombres réels et la seconde un arrondi à l'entier le plus proche.

2.4.2 Prédiction

L'étape de prédiction dépend du modèle appliqué au signal et donc des hypothèses faites sur ce dernier.

Nous ne considérons que des signaux aléatoires. Leurs propriétés statistiques sont complètement décrites par leurs lois temporelles. Nous distinguerons, comme cela est fait classiquement, différentes stationnarités : au sens strict (SSS) ou une plus faible, la stationnarité au sens large (*Wide Sens Stationarity WSS*). Dans ce cas, une connaissance *a priori*, ou après un processus d'apprentissage, des statistiques du signal permet de fixer les paramètres du modèle ou du prédicteur, qui ne varient pas au cours du temps. En revanche, quand le signal n'est pas stationnaire, mais localement stationnaire ou aux propriétés statistiques lentement variables dans le

temps, la prédiction doit être adaptative pour suivre les variations des statistiques du signal.

Quand le prédicteur est linéaire, il suffit de connaître les moments d'ordre 1 et 2 du signal pour déterminer la prédiction affine qui minimise l'erreur quadratique moyenne du résidu de prédiction et quand le signal est stationnaire au sens large, le filtre optimal (c'est-à-dire celui qui minimise l'erreur quadratique moyenne du résidu de prédiction) est celui de Wiener, il s'obtient en résolvant les équations de Yule-Walker pour une prédiction à passé fini [15]. Quand le signal n'est pas stationnaire au sens large, mais aux statistiques lentement variables, il existe de nombreuses méthodes récursives dans le temps qui adaptent les paramètres d'un prédicteur linéaire aux variations des statistiques du signal. Citons l'algorithme des moindres carrés adaptatifs, le filtre de Kalman ou encore l'algorithme du gradient stochastique.

Les prédicteurs linéaires se rencontrent le plus souvent, car se sont les plus simples à mettre en œuvre, les mieux connus et ils sont en général peu sensibles aux erreurs d'arrondis inévitables quand ils sont implantés dans un ordinateur. Toutefois, ils peuvent être peu efficaces dans certaines situations et il est alors préférable d'utiliser des prédicteurs non linéaires comme ceux à base de réseaux de neurones.

Une autre famille de modèles repose sur un regroupement par blocs d'échantillons consécutifs, donnant ainsi une représentation vectorielle du signal, qui est modélisée par un mélange instantané, ou non, de sources indépendantes ou décorrélées. Des algorithmes basés sur la connaissance des moments d'ordre 1 et 2 (analyse en composantes principales³) ou d'ordres supérieurs (analyse en composantes indépendantes) [80, 36, 29] permettent de déterminer les paramètres de tels modèles qui collent le mieux aux données. Ces modèles sont utilisés en codage : les "sources" estimées ont moins de redondance entre elles que le signal initial et peuvent être codées séparément.

Un autre facteur important est la rapidité avec laquelle le filtre peut suivre les changements des statistiques des signaux non stationnaires. Comme le codage prédictif dépend fortement de la précision, l'aptitude du prédicteur à suivre les changements du signal est une propriété cruciale.

2.4.2.1 Prédiction linéaire

Filtre de Wiener Pour les signaux stationnaires au sens large dont les moments d'ordre 1 et 2 sont connus, le filtre de Wiener est celui qui minimise l'erreur quadratique moyenne du résidu de prédiction

$$e(n) = d(n) - (w \star x)(n),$$

où $d(n)$ est la sortie désirée, w est la réponse impulsionnelle du filtre et $x(n)$ le signal observé. Le filtre optimal est solution des équations de Wiener-Hopf [15]. Nous supposons sans perte de généralité que les signaux sont centrés. Quand on impose au filtre d'être causal et RIF d'ordre N , on peut introduire le vecteur observation $\underline{x}(n) = [x(n), x(n-1), \dots, x(n-N)]^T$ et $\underline{w} = [w_0, \dots, w_N]^T$ le vecteur des coefficients du filtre, de sorte que

$$e(n) = d(n) - \underline{w}^T \underline{x}(n)$$

³Appelée transformation Karhunen-Loève dans les cours de codage

et le filtre optimal est solution des équations de Yule-Walker

$$\mathbf{R}_{xx}\underline{w} = \underline{r}_{dx} \quad (2.16)$$

où \underline{r}_{dx} est le vecteur d'intercorrélation entre la valeur recherchée $d(n)$ et le vecteur observation $\underline{x}(n)$ et \mathbf{R}_{xx} la matrice d'autocorrélation de ce dernier.

Méthodes récursives dans le temps Quand les moments d'ordre 2 des signaux sont inconnus, la méthode des moindres carrés récursifs permet de les estimer et de résoudre l'équation (2.16) (dans laquelle tous les termes dépendent de n maintenant) sans inverser la matrice \mathbf{R}_{xx} . Quand les signaux ne sont pas WSS, mais presque stationnaires (lentement variables) l'algorithme des moindres carrés adaptatifs permet de résoudre l'équation (2.16) sans inverser de matrice et de suivre les variations lentes des statistiques.

Algorithme du gradient stochastique Une alternative aux méthodes récursives dans le temps est l'algorithme du gradient stochastique (*Least Mean Square LMS*). Les coefficients sont mis à jour suivant l'équation

$$\underline{w}(n+1) = \underline{w}(n) + \mu \left(-\frac{\partial[e^2(n)]}{\partial \underline{w}(n)} \right) = \underline{w}(n) + \mu \underline{x}(n)e(n) \quad (2.17)$$

où μ est un pas d'adaptation. L'algorithme du gradient stochastique est largement utilisé en raison de sa complexité réduite et de sa facilité d'implantation, car il ne demande pas d'estimations de fonctions de corrélation. La vitesse de convergence est généralement lente.

L'algorithme du gradient stochastique peut aussi être implanté récursivement, bénéficiant des avantages des filtres à réponse impulsionnelle infinie (RII). Plus de détails sur les filtres RII adaptatifs peuvent être trouvés dans [100]. En raison de la présence de pôles dans la fonction de transfert du filtre, sa stabilité doit être testée. Il faut également prendre en considération l'ordre du filtre, qui, judicieusement choisi, permet l'élimination des minima locaux [101, 13].

L'algorithme du gradient stochastique a été appliqué au codage sans perte des images dans une structure prédictive en pyramide [84]. Pour chaque résolution, des blocs de données de la résolution précédente sont utilisés comme entrées du prédicteur pour estimer les pixels de la résolution qui suit. Afin d'éviter les effets de blocs, ces derniers se chevauchent dans la procédure de mise à jour. Les prédicteurs adaptatifs ont donné des résultats de compression, mesurés par l'entropie du premier ordre, supérieurs à ceux d'autres techniques pyramidales aux dépens du coût de calcul plus élevé. Aussi, dans une structure de décomposition multi-résolution, l'algorithme du gradient stochastique est utilisé par Oktem *et al.* dans [82].

La prédiction adaptative basée sur un filtrage récursif améliore les performances de compression sans perte des images, cela peut s'observer en comparant le prédicteur basé sur un modèle ARMA à celui basé sur un modèle MA [33, 81].

Collection de filtres Une autre méthode de prédiction adaptative consiste en la sélection du prédicteur dans un ensemble de filtres préalablement défini. La méthode

de sélection peut utiliser un modèle, c'est le cas de plusieurs systèmes de compression sans perte tels que LOCO-I [120] et CALIC [124]. Le choix du prédicteur est fait en fonction de la direction du contour dominant le plus proche du pixel à estimer. Nous détaillerons plus loin ces deux systèmes. Une autre approche, simple, est décrite dans [95], la sélection du prédicteur est faite en fonction de l'amplitude de l'erreur de prédiction.

Prédiction linéaire basée sur des moments d'ordres supérieurs à 2 Les cumulants d'ordres supérieurs peuvent être très utiles dans des applications impliquant des signaux non-gaussiens. Comme les images d'erreur de prédiction linéaire sont généralement bien modélisées par une distribution laplacienne [59], les images naturelles peuvent être considérées comme non gaussiennes [107]. La prédiction linéaire basée sur des cumulants a été décrite dans [102]. Des filtres de type ARMA mono-dimensionnels, basés sur des statistiques d'ordres supérieurs, ont été introduits dans [30] pour de la prédiction. Ils sont une extension des filtres MA proposés dans [31]. Dans [52], les prédicteurs linéaires de type ARMA basés sur des cumulants d'ordres 2 et 3 ont permis d'obtenir des images plus nettes à l'œil nu, grâce à l'exploitation des statistiques d'ordre 3. Cependant, les auteurs ont remarqué que ces filtres ne sont pas optimaux au sens de l'erreur quadratique moyenne. Une autre illustration de cette sous-optimalité en compression de données peut être trouvée dans [79].

Les prédicteurs linéaires sont parfois inadéquats, notamment quand le signal à coder est non gaussien et lorsque la dépendance entre les échantillons est non linéaire. Nous présentons maintenant des prédicteurs non linéaires.

2.4.2.2 Réseaux de neurones

Les réseaux de neurones se sont avérés très efficaces pour l'étape de prédiction en compression sans perte. Dans [54], un perceptron multi-couche (PMC) utilisant l'apprentissage stochastique (en-ligne) est comparé favorablement au prédicteur JPEG sans perte. Dans [72], les auteurs utilisent un réseau de Volterra pour la prédiction mono-dimensionnelle et bi-dimensionnelle des images. Il a été démontré que pour l'étape de prédiction les réseaux de neurones offrent une satisfaction équivalente sinon meilleure que les modèles prédictifs linéaires. Dans [62], les auteurs utilisent encore des PMC pour la prédiction d'images multi-spectrales avec un taux d'apprentissage variable.

2.4.2.3 *Lifting scheme* réversible

Nous avons vu au chapitre précédent la structure en *lifting scheme* d'un banc de filtres biorthogonal et nous avons rappelé le théorème de Sweldens et Daubechies qui affirme que toute décomposition en ondelettes associée à des filtres RIF (pour les filtres d'analyse et de synthèse) admet une représentation en *lifting scheme*.

Quand les données d'entrée prennent des valeurs entières (dans \mathbb{Z}) il suffit d'ajouter un arrondi à l'entier le plus proche dans les branches verticales de la structure en *lifting scheme*, après chaque filtre prédicteur P ou de mise à jour U , pour que

les données restent entières dans les branches horizontales de la structure et par conséquent en sortie.

Ces dernières années, il y a eu un intérêt croissant pour des décompositions en ondelettes réversibles, *i.e.* transformant des entiers en entiers, pour les applications de codage d'images [9]. Elles approximent les décompositions en ondelettes linéaires, tout en ayant la propriété d'être réversibles, permettant ainsi un codage progressif en qualité pouvant aller jusqu'au sans perte. Elles sont, de ce fait, largement utilisées dans les systèmes de compression. Parmi elles, la transformation (5,3) [28] est utilisée en compression sans perte (ou progressive allant jusqu'au sans perte) du standard JPEG 2000.

Les performances en compression avec et sans perte de plusieurs transformations en ondelettes réversibles ont été comparées dans [7, 9]. Le contenu de l'image s'avère un facteur important ayant une influence sur l'efficacité de la transformation.

Parmi toutes les transformations réversibles considérées, la (5,3) et la (2,6) (dont les équations sont données en annexe) présentent les plus basses complexités. Tandis que la (9,7), efficace en compression avec pertes, à une complexité significativement plus grande que les autres transformations. Il faut noter également qu'aucune de ces transformations ne présente de meilleures performances à la fois en compressions avec et sans pertes. Ce qui nous conduit à la conclusion que le choix de la transformation la plus appropriée dépend des besoins de l'utilisateur. Par exemple, dans le cas des applications exigeant la basse complexité algorithmique, la transformation (5,3) est particulièrement intéressante. Elle est raisonnablement bien adaptée pour les compressions avec et sans pertes.

La transformation S+P [92] décrite dans le prochain paragraphe a été présentée comme une version améliorée des transformations en ondelettes réversibles.

La transformation S+P. Initialement proposée par Said et Pearlman [92], la transformation S+P est une transformation d'entiers en entiers. C'est une amélioration de la transformation S (voir annexe) où le signal de détail est ajusté par une étape de prédiction en utilisant l'un des prédicteurs (A, B ou C) proposés par les auteurs. La transformation directe d'un signal $x(n)$ est donnée par

$$x_h^{(1)}(n) = x(2n+1) - x(2n) \quad (2.18)$$

$$x_\ell(n) = x(2n) + \left\lfloor \frac{x_h^{(1)}(n)}{2} + \frac{1}{2} \right\rfloor \quad (2.19)$$

$$x_h(n) = x_h^{(1)}(n) + \lfloor \alpha_{-1} (x_\ell(n-2) - x_\ell(n-1)) + \alpha_0 (x_\ell(n-1) - x_\ell(n)) + \alpha_1 (x_\ell(n) - x_\ell(n+1)) - \beta_1 x_h^{(1)}(n+1) \rfloor. \quad (2.20)$$

La transformation inverse se déduit immédiatement de la transformation directe par

$$x_h^{(1)}(n) = x_h(n) - \lfloor \alpha_{-1} (x_\ell(n-2) - x_\ell(n-1)) + \alpha_0 (x_\ell(n-1) - x_\ell(n)) + \alpha_1 (x_\ell(n) - x_\ell(n+1)) - \beta_1 x_h^{(1)}(n+1) \rfloor \quad (2.21)$$

$$x(2n) = x_\ell(n) - \left\lfloor \frac{x_h^{(1)}(n)}{2} + \frac{1}{2} \right\rfloor \quad (2.22)$$

$$x(2n+1) = x_h^{(1)}(n) + x(2n). \quad (2.23)$$

Selon les valeurs de α_{-1} , α_0 , α_1 et β_1 , trois prédicteurs sont proposés.

Prédicteur/Coef.	α_{-1}	α_0	α_1	β_1
A	0	1/4	1/4	0
B	0	1/4	3/8	1/4
C	-1/16	4/16	8/16	6/16

La transformation S+P avec le prédicteur A est aussi appelée la transformation TS ([28]).

Depuis son introduction, la transformation S+P est devenue très populaire en compression sans perte. Le prédicteur B est bien adapté aux images naturelles et le prédicteur C aux images lisses telles que les images médicales IRM.

Said et Pearlman ont associé la transformation S+P à un codeur entropique, qui applique un codage arithmétique (ou de Huffman) adaptatif aux valeurs des coefficients de l'image décomposée en la balayant, sous-bande par sous-bande. Cela donne un codeur sans perte progressif en résolution. Ils ont également associé la transformation S+P au codeur entropique SPIHT pour donner un codeur sans perte permettant un codage progressif en qualité.

2.5 Principaux systèmes de compression sans perte

Ces dernières années, plusieurs nouveaux schémas de compression sans perte ont été proposés [57, 120, 124, 91, 92, 128, 93]. Cela montre, non seulement qu'il y a beaucoup d'intérêt pour le développement et l'application des schémas de compression sans perte, mais aussi que des systèmes complètement satisfaisants n'ont pas encore été trouvés. L'objet de la présente section est de donner une idée des différentes stratégies de codage sans perte, en insistant sur la transformation (prédiction) utilisée. Le pixel courant est noté P et sa prédiction \hat{P} . La figure 2.5 précise les notations des pixels proches de P situés dans un voisinage semi-causal.

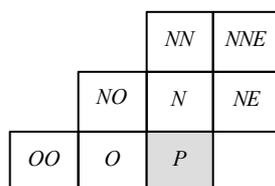


FIG. 2.5 – Notations utilisées pour spécifier la position des voisins du pixel P (“nord”, “ouest”,...).

2.5.1 LOCO-I

LOCO-I (*Low Complexity, Context-Based, Lossless Image Compression Algorithm*) a été développé par Weinberger *et al.* [120]. Il est considéré par le comité de l'ISO comme un remplacement du standard LJPEG (*Lossless JPEG*) [118], pour les applications nécessitant une complexité minimale. Il est principalement basé sur

la prédiction de chaque pixel de l'image à partir de son voisinage. La prédiction \hat{P} de P est donnée par

Si $NO \geq \max(N, O)$
 $\hat{P} = \min(N, O)$
sinon Si $NO \leq \min(N, O)$
 $\hat{P} = \max(N, O)$
sinon
 $\hat{P} = N + O - NO$
Fin

À partir des valeurs de cinq plus proches voisins du pixel courant, quatre gradients locaux sont calculés et quantifiés en un nombre limité de contextes. La distribution de probabilité de l'erreur de prédiction conditionnée par le contexte est approximée par une distribution de Laplace, et un codeur de Golomb-Rice est utilisé.

2.5.2 CALIC

CALIC (*Context-based Adaptive Lossless Image Codec*) [124] est le plus efficace et le plus complexe des systèmes de compression sans perte. Malgré la complexité de l'algorithme, la quantité de mémoire utilisée correspond à la taille de deux lignes en plus de 3,3 K-octets pour la modélisation.

Dans une première étape, un prédicteur, appelé GAP (*Gradient-Adjusted Predictor*), est appliqué pour estimer la valeur de chaque pixel de l'image à partir de son voisinage causal. Le fait que les erreurs de prédiction ne soient pas complètement décorrélées conduit à une seconde étape de prédiction en fonction du contexte. De plus, le codec peut basculer automatiquement vers un mode binaire, dédié aux régions où les pixels ne prennent que deux valeurs possibles. Une description schématique du système de codage est donnée à la figure 2.6

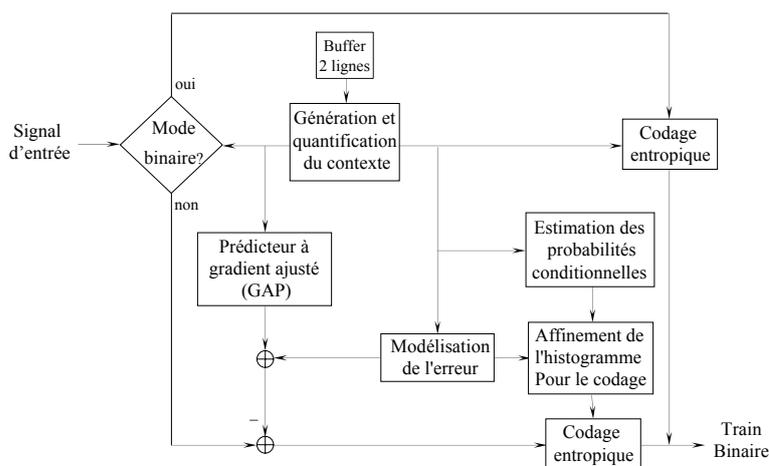


FIG. 2.6 – Description schématique du codeur CALIC.

En utilisant un voisinage du pixel à estimer (voir figure 2.5), les gradients dans les directions horizontale, verticale et diagonale sont calculés. Ces derniers sont utilisés

pour choisir un prédicteur non-linéaire. Soient d_h et d_v les gradients d'amplitude dans les directions horizontale et verticale

$$d_h = |O - OO| + |N - NO| + |N - E|$$

et

$$d_v = |O - NO| + |N - NN| + |NE - NNE|.$$

Ces gradients détectent l'orientation et l'amplitude des contours et déterminent les filtres prédicteurs selon la procédure suivante.

Si ($d_h - d_v > 80$) contour horizontal net

$$\hat{P} = O$$

sinon Si ($d_h - d_v < -80$) contour vertical net

$$\hat{P} = N$$

sinon

$$P_1 = (N + O)/2 + (NE + NO)/4$$

Si ($d_h - d_v > 32$) contour horizontal

$$\hat{P} = (P_1 + O)/2$$

sinon Si ($d_h - d_v > 8$) contour horizontal peu prononcé

$$\hat{P} = (3P_1 + N)/2$$

sinon Si ($d_h - d_v < -32$) contour vertical

$$\hat{P} = (P_1 + N)/2$$

sinon Si ($d_h - d_v < -8$) contour vertical peu prononcé

$$\hat{P} = (3P_1 + N)/2$$

Fin

Fin

2.5.3 JPEG 2000

Le nouveau standard de compression d'images JPEG 2000⁴ [93] couvre une large gamme d'applications allant des appareils photographiques numériques portatifs jusqu'aux systèmes d'imagerie médicale. Il se compose de plusieurs parties dont les six premières sont les suivantes.

- Partie 1. Le système de codage d'image (le noyau du standard), fixé depuis fin 2000 pour le codage avec perte et plus récemment pour le codage sans perte.
- Partie 2. Extensions (ajouts de fonctionnalités au noyau).
- Partie 3. Partie dédiée aux séquences d'images.
- Partie 4. Test de conformité.
- Partie 5. Logiciel de références (actuellement, une implantation Java (JJ2000⁵) et une autre en C (JASPER⁶) sont disponibles).
- Partie 6. Images composées (magazines, fax, ...).

⁴Plus de détails sont disponibles à l'adresse <http://www.jpeg.org/JPEG2000.htm>

⁵JJ2000, une implantation du standard JPEG 2000 en Java, par EPFL, ERICSSON et le centre de recherche français de Canon, disponible à l'adresse <http://jj2000.epfl.ch/>

⁶JASPER, une implantation du standard JPEG 2000 en C, par le Laboratoire de traitement d'images UBC (Vancouver), disponible à l'adresse <http://www.ece.ubc.ca/~mdadams/jasper/>

Dans cette norme, la compression sans perte est introduite dans la partie 1 [32]. Elle est basée sur la version réversible (*lifting*) de la transformation en ondelette (5,3). Cette dernière a été considérée comme un bon compromis entre l'efficacité de codage et la complexité [94]. Les équations décrivant cette transformation sont données en annexe C.

2.5.4 Autres systèmes à base de *lifting schemes adaptés*

Nous avons vu à la fin du chapitre précédent différents *lifting schemes* adaptés, intégrés dans des codeurs d'images. Le codeur entropique utilisé est en général arithmétique adaptatif avec parfois une modélisation de contexte.

2.6 Conclusion

Dans ce chapitre, nous avons présenté des éléments de la théorie du codage utiles à la suite de notre présentation. Nous avons vu qu'une décomposition hiérarchique pyramidale permet un codage progressif en résolution et peut donner lieu également à un codage progressif en qualité.

Dans le chapitre suivant nous présentons la structure en *lifting scheme* généralisé que nous avons étudiée, afin de connaître les limites de performance du codage par prédiction linéaire associé à une décomposition hiérarchique pyramidale, quand toute l'information disponible au décodeur est exploitée.

CHAPITRE 3

Synthèse de bancs filtres adaptés proposée

3.1 Introduction

À la fin du chapitre 1, nous avons vu que la structure en *lifting scheme* adaptée étudiée par Gerek et Çetin n'exploite pas toute l'information disponible pour prédire la valeur du pixel courant. Dans notre démarche nous avons choisi d'utiliser toute l'information disponible pour prédire la valeur du pixel courant, afin de connaître les limites de performance du codage par prédiction linéaire associé à des décompositions hiérarchiques pyramidales. Pour adapter les filtres nous avons choisi les méthodes des moindres carrés (pour un signal stationnaire au sens large) ou des moindres carrés adaptatifs (si la stationnarité n'est que locale). Pour cela nous avons introduit une structure en *lifting scheme* généralisé qui permet un codage sans perte et progressif en résolution.

Dans ce chapitre nous commençons par justifier le choix du critère d'adaptation que nous avons retenu (minimisation de la variance) pour ajuster la structure au signal à coder. Nous précisons en particulier les hypothèses faites sur le signal à coder pour que le filtre prédictif soit optimal en codage sans perte. Puis nous décrivons la structure en *lifting scheme* généralisé que nous avons introduite. Nous présentons deux variantes en fonction de la stationnarité du signal à coder : globale ou locale. Enfin nous montrons les performances de ces variantes (en termes de variance du résidu de prédiction et d'entropie d'ordre 1 des coefficients transformés) en les comparant à celles des décompositions en ondelettes (5,3) et (9,7) de Daubechies et à la décomposition adaptée de Gerek et Çetin.

3.2 Justification du critère de minimisation de la variance

Nous avons vu au chapitre 1 qu'en codage avec pertes, pour des décompositions par banc de filtres orthogonales appliquées à des signaux gaussiens, le gain de codage est maximum quand le produit des variances des signaux de sous-bandes est minimal, et que l'optimalité n'est plus atteinte dès qu'une (au moins) de ces hypothèses (orthogonalité/gaussianité) n'est pas satisfaite. De plus, nous avons vu qu'en codage sans perte, plusieurs systèmes de compression utilisent comme critère d'adaptation du filtre prédictif la minimisation de la variance du résidu de prédiction. Dans cette section nous proposons une justification de ce critère. Ce n'est pas une démonstration mathématique. Elle est basée sur le principe qu'à haute résolution, une quantification scalaire donne une valeur quantifiée X^Q indépendante du bruit de quantification $X - X^Q$. Seule la décorrélation peut être démontrée : $E[X^Q(X - X^Q)] = 0$, mais nous aurons besoin de l'indépendance. Nous commençons par montrer que les opérations de quantification et de transformation peuvent commuter dans le sens où les signaux obtenus ont la même loi temporelle (sous certaines conditions).

Considérons un vecteur aléatoire \underline{X} , gaussien de moyenne \underline{m} et de matrice variance-covariance \mathbf{K}_x , que nous noterons $\mathcal{N}(\underline{m}, \mathbf{K}_x)$, quantifié à haute résolution avec un quantificateur scalaire uniforme de pas de quantification q , identique pour toutes les composantes. La version quantifiée résultante, notée \underline{X}^Q , est obtenue en utilisant le modèle de bruit additif de l'opération de quantification :

$$\underline{X}^Q = \underline{X} + q\underline{U} \quad (3.1)$$

où $\underline{U} = [U_1, \dots, U_N]^T$ a ses composantes indépendantes et identiquement distribuées (*i.i.d.*) et les variables aléatoires U_i , $i = 1, \dots, N$ sont uniformes sur l'intervalle $[-1/2, 1/2[$ et indépendantes de \underline{X}^Q .

Nous étudions des matrices de transformations de forme triangulaire inférieure, dont les éléments sur la diagonale principale sont égaux à 1 :

$$\mathbf{A} = \begin{bmatrix} \underline{a}_1^T \\ \vdots \\ \underline{a}_N^T \end{bmatrix} \quad (3.2)$$

On note $\underline{Z} = \mathbf{A}\underline{X}^Q$, $\underline{Y} = \mathbf{A}\underline{X} = [Y_1, \dots, Y_N]^T$ et on pose

$$Z_i = \underline{a}_i^T \underline{X}^Q = \underline{a}_i^T \underline{X} + q\underline{a}_i^T \underline{U} = Y_i + qV_i \quad (3.3)$$

le terme $V_i = \underline{a}_i^T \underline{U}$ est une somme pondérée des éléments du vecteur \underline{U} . En supposant N assez grand, V_i peut être considérée comme une variable aléatoire gaussienne¹

$$V_i \equiv \mathcal{N}(0, \frac{\|\underline{a}_i\|^2}{12}).$$

¹La matrice \mathbf{A} étant triangulaire inférieure, pour i petit, V_i n'est pas approximativement gaussien, mais il est indépendant de Y_i et sa variance qui vaut $q^2 \frac{\|\underline{a}_i\|^2}{12}$ est négligeable devant celle de Y_i , Y_i^Q suit approximativement la loi gaussienne $\mathcal{N}(\underline{a}_i^T \underline{m}, \underline{a}_i^T \mathbf{K}_x \underline{a}_i)$.

Les bruits de quantification \underline{U} sont indépendants de $\underline{X}^{\mathcal{Q}}$, donc V_i et Y_i sont indépendantes. De plus, les variables aléatoires Y_i et V_i sont gaussiennes, ce que implique que Z_i est gaussienne² :

$$Z_i \equiv \mathcal{N}(\underline{a}_i^T \underline{m}, \underline{a}_i^T \mathbf{K}_x \underline{a}_i + q^2 \frac{\|\underline{a}_i\|^2}{12}).$$

La quantification étant faite à haute résolution, le pas de quantification q est petit, ce qui nous permet de négliger le terme $q^2 \frac{\|\underline{a}_i\|^2}{12}$ et par conséquent d'écrire

$$Z_i \equiv \mathcal{N}(\underline{a}_i^T \underline{m}, \underline{a}_i^T \mathbf{K}_x \underline{a}_i).$$

Ainsi Z_i suit la même loi que Y_i , à l'ordre 1 (par rapport à q).

En tenant compte de la forme spéciale de la matrice de transformation \mathbf{A} nous avons

$$Z_i = X_i^{\mathcal{Q}} + \sum_{j=1}^{i-1} a_{ij} X_j + q W_i \quad (3.4)$$

avec

$$W_i = \sum_{j=1}^{i-1} a_{ij} U_j.$$

La structure de la matrice de transformation \mathbf{A} permet d'obtenir une version réversible qui transforme des entiers en entiers

$$\tilde{Y}_i^{\mathcal{Q}} = X_i^{\mathcal{Q}} + q \left[\sum_{j=1}^{i-1} a_{ij} \frac{X_j}{q} + W_i + 1/2 \right] = Z_i + q B_i \quad (3.5)$$

avec

$$B_i = \left[\sum_{j=1}^{i-1} a_{ij} \frac{X_j}{q} + W_i + 1/2 \right] - \left(\sum_{j=1}^{i-1} a_{ij} \frac{X_j}{q} + W_i \right).$$

L'équation (3.5) définit une transformation réversible \mathbf{A}_{rev} non linéaire :

$$\underline{\tilde{Y}}^{\mathcal{Q}} = (\tilde{Y}_1^{\mathcal{Q}}, \dots, \tilde{Y}_N^{\mathcal{Q}})^T = \mathbf{A}_{rev} [\underline{X}^{\mathcal{Q}}]$$

En regardant la forme de $\tilde{Y}_i^{\mathcal{Q}}$ dans la relation (3.5), on peut remarquer qu'on l'obtient comme si on avait quantifié à haute résolution, avec un pas de quantification q , la variable aléatoire $\sum_{j=1}^{i-1} a_{ij} X_j + q W_i$, et en déduire que le bruit de quantification B_i est indépendant de $\tilde{Y}_i^{\mathcal{Q}}$ et suit une loi uniforme sur $[-1/2, 1/2[$. Ainsi nous avons montré qu'à l'ordre 1 par rapport à q , Z_i suit la même loi que Y_i , donc $\tilde{Y}_i^{\mathcal{Q}} = Z_i + q B_i$ suit la même loi que $Y_i^{\mathcal{Q}}$, résultat d'une quantification scalaire uniforme de pas q appliquée à Y_i . Autrement dit, les signaux $\underline{Y}^{\mathcal{Q}}(n)$ et $\tilde{\underline{Y}}^{\mathcal{Q}}(n)$ ci-dessous sont statistiquement équivalents au sens large (*i.e.*, ils ont la même loi temporelle) et donc la même entropie d'ordre 1.

²En effet, en notant ϕ_Z , ϕ_Y et ϕ_V les fonctions caractéristiques respectives de Z_i , Y_i et V_i , l'indépendance entre Z_i et V_i entraîne

$$\phi_Y(u) = \phi_Z(u) \phi_V(-qu)$$

ϕ_Y et ϕ_V étant les fonctions caractéristiques de variables aléatoires gaussiennes, cette dernière relation implique que Z est gaussienne.

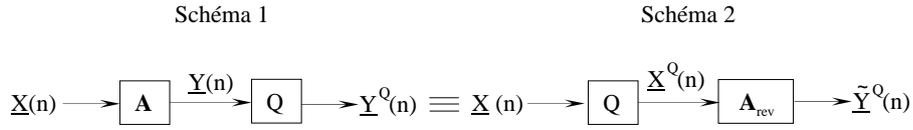


FIG. 3.1 – Les signaux $\underline{\mathbf{Y}}^{\mathcal{Q}}(n)$ et $\tilde{\underline{\mathbf{Y}}}^{\mathcal{Q}}(n)$, sorties respectives des schémas 1 et 2 sont statistiquement équivalents au sens large, ils ont donc la même entropie d'ordre 1, sous les hypothèses citées plus haut.

Nous avons également vérifié la validité de la conservation de l'entropie d'ordre 1 par des simulations en suivant une méthode de Monte Carlo et nous avons constaté, de plus, que la conservation de l'entropie d'ordre 1 restait valable pour des signaux non gaussiens.

La transformation réversible conserve l'entropie jointe

$$H(\tilde{\underline{\mathbf{Y}}}^{\mathcal{Q}}) = H(\underline{\mathbf{X}}^{\mathcal{Q}})$$

et il est tout aussi difficile de construire un codeur entropique qui donne une longueur moyenne du flot de bits égale à la valeur asymptotique $H(\underline{\mathbf{X}}^{\mathcal{Q}})$ à partir de l'image initiale, qu'à partir des coefficients transformés $\tilde{\underline{\mathbf{Y}}}^{\mathcal{Q}}$. En revanche, quand on impose au codeur entropique d'approcher l'entropie d'ordre 1 à la place du débit entropique, la transformation \mathbf{A} trouve son intérêt, car il n'y a pas conservation de la somme des entropies marginales (en général)

$$\sum_{i=1}^N H(X_i^{\mathcal{Q}}) \neq \sum_{i=1}^N H(\tilde{Y}_i^{\mathcal{Q}}).$$

Se pose alors le problème de trouver la transformation \mathbf{A} dont la version réversible suivie d'un codeur entropique d'ordre 1 minimise la taille moyenne du train de bits.

Autrement dit, chercher \mathbf{A} telle que $\sum_{i=1}^N H(\tilde{Y}_i^{\mathcal{Q}})$ soit minimale. Or d'après ce qui précède, en introduisant l'information mutuelle $I(Y_1; \dots; Y_N)$ entre les composantes de $\underline{\mathbf{Y}}$, $h(Y_i)$ l'entropie (différentielle) de la variable aléatoire continue Y_i et $h(\underline{\mathbf{Y}})$ l'entropie (différentielle) du vecteur $\underline{\mathbf{Y}}$, il vient

$$\begin{aligned} \sum_{i=1}^N H(\tilde{Y}_i^{\mathcal{Q}}) &\approx \sum_{i=1}^N H(Y_i^{\mathcal{Q}}) \approx \sum_{i=1}^N h(Y_i) - N \log_2 q \\ &\approx h(\underline{\mathbf{Y}}) + I(Y_1; \dots; Y_N) - N \log_2 q \approx h(\underline{\mathbf{X}}) + I(Y_1; \dots; Y_N) - N \log_2 \tilde{q} \end{aligned} \quad (3.6)$$

car $H(Y_i^{\mathcal{Q}}) \approx h(Y_i) - \log_2 q$ (formule de Bennett valable sous l'hypothèse d'une quantification à haute résolution) et $h(\underline{\mathbf{Y}}) = h(\underline{\mathbf{X}}) + \log_2 |\det \mathbf{A}| = h(\underline{\mathbf{X}})$ (la matrice \mathbf{A} est triangulaire avec des 1 sur la diagonale principale). Ainsi la transformation \mathbf{A} optimale est celle qui minimise l'information mutuelle $I(Y_1; \dots; Y_N)$ entre coefficients transformés (sans arrondi à l'entier le plus proche ni quantification) ou encore celle qui minimise la somme des entropies différentielles marginales des coefficients transformés.

Dans le cas de données gaussiennes \underline{X} , les coefficients transformés sont gaussiens, de variance $\sigma_{\tilde{Y}_i}^2$ pour Y_i , et la transformation \mathbf{A} optimale est celle qui minimise le produit des variances $\prod_{i=1}^N \sigma_{\tilde{Y}_i}^2$.

En général les images issues de systèmes d'acquisition ne sont pas gaussiennes et la transformation \mathbf{A} optimale est celle qui minimise l'information mutuelle entre coefficients transformés, toutefois comme nous l'avons vu aux chapitres précédents, le critère à minimiser classiquement retenu en compression d'images est le produit des variances des coefficients transformés.

En notant $\tilde{Y}_i = (Y_i - E[Y_i]) / \sigma_{Y_i}$ la variable réduite associée à Y_i , de la relation $h(aX) = h(X) + \frac{1}{2} \log_2 |a|$ (pour $a \neq 0$) [37], résulte la suivante

$$\sum_{i=1}^N h(Y_i) = \sum_{i=1}^N h(\tilde{Y}_i) + \frac{1}{2} \log_2 \left[\prod_{i=1}^N \sigma_{\tilde{Y}_i}^2 \right]. \quad (3.8)$$

Il est alors clair que choisir comme critère la minimisation de $\prod_{i=1}^N \sigma_{\tilde{Y}_i}^2$ à la place de celle de $\sum_{i=1}^N h(Y_i)$ est justifiée, si et seulement si pour toute matrice \mathbf{A} (triangulaire avec des 1 sur la diagonale principale), le vecteur $\underline{Y} = \mathbf{A}\underline{X}$ vérifie $\sum_{i=1}^N h(\tilde{Y}_i) = C^{te} = \sum_{i=1}^N h(\tilde{X}_i) = Nh(\tilde{X}_1)$, ou de façon équivalente $h(\tilde{Y}_i) = h(\tilde{X}_1)$ pour $1 \leq i \leq N$.

Dans la suite nous utiliserons ce critère (minimisation de la variance) pour adapter nos structures au signal à coder.

3.3 Structure en *lifting scheme* généralisé

Nous commençons par présenter la structure en *lifting scheme* généralisé dans le cas 1D, puis nous développerons l'étude de la structure dans le cas 2D.

3.3.1 Schéma de principe pour des signaux 1D

Le schéma de principe est présenté à la figure 3.2. Les composantes polyphases du signal d'entrée sont obtenues par une décomposition polyphase (*lazy wavelet transform*) [104, 70]. L'une d'entre elles $x_2(n)$ est estimée en utilisant deux filtres $A(z)$ et $B(z)$ et le signal de détails correspond à l'erreur d'estimation $x_2(n) - \hat{x}_2(n)$. La matrice polyphase du banc de filtres d'analyse vaut

$$E(z) = \begin{bmatrix} 1 & 0 \\ -A(z) & 1 - B(z) \end{bmatrix}. \quad (3.9)$$

C'est une version généralisée d'un *lifting scheme* classique : le filtre $B(z)$ est ajouté afin d'améliorer la prédiction. En utilisant cette transformation, le signal d'approximation x_ℓ n'est que la version sous-échantillonnée x_1 du signal d'entrée x et le signal de détails est l'erreur d'estimation. La reconstruction parfaite est assurée si $B(z) \neq 1$

et la matrice polyphase du banc de synthèse vaut alors

$$E^{-1}(z) = \frac{1}{1-B(z)} \begin{bmatrix} 1-B(z) & 0 \\ A(z) & 1 \end{bmatrix}. \quad (3.10)$$

Il faut aussi que le filtre $B(z)$ soit causal, puisqu'à la reconstruction seuls les échantillons passés de $x_2(n)$ sont disponibles pour reconstruire $\hat{x}_2(n)$. La composante polyphase x_1 étant disponible à la reconstruction, le filtre $A(z)$ peut être causal ou non. Une autre condition importante pour la reconstruction parfaite est la stabilité du filtre $\frac{1}{1-B(z)}$. En effet, des petites perturbations, dues au fait que les opérations de filtrages sont effectuées avec une arithmétique à virgule flottante, pourraient entraîner la divergence de la procédure de reconstruction quand ce filtre est instable. Par conséquent, un test de stabilité [25] devrait être nécessaire.

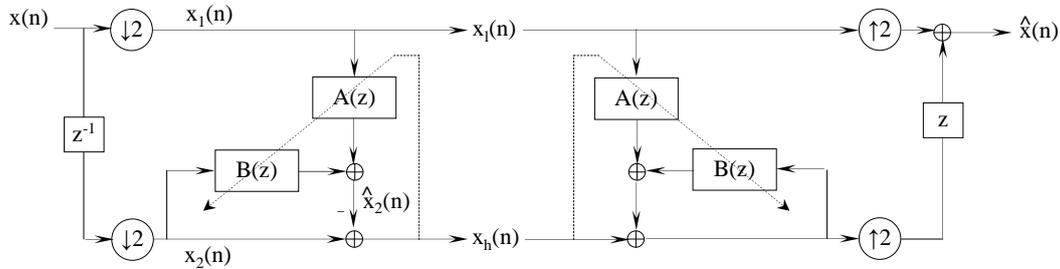


FIG. 3.2 – Étape de prédiction d'une structure en *lifting scheme* généralisé. Un filtre prédicteur supplémentaire $B(z)$ est introduit par rapport au schéma classique. Les filtres A et B sont adaptés au signal d'entrée.

Enfin, cette structure peut être munie d'une étape de mise à jour (*Update*) des *liftings schemes* classiques. Nous utiliserons par la suite la S-Transform comme étape de mise à jour avant celle de prédiction (entre la décomposition polyphase et l'étape de prédiction).

3.3.2 Extension aux signaux 2D

L'extension aux signaux bi-dimensionnels de la structure en *lifting scheme* généralisé présentée au paragraphe précédent peut être faite de deux manières différentes suivant la décomposition polyphase adoptée. Dans les deux cas les filtres $A(z_1, z_2)$ et $B(z_1, z_2)$ sont 2D et non séparables *a priori*. Nous considérons deux sortes de décomposition polyphase en deux sous-bandes. L'une correspond au sous-échantillonnage en quinconce représenté à la figure 3.3(b). Le signal $x(m, n)$ subit une rotation d'angle $\pi/4$ avant un sous-échantillonnage séparable. L'autre correspond à un sous-échantillonnage séparable sur l'image originale (voir la figure 3.3). Comme dans le cas 1D, la causalité du filtre ou la semi-causalité de $B(z_1, z_2)$ est nécessaire à la reconstruction parfaite. Les deux filtres de la figure 3.4 sont donnés par leurs

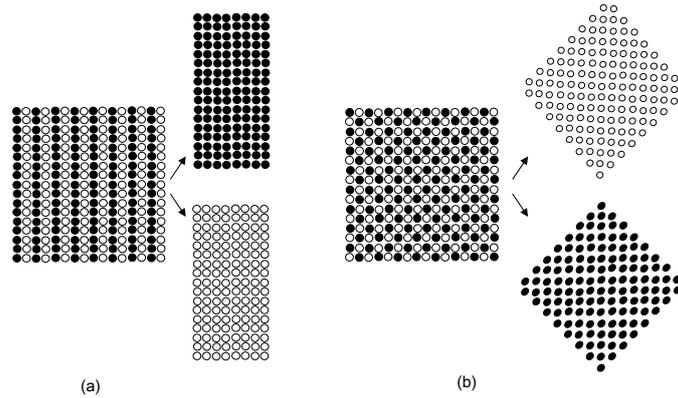


FIG. 3.3 – Illustration de la décomposition polyphase d'une image, avec un sous-échantillonnage (a) séparable et (b) en quiconce.

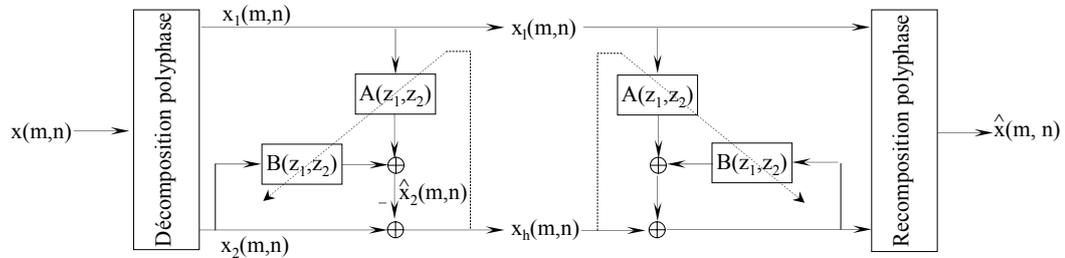


FIG. 3.4 – Étape de prédiction d'une structure en *lifting scheme* généralisé pour les signaux 2D. Un filtre prédictif supplémentaire, $B(z_1, z_2)$ est introduit par rapport au schéma classique. Les filtres A et B sont adaptés au signal d'entrée.

fonctions de transfert

$$A(z_1, z_2) = \sum_{(i,j) \in \Delta_1} a_{ij} z_1^{-i} z_2^{-j} \quad (3.11)$$

$$B(z_1, z_2) = \sum_{(i,j) \in \Delta_2} b_{ij} z_1^{-i} z_2^{-j} \quad (3.12)$$

où Δ_1 est un sous-ensemble quelconque de \mathbb{Z}^2 (aucune condition de causalité n'est imposée) et Δ_2 un sous-ensemble inclus dans un demi-plan non symétrique de \mathbb{Z}^2 pour assurer la semi-causalité du filtre $B(z_1, z_2)$.

La prédiction de $x_2(m, n)$ est donnée par

$$\hat{x}_2(m, n) = \sum_{(i,j) \in \Delta_1} a_{ij} x_1(m-i, n-j) + \sum_{(i,j) \in \Delta_2} b_{ij} x_2(m-i, n-j) \quad (3.13)$$

et le critère d'adaptation des coefficients des filtres est l'erreur quadratique moyenne

$$E [|x_2(m, n) - \hat{x}_2(m, n)|^2] \quad \text{minimale}$$

où E est l'espérance mathématique. Pour un signal $x(m, n)$ stationnaire au sens large, la théorie de l'estimation linéaire en moyenne quadratique permet d'affirmer

que l'erreur de prédiction

$$x_h(m, n) = x_2(m, n) - \hat{x}_2(m, n) \quad (3.14)$$

(c'est aussi l'image de détail après l'étape de prédiction) est décorrélée des observation $x_1(m - i, n - j)|_{(i,j) \in \Delta_1}$ et $x_2(m - i, n - j)|_{(i,j) \in \Delta_2}$.

Décorrélation De plus, en étendant au cas 2D les résultats de la prédiction à un pas et passé infini, si le signal $x(m, n)$ est WSS, alors quand les supports Δ_1 et Δ_2 des filtres tendent vers l'infini (Δ_1 remplit alors tout le plan et Δ_2 le demi-plan non symétrique), le résidu $x_h(m, n)$ tend vers un bruit blanc. Cela explique pourquoi, quand les supports Δ_1 et Δ_2 sont assez grands, les coefficients voisins de $x_h(m, n)$ sont très légèrement corrélés entre eux et avec les coefficients de la sous-bande x_ℓ . Ceci, même quand l'hypothèse de stationnarité au sens large n'est satisfaite que localement. Voir le paragraphe 4.2.1.2 où nous comparons les coefficients de corrélation entre coefficients transformés voisins inter- et intra-bandes.

Étape de mise à jour fixe Comme pour le cas 1D, la structure en *lifting scheme* généralisé que nous proposons peut être munie d'une étape de mise à jour à coefficients fixes avant l'étape de prédiction et nous verrons au chapitre suivant l'intérêt, à basse résolution, de cette étape (nous avons choisi la S-Transform comme étape de mise à jour).

Traitement des bords (extension) Les images sont des signaux 2D à support fini, il est donc nécessaire d'adopter une stratégie pour effectuer le filtrage aux bords de l'image. Les stratégies souvent utilisées sont le prolongement par des zéros, la périodisation du signal de période l'étendue du support, ou encore une extension symétrique. Les deux premières (prolongation par des zéros, périodisation) génèrent des discontinuités aux bords. Les variations brusques se traduisent par des pics de grande énergie dans le signal de détail. Dans le cas d'une extension symétrique, le problème disparaît puisque le signal est prolongé continûment par une version miroir de lui même. Nous avons choisi une extension symétrique associée à un sous-échantillonnage séparable et, pour des raisons de facilité d'implantation, un prolongement par des zéros dans le cas d'un sous-échantillonnage en quinconce.

3.3.3 Réversibilité de la structure proposée

La réversibilité de la décomposition n'est pas assurée avec l'équation (3.14), car les coefficients des filtres sont réels. Pour y remédier, il suffit d'appliquer un arrondi à l'estimation avant de la retrancher à la valeur exacte pour générer le signal de détail, qui deviendra dans ce cas

$$x_h(m, n) = x_2(m, n) - \left\lfloor \sum_{(i,j) \in \Delta_1} a_{i,j} x_1(m - i, n - j) + \sum_{(i,j) \in \Delta_2} b_{i,j} x_2(m - i, n - j) + \frac{1}{2} \right\rfloor. \quad (3.15)$$

où $\lfloor \cdot \rfloor$ est l'arrondi à l'entier inférieur le plus proche.

REMARQUE 1. Dans le cas où l'arrondi n'est pas introduit dans l'équation (3.15), un test de stabilité de $\frac{1}{1-B(z_1, z_2)}$ est nécessaire à chaque décomposition, car quand ce dernier est instable, les faibles erreurs dues à la précision limitée des calculs peuvent conduire à la divergence lors de la reconstruction. En revanche, elles sont remises à zéro si l'arrondi est utilisé. Ce qui rend inutile ce test de stabilité et réduit la complexité.

3.3.4 Décomposition multi-résolution

Le schéma présenté au paragraphe précédent (cf. fig. 3.4) est à la base de la décomposition multi-résolution d'un signal 2D. Intéressons-nous tout d'abord au cas d'un seul niveau de décomposition du signal en quatre sous-bandes. Le signal d'entrée $x(m, n)$ est remplacé par les deux signaux de sous-bande x_1 et x_2 par simple décomposition polyphase. Le signal de sous-bande d'approximation x_ℓ est égal à x_1 et le signal de sous-bande de détails x_h est obtenu par la relation (3.14) (par la relation (3.15) si la contrainte de réversibilité est imposée). Le même procédé est appliqué à x_ℓ pour générer les deux signaux de sous-bandes $x_{\ell\ell}$ et $x_{\ell h}$. Contrairement à ce qui se fait habituellement, notamment dans le cas des transformations en ondelettes, une simple décomposition polyphase est appliquée au signal x_h pour générer les deux signaux de sous-bande $x_{h\ell}$ et x_{hh} , comme indiqué sur la figure 3.5. En effet, à l'issue de la décomposition du signal suivant le schéma de la figure 3.4,

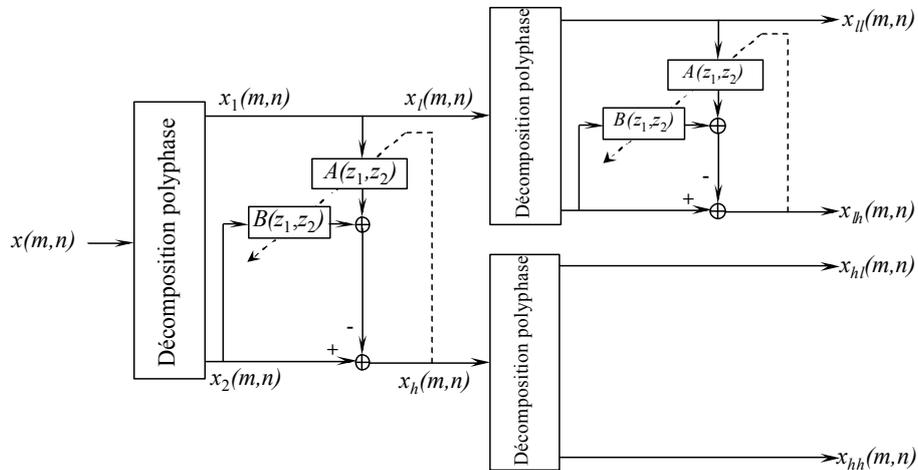


FIG. 3.5 – Un niveau de décomposition de l'image avec la structure adaptée.

la corrélation entre les échantillons du signal x_h est presque éliminée (voir les paragraphes 3.3.2 et 4.2.1.2). Par conséquent, la réduction de la variance, obtenue en appliquant un filtre prédicteur après la décomposition polyphase pour obtenir x_{hh} , est si faible ici, que nous avons choisi de ne pas appliquer de prédiction pour réduire la complexité du codec.

Pour avoir une décomposition à plusieurs niveaux, le même procédé est appliqué sur le signal de sous-bande d'approximation $x_{\ell\ell}$.

3.3.5 Adaptation de la structure 2D proposée

Pour résoudre le problème d'estimation en moyenne quadratique 2D (3.13), nous commençons par balayer les éléments des sous-ensembles Δ_1 et Δ_2 dans un ordre prédéfini. Nous construisons ainsi, à partir des familles respectives $x_1(m-i, n-j)|_{(i,j) \in \Delta_1}$ et $x_2(m-i, n-j)|_{(i,j) \in \Delta_2}$ de pixels, des vecteurs observations $\underline{y}_1(m, n)$ et $\underline{y}_2(m, n)$ de dimensions respectives r_1 et r_2 . Les balayages choisis pour décrire Δ_1 et Δ_2 imposent un ordonnancement des coefficients $a_{i,j}|_{(i,j) \in \Delta_1}$ et $b_{i,j}|_{(i,j) \in \Delta_2}$ des filtres $A(z_1, z_2)$ et $B(z_1, z_2)$ pour construire les vecteurs respectifs \underline{a} et \underline{b} tels que

$$\hat{x}_2(m, n) = [\underline{a}^T, \underline{b}^T] \begin{bmatrix} \underline{y}_1(m, n) \\ \underline{y}_2(m, n) \end{bmatrix} = \underline{c}^T \underline{y}(m, n) \quad (3.16)$$

avec $\underline{c} = [\underline{a}^T, \underline{b}^T]^T$ et $\underline{y}(m, n) = [\underline{y}_1(m, n)^T, \underline{y}_2(m, n)^T]^T$.

En appliquant, par exemple le principe d'orthogonalité, on trouve que le vecteur \underline{c} est solution des équations normales

$$\Gamma_Y \underline{c} = \underline{\gamma}_{yx} \quad (3.17)$$

avec $\Gamma_Y = E [\underline{y}(m, n) \underline{y}(m, n)^T]$ et $\underline{\gamma}_{yx} = E [\underline{y}(m, n) x_2(m, n)]$. Nous verrons par la suite comment estimer ces espérances à partir des données. Nous distinguerons deux cas, suivant l'hypothèse de stationnarité au sens large faite sur le signal $x(m, n)$: globale (i.e., sur toute l'image) ou locale.

3.4 Méthode d'adaptation globale (GAE)

3.4.1 Choix du critère

Le signal d'entrée est supposé être une réalisation d'un processus aléatoire bi-dimensionnel stationnaire au sens large. Le sous-ensemble $\Delta_1 \subset \mathbb{Z}^2$ est un rectangle centré en $(0, 0)$:

$$\Delta_1 = \{(i, j) \in \mathbb{Z}^2 : |i| \leq p \text{ et } |j| \leq q\} \quad (3.18)$$

et le sous-ensemble $\Delta_2 \subset \mathbb{Z}^2$ est un demi-plan non symétrique borné :

$$\Delta_2 = \left\{ (i, j) \in \mathbb{Z}^2 : \begin{array}{l} (i = 0 \text{ et } 1 \leq j \leq q) \\ \text{ou} \\ (1 \leq i \leq p \text{ et } |j| \leq q) \end{array} \right\}. \quad (3.19)$$

Les vecteurs,

$$\begin{aligned} \underline{y}_1(m, n) = & [x_1(m+p, n+q), \dots, x_1(m+p, n-q), \\ & x_1(m+p-1, n+q), \dots, x_1(m+p-1, n-q), \dots \\ & \dots, x_1(m-p+1, n+q), \dots, x_1(m-p+1, n-q), \\ & x_1(m-p, n+q), \dots, x_1(m-p, n-q)]^T \end{aligned} \quad (3.20)$$

de dimension $r_1 = (2p + 1)(2q + 1)$ et

$$\begin{aligned} \underline{y}_2(m, n) = [& x_2(m, n - 1), \dots, x_2(m, n - q), \\ & x_2(m - 1, n + q), \dots, x_2(m - 1, n - q), \dots \\ & \dots, x_2(m - p + 1, n + q), \dots, x_2(m - p + 1, n - q), \\ & x_2(m - p, n + q), \dots, x_2(m - p, n - q)]^T \end{aligned} \quad (3.21)$$

de dimension $r_2 = p(2q + 1) + q$, contiennent les valeurs des échantillons respectifs $x_1(m - i, n - j)|_{(i,j) \in \Delta_1}$ et $x_2(m - i, n - j)|_{(i,j) \in \Delta_2}$. Les vecteurs \underline{a} et \underline{b} valent

$$\underline{a} = [a_{-p,-q}, \dots, a_{-p,q}, a_{-p+1,-q}, \dots, a_{-p+1,q}, \dots, a_{p,-q}, \dots, a_{p,q}]^T \quad (3.22)$$

et

$$\underline{b} = [b_{0,1}, \dots, b_{0,q}, b_{1,-q}, \dots, b_{1,q}, \dots, b_{p,-q}, \dots, b_{p,q}]^T. \quad (3.23)$$

Le critère à minimiser vaut ici

$$J_1 = E [x_h(m, n)^2] = \frac{1}{M_2 N_2} \sum_{m=1}^{M_2} \sum_{n=1}^{N_2} x_h(m, n)^2 \quad (3.24)$$

où (M_2, N_2) est la dimension de x_2 .

Regroupons les éléments $x_h(m, n)|_{1 \leq m \leq M_2, 1 \leq n \leq N_2}$ dans le vecteur $\underline{\mathcal{X}}_h$ en balayant x_h ligne par ligne :

$$\underline{\mathcal{X}}_h = [x_h(1, 1), \dots, x_h(1, N_2), \dots, x_h(M_2, 1), \dots, x_h(M_2, N_2)]^T \quad (3.25)$$

nous avons alors l'écriture matricielle suivante :

$$\underline{\mathcal{X}}_h = \underline{\mathcal{X}}_2 - \mathcal{Y}\underline{c} \quad (3.26)$$

avec

$$\underline{\mathcal{X}}_2 = [x_2(1, 1), \dots, x_2(1, N_2), \dots, x_2(M_2, 1), \dots, x_2(M_2, N_2)]^T \quad (3.27)$$

et la matrice de dimension $M_2 N_2 \times r$, qui vaut

$$\mathcal{Y} = [\underline{y}(1, 1), \dots, \underline{y}(1, N_2), \dots, \underline{y}(M_2, 1), \dots, \underline{y}(M_2, N_2)]^T. \quad (3.28)$$

Le critère à minimiser (3.24) s'écrit

$$M_2 N_2 J_1 = \|\underline{\mathcal{X}}_h\|_2^2 = \|\underline{\mathcal{X}}_2 - \mathcal{Y}\underline{c}\|_2^2 = [\underline{\mathcal{X}}_2 - \mathcal{Y}\underline{c}]^T [\underline{\mathcal{X}}_2 - \mathcal{Y}\underline{c}] \quad (3.29)$$

et les coefficients du vecteur \hat{c} sont solutions des équations de Yule-Walker

$$\mathcal{Y}^T \mathcal{Y} \hat{c} = \mathcal{Y}^T \underline{\mathcal{X}}_2. \quad (3.30)$$

Si la matrice $\mathcal{Y}^T \mathcal{Y}$ n'est pas singulière, alors le vecteur optimal est donné par :

$$\hat{c} = (\mathcal{Y}^T \mathcal{Y})^{-1} \mathcal{Y}^T \underline{\mathcal{X}}_2. \quad (3.31)$$

Étant donnés les ordres des filtres $A(z_1, z_2)$ et $B(z_1, z_2)$, les vecteurs $\underline{y}_1(m, n)$ et $\underline{y}_2(m, n)$ ont des supports situés en dehors de l'horizon d'observation de x . Les trois

types de fenêtrage des données décrits ci-dessous ont été implantés et testés.

Méthode pré-fenêtrée. Les signaux de sous-bandes $x_1(m, n)$ et $x_2(m, n)$ sont supposés nuls pour $m \leq 0$ ou $n \leq 0$, et aucune hypothèse n'est faite sur leurs valeurs pour $m > M_2$ ou $n > N_2$.

Méthode d'autocorrélation. Les signaux de sous-bande $x_1(m, n)$ et $x_2(m, n)$ sont supposés nuls en dehors de l'horizon d'observation.

Méthode de covariance. Aucune hypothèse n'est faite sur les valeurs des signaux x_1 et x_2 en dehors de l'horizon d'observation.

D'après les expérimentations que nous avons faites (sur des images), la méthode d'autocorrélation donne assez souvent et en moyenne des entropies d'ordre 1 légèrement plus élevées que les deux autres méthodes. Dans la suite de nos simulations, nous avons adopté la méthode pré-fenêtrée, qui offre souvent et en moyenne les mêmes performances que la méthode de covariance, avec une complexité moindre.

3.4.2 Calcul rapide de la matrice du système

Dans ce paragraphe, nous noterons $e(m, n) = x_2(m, n) - \hat{x}_2(m, n)$ à la place de $x_h(m, n)$, pour disposer de la lettre h comme indice. L'étape de la méthode GAE qui demande le plus d'opérations est le calcul des coefficients des équations de Yule-Walker. Il est donc important de réduire sa complexité en tenant compte de la grande redondance entre ces coefficients. Nous présentons dans ce paragraphe les détails de la méthode proposée, c'est une extension de la méthode pré-fenêtrée [45] appliqués au cas de la prédiction linéaire 2D.

Pour deux vecteurs $\underline{u} \in \mathbb{R}^\alpha$ et $\underline{v} \in \mathbb{R}^\beta$, on notera $\mathbf{T}[\underline{u}, \underline{v}]$ la matrice de Toeplitz de dimension $\alpha \times \beta$ dont la première colonne (resp. ligne) est égale à \underline{u} (resp. \underline{v}^T). Pour une matrice \mathbf{A} de dimension $m \times n$, l'élément situé à l'intersection de la ligne $(i+1)$ et de la colonne $(j+1)$ ($0 \leq i < m$ et $0 \leq j < n$) sera noté $[\mathbf{A}]_{i,j}$. Introduisons le vecteur $\underline{b}_0 = (0, \dots, 0, -1, b_{0,1}, b_{0,2}, \dots, b_{0,q})^T$ de dimension $2q+1$, les p vecteurs $\underline{b}_h = (b_{h,-q}, b_{h,-q+1}, \dots, b_{h,q})^T$ ($1 \leq h \leq p$) de dimension $2q+1$, les $2p+1$ vecteurs $\underline{a}_h = (a_{h,-q}, a_{h,-q+1}, \dots, a_{h,q})^T$ ($-p \leq h \leq p$) de dimension $2q+1$ et les vecteurs $\underline{b} = (\underline{b}_0^T, \underline{b}_1^T, \dots, \underline{b}_p^T)^T$ et $\underline{a} = (\underline{a}_{-p}^T, \underline{a}_{-p+1}^T, \dots, \underline{a}_p^T)^T$.

Après la décomposition polyphase appliquée à l'image $x(m, n)$ de dimension $M \times N$, les images de sous-bandes $x_1(m, n)$ et $x_2(m, n)$ ont pour dimensions respectives $M_1 \times N_1$ et $M_2 \times N_2$.

À la $(m+1)$ ème ligne du signal de sous-bande x_2 , associons les $p+1$ matrices de Toeplitz ($0 \leq h \leq p$)

$$\begin{aligned} \mathbf{X}_h(m) = & \mathbf{T}[(x_2(m-h, q), \dots, x_2(m-h, N_2-1))^T, \\ & (x_2(m-h, q), \dots, x_2(m-h, 0), 0, \dots, 0)^T] \end{aligned} \quad (3.32)$$

de dimension $(N_2 - q) \times (2q + 1)$ et la matrice bloc

$$\mathbf{X}(m) = [\mathbf{X}_0(m) \quad \mathbf{X}_1(m) \quad \dots \quad \mathbf{X}_p(m)] \quad (3.33)$$

de dimension $(N_2 - q) \times (2q + 1)(p + 1)$. De même, à la $(m + 1)^{\text{ème}}$ ligne du signal x_1 associons les $2p + 1$ matrices de Toeplitz $(-p \leq h \leq p)$

$$\mathbf{Y}_h(m) = \mathbf{T}[(x_1(m - h, q), \dots, x_1(m - h, N_2 - 1))^T, \\ (x_1(m - h, q), \dots, x_1(m - h, 0), 0, \dots, 0)^T] \quad (3.34)$$

de dimension $(N_2 - q) \times (2q + 1)$ et la matrice bloc

$$\mathbf{Y}(m) = [\mathbf{Y}_{-p}(m) \quad \mathbf{Y}_{-p+1}(m) \quad \dots \quad \mathbf{Y}_p(m)]$$

de dimension $(N_2 - q) \times (2p + 1)(2q + 1)$. Avec ces notations, l'erreur d'estimation

$$\underline{e}(m) = (x_2(m, 0) - \hat{x}_2(m, 0), \dots, \\ x_2(m, N_2 - q - 1) - \hat{x}_2(m, N_2 - q - 1))^T$$

associée à la $(m + 1)^{\text{ème}}$ ligne de x_2 vérifie la relation $-\underline{e}(m) = \mathbf{X}(m)\underline{b} + \mathbf{Y}(m)\underline{a}$. De plus, il résulte des relations (3.32) et (3.34) que

$$\mathbf{Y}_k(m) = \mathbf{Y}_{k-1}(m - 1) = \mathbf{Y}_0(m - k),$$

$\mathbf{X}_k(m) = \mathbf{Y}_k(m) = \mathbf{0}$ (si $m < k$) et que pour $0 < k \leq p$ et $0 \leq m < M_2 - p - 1$

$$\mathbf{X}_k(m) = \mathbf{X}_{k-1}(m - 1) = \mathbf{X}_0(m - k). \quad (3.35)$$

Les équations d'estimation donnent alors

$$-(\underline{e}(0)^T, \underline{e}(1)^T, \dots, \underline{e}(M_2 - p - 1)^T)^T = \mathbf{X}\underline{b} + \mathbf{Y}\underline{a}, \quad (3.36)$$

avec les matrices Toeplitz par bloc \mathbf{X} et \mathbf{Y} , dont les premières colonnes sont respectivement $(\mathbf{X}_0(0)^T, \mathbf{X}_0(1)^T, \dots, \mathbf{X}_0(M_2 - p - 1)^T)^T$ et $(\mathbf{Y}_0(p)^T, \mathbf{Y}_0(p+1)^T, \dots, \mathbf{Y}_0(M_2 - 1)^T)^T$ et dont les premières lignes sont respectivement $(\mathbf{X}_0(0), \mathbf{0}, \dots, \mathbf{0})$ et $(\mathbf{Y}_0(p), \mathbf{Y}_0(p-1), \dots, \mathbf{Y}_0(0), \mathbf{0}, \dots, \mathbf{0})$. Puisque chaque bloc $\mathbf{X}_0(k)$ ou $\mathbf{Y}_0(k)$ est de Toeplitz, nous pouvons remarquer que les matrices \mathbf{X} et \mathbf{Y} sont à la fois de Toeplitz et Toeplitz par bloc. Les coefficients optimaux du filtre sont solutions des équations de Yule-Walker [53] que l'on peut écrire

$$(\underline{b}^T, \underline{a}^T)\mathbf{C} = (\underline{0}^T, -W, \underline{0}^T),$$

où

$$\mathbf{C} = \begin{bmatrix} \mathbf{X} & \mathbf{Y} \end{bmatrix}^T \begin{bmatrix} \mathbf{X} & \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{X} & \mathbf{X}^T\mathbf{Y} \\ \mathbf{Y}^T\mathbf{X} & \mathbf{Y}^T\mathbf{Y} \end{bmatrix} \quad (3.37)$$

est une matrice symétrique d'ordre $\omega' = (2q + 1)(3p + 2) + 1$. Puisque les $p + 1$ premières composantes de \underline{b} sont connues, la dimension effective du système est $\omega = 6pq + 3p + 3q + 2$. Toutefois, il est plus rapide de calculer d'abord l'entière matrice \mathbf{C} puis d'en extraire la matrice du système, que de calculer directement cette dernière. La relation (3.36) diffère des équations bien connues rencontrées avec la méthode préfenêtrée [45] [53] à cause des termes supplémentaires $\mathbf{Y}\underline{a}$. Par conséquent, le rang de déplacement de la matrice \mathbf{C} ne vaut pas 1 [45]. Mais les éléments de \mathbf{C} ont encore une grande redondance et le raisonnement présenté dans [61] peut s'adapter à cette matrice.

Pour être bref, seules les relations qui permettent de calculer le bloc $\mathbf{X}^T \mathbf{X}$ de \mathbf{C} sont données. Le même raisonnement pourra être effectué pour les autres blocs. Il résulte des équations (3.36) et (3.33) que

$$\mathbf{X}^T \mathbf{X} = \sum_{m=0}^{M_2-p-1} \mathbf{X}(m)^T \mathbf{X}(m) = \left[\sum_{m=0}^{M_2-p-1} \mathbf{X}_k(m)^T \mathbf{X}_\ell(m) \right]$$

(pour $0 \leq k, \ell < p$), où la dernière expression est une représentation par bloc de $\mathbf{X}^T \mathbf{X}$. Soit

$$\mathbf{T}_{k,\ell} = \sum_{m=0}^{M_2-p-1} \mathbf{X}_k(m)^T \mathbf{X}_\ell(m) \quad (0 \leq k, \ell < p)$$

le bloc de dimension $(2q+1) \times (2q+1)$. On a $\mathbf{T}_{k,\ell}^T = \mathbf{T}_{\ell,k}$. La relation (3.35) donne

$$\mathbf{T}_{k,\ell} = \mathbf{T}_{k-1,\ell-1} - \mathbf{X}_0(M_2-p-k)^T \mathbf{X}_0(M_2-p-\ell)$$

pour $1 \leq k \leq \ell \leq p$. Il résulte directement de (3.32) que $\mathbf{X}_0(m-k)^T \mathbf{X}_0(m-\ell)$ ($0 \leq k \leq \ell \leq p$ et $\ell \leq m < M_2-p$) a un rang de déplacement de 2. Par conséquent, pour $0 \leq \ell \leq p$, on a

$$\begin{aligned} [\mathbf{T}_{0,\ell}]_{i,j} &= [\mathbf{T}_{0,\ell}]_{i-1,j-1} + \sum_{m=\ell}^{M_2-p-1} x_2(m, q-i)x_2(m-\ell, q-j) \\ &\quad - \sum_{m=\ell}^{M_2-p-1} x_2(m, N_2-i)x_2(m-\ell, N_2-j). \end{aligned}$$

En conclusion, le calcul d'un bloc nécessite (en additions et multiplications) :

- $O(N_2)$ opérations pour $\mathbf{T}_{k,\ell}$ avec $0 \leq k \leq \ell \leq p$,
- $2(4q+1)M_2N_2 + o(M_2N_2)$ opérations pour $\mathbf{T}_{0,\ell}$ avec $p \geq \ell \geq 1$,
- $2(2q+1)M_2N_2 + o(M_2N_2)$ opérations pour $\mathbf{T}_{0,0}$,

et le calcul de $\mathbf{X}^T \mathbf{X}$ nécessite $2[(4q+1)p + 2q+1]M_2N_2 + o(M_2N_2)$ opérations. De même, le calcul de $\mathbf{X}^T \mathbf{Y}$ nécessite $2(4q+1)(3p+1) + o(M_2N_2)$ opérations et celui de $\mathbf{Y}^T \mathbf{Y}$ $2[(4q+1)2p + 2q+1]M_2N_2 + o(M_2N_2)$ opérations. Enfin, le calcul de \mathbf{C} coûte $2[2(4q+1)(3p+1) + 1]M_2N_2 + o(M_2N_2)$ opérations.

3.5 Méthode d'adaptation locale (LAE)

3.5.1 Choix du critère

Le signal d'entrée est supposé être une réalisation d'un processus aléatoire bi-dimensionnel aux statistiques d'ordres 1 et 2 localement stationnaires. Nous appliquons alors les moindres carrés adaptatifs. Pour cela, nous introduisons un facteur d'oubli α , $0 < \alpha \ll 1$, et le critère à minimiser (estimation locale de la variance du résidu de prédiction) vaut

$$J_2(m, n) = \sum_{i=1}^{m-1} \sum_{j=1}^{N_2} \alpha^{N_2(m-i)+n-j} x_h(i, j)^2 + \sum_{i=1}^n \alpha^{n-j} x_h(m, j)^2 \quad (3.38)$$

avec

$$x_h(i, j) = x_2(i, j) - \underline{c}^T(m, n) \underline{y}(i, j). \quad (3.39)$$

Remarquons que, dans cette relation, nous avons utilisé l'indice (m, n) pour le vecteur \underline{c} et l'indice (i, j) pour les autres variables. C'est dû à la nature de notre optimisation : pour chaque (m, n) , $\underline{c}(m, n)$ dépend de $\underline{y}(i, j)$ et $x_2(i, j)$ pour $1 \leq i \leq m$ et $1 \leq j \leq n$. La manière avec laquelle est appliqué le facteur d'oubli est propre à la méthode de balayage de l'image (de gauche à droite et de haut en bas).

Comme nous l'avons déjà dit, pour la méthode LAE des sous-ensembles de Δ_1 et Δ_2 sont utilisés. En général dans une image, les pixels qui contiennent le plus d'information pour estimer le pixel courant sont ses plus proches voisins. Il est donc naturel de réordonner les composantes des vecteurs $\underline{y}_1(m, n)$ et $\underline{y}_2(m, n)$, de telle sorte que leur ordre d'apparition soit une fonction croissante de la distance au pixel courant. Ainsi, seuls r_1 coefficients de Δ_1 sont retenus, ceux numérotés de 1 à r_1 dans

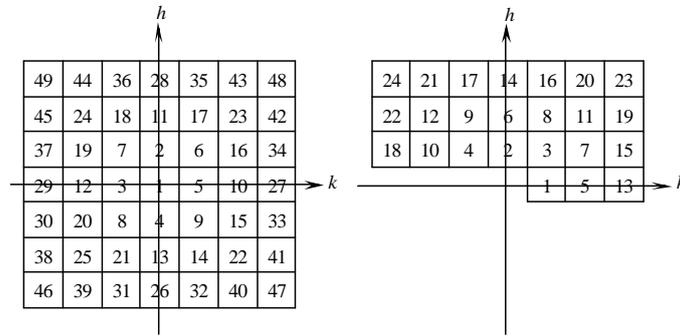


FIG. 3.6 – *Ordre d'apparition des échantillons dans le vecteur observation pour $p \leq 3$, à gauche dans Δ_1 , à droite Δ_2 (méthode LAE).*

le schéma de gauche de la figure 3.6, et seuls r_2 coefficients de Δ_2 sont retenus, ceux numérotés de 1 à r_2 sur le schéma de droite de la figure 3.6. Cela permet de diminuer l'ordre des filtres et par conséquent le temps de calcul sans réduire les performances.

En injectant (3.39) dans (3.38), le critère J_2 peut s'écrire comme suit :

$$J_2(m, n) = \kappa(m, n) - 2\underline{c}^T(m, n)\underline{\Theta}(m, n) + \underline{c}^T(m, n)\underline{\Phi}(m, n)\underline{c}^T(m, n) \quad (3.40)$$

avec

$$\kappa(m, n) = \sum_{i=1}^{m-1} \sum_{j=1}^{N_2} \alpha^{N_2(m-i)+n-j} x_2(i, j)^2 + \sum_{j=1}^n \alpha^{n-j} x_2(m, j)^2 \quad (3.41)$$

$$\underline{\Theta}(m, n) = \sum_{i=1}^{m-1} \sum_{j=1}^{N_2} \alpha^{N_2(m-i)+n-j} x_2(i, j) \underline{y}(i, j) + \sum_{j=1}^n \alpha^{n-j} x_2(m, j) \underline{y}(m, j) \quad (3.42)$$

et

$$\underline{\Phi}(m, n) = \sum_{i=1}^{m-1} \sum_{j=1}^{N_2} \alpha^{N_2(m-i)+n-j} \underline{y}(i, j) \underline{y}(i, j)^T + \sum_{j=1}^n \alpha^{n-j} \underline{y}(m, j) \underline{y}(m, j)^T. \quad (3.43)$$

Le critère $J_2(m, n)$ peut maintenant être minimisé en annulant ses dérivées partielles par rapport à $\underline{c}(m, n)$:

$$\frac{\partial J_2(m, n)}{\partial \underline{c}(m, n)} = -2\underline{\Theta}(m, n) + 2\underline{\Phi}(m, n)\underline{c}(m, n). \quad (3.44)$$

Nous obtenons les équations de Yule-Walker :

$$\mathbf{\Phi}(m, n)\underline{c}(m, n) = \underline{\Theta}(m, n) \quad (3.45)$$

ayant pour solution

$$\hat{c}(m, n) = \mathbf{\Phi}(m, n)^{-1}\underline{\Theta}(m, n) \quad (3.46)$$

Des relations (3.42) et (3.43) nous déduisons les expressions récursives suivantes

$$\mathbf{\Phi}(m, n) = \alpha\mathbf{\Phi}(m, n-1) + \underline{y}(m, n)\underline{y}^T(m, n) \quad (3.47)$$

et

$$\underline{\Theta}(m, n) = \alpha\underline{\Theta}(m, n-1) + x_2(m, n)\underline{y}^T(m, n). \quad (3.48)$$

La méthode des moindres carrés adaptatifs donne une expression de $\mathbf{\Phi}^{-1}(m, n)$ en fonction de $\mathbf{\Phi}^{-1}(m, n-1)$ de sorte que l'équation (3.45) soit résolue sans avoir recours à l'inversion de la matrice $\mathbf{\Phi}(m, n)$ à chaque étape (m, n) . Pour cela, nous utilisons le lemme d'inversion matricielle,

$$(\mathbf{S}^{-1} + \mathbf{U}\mathbf{V}\mathbf{U}^T)^{-1} = \mathbf{S} - \mathbf{S}\mathbf{U}[\mathbf{V} + \mathbf{U}^T\mathbf{S}\mathbf{U}]^{-1}\mathbf{U}^T\mathbf{S},$$

pour

$$\mathbf{R} = \mathbf{\Phi}(m, n), \quad \mathbf{S}^{-1} = \alpha\mathbf{\Phi}^{-1}(m, n-1), \quad \mathbf{U} = \underline{Y}(m, n), \quad \mathbf{V}^{-1} = 1$$

ce qui donne

$$\mathbf{\Phi}^{-1}(m, n) = \alpha^{-1}\mathbf{\Phi}^{-1}(m, n-1) - \frac{\alpha^{-1}\mathbf{\Phi}^{-1}(m, n-1)\underline{Y}(m, n)\underline{Y}^T(m, n)\mathbf{\Phi}^{-1}(m, n-1)}{\alpha + \underline{Y}^T(m, n)\mathbf{\Phi}^{-1}(m, n-1)\underline{Y}(m, n)} \quad (3.49)$$

Cette relation nécessite une valeur initiale $\mathbf{\Phi}(1, 1)$ afin de commencer la récursion. La manière la plus simple est de la prendre égale à $\delta\mathbf{I}$, où δ est une constante de petite valeur. Soit le vecteur $\underline{g}(m, n)$ donné par

$$\underline{g}(m, n) = \frac{\mathbf{\Phi}^{-1}(m, n-1)\underline{Y}(m, n)}{\alpha + \underline{Y}^T(m, n)\mathbf{\Phi}^{-1}(m, n-1)\underline{Y}(m, n)} \quad (3.50)$$

l'équation (3.49) devient

$$\mathbf{\Phi}^{-1}(m, n) = \alpha^{-1}\mathbf{\Phi}^{-1}(m, n-1) - \alpha^{-1}\underline{g}(m, n)\underline{Y}^T(m, n)\mathbf{\Phi}^{-1}(m, n-1) \quad (3.51)$$

et l'équation (3.50) peut être réarrangée pour avoir

$$\underline{g}(m, n) = \mathbf{\Phi}^{-1}(m, n)\underline{Y}(m, n) \quad (3.52)$$

ce vecteur est appelé le *gain d'adaptation* car il apparaît comme gain appliqué à l'erreur de prédiction dans l'équation de mise à jour de \hat{c}

$$\hat{c}(m, n) = \hat{c}(m, n-1) + \underline{g}(m, n) [x_2(m, n) - \hat{c}^T(m, n-1)\underline{y}(m, n)]. \quad (3.53)$$

Nous pouvons maintenant résumer la procédure de décomposition adaptative.

3.5.2 Algorithme de décomposition

```

Décomposition polyphase du signal 2D d'entrée en deux signaux  $x_1, x_2$ 
Initialisations
   $\underline{c} = [1, 0, \dots, 0]^T \in \mathbb{R}^{r_1+r_2}$ 
   $\Phi^{-1} = \delta^{-1}\mathbf{I}, 0 < \delta \ll 1.$ 
   $\alpha$ , facteur d'oubli
Pour  $m = 1, 2, \dots$ 
  Pour  $n = 1, 2, \dots$ 
    construire le vecteur  $\underline{y}_1(m, n).$ 
    construire le vecteur  $\underline{y}_2(m, n).$ 
    construire le vecteur observation  $\underline{y}(m, n) = [\underline{y}_1^T(m, n), \underline{y}_2^T(m, n)]^T$ 
     $x_\ell(m, n) = x_1(m, n)$ 
     $x_h(m, n) = x_2(m, n) - \underline{c}^T \underline{y}(m, n)$ 
     $\Phi^{-1} = \alpha^{-1}\Phi^{-1} - [\alpha^{-1}\Phi^{-1}\underline{y}(m, n)\underline{y}(m, n)^T\Phi^{-1}] / [\alpha + \underline{y}(m, n)^T\Phi^{-1}\underline{y}(m, n)]$ 
     $\underline{g}(m, n) = \Phi^{-1}\underline{y}(m, n)$ 
     $\underline{c} = \underline{c} + \underline{g}(m, n)x_h(m, n)$ 
  fin boucle  $n$ 
fin boucle  $m$ 

```

La complexité³ de la méthode LAE est en $O(M_2N_2(r_1 + r_2)^2)$ [53] pour un niveau de décomposition. Notons que cette procédure de décomposition utilise l'algorithme des moindres carrés adaptatif unidimensionnel [15, 53]. La construction du vecteur observation $\underline{y}(m, n)$ dans lequel est réarrangé le voisinage bi-dimensionnel de l'échantillon $x_2(m, n)$ donne à cet algorithme un caractère 2D, c'est pour cette raison que nous gardons les indices (m, n) dans la notation des vecteurs.

3.6 Application à des signaux synthétiques

3.6.1 Construction des signaux

Deux groupes de signaux 2-D synthétiques gaussiens, codés sur 8 bits, sont utilisés dans les tests qui suivent. Le premier groupe est composé de 8 signaux AR-2D de dimension 512×512 et globalement stationnaires. La moitié de ces signaux sont générés selon un modèle AR-2D avec une causalité quart-plan et l'autre moitié avec une causalité demi-plan non-symétrique. Ils sont présentés figure 3.7.

Le second groupe est un ensemble de 8 images de dimension 512×512 , dont les différentes zones qui les composent sont des motifs AR-2D simulant une stationnarité locale (c'est-à-dire par zone). Les textures des différentes zones ont été générées suivant des modèles AR-2D avec causalité quart-plan ou demi-plan non-symétrique (choisie au hasard). Ils sont montrés à la figure 3.7.

³Nous n'avons pas implanté une version rapide (voir [66]) de l'algorithme des moindres carrés adaptatifs.

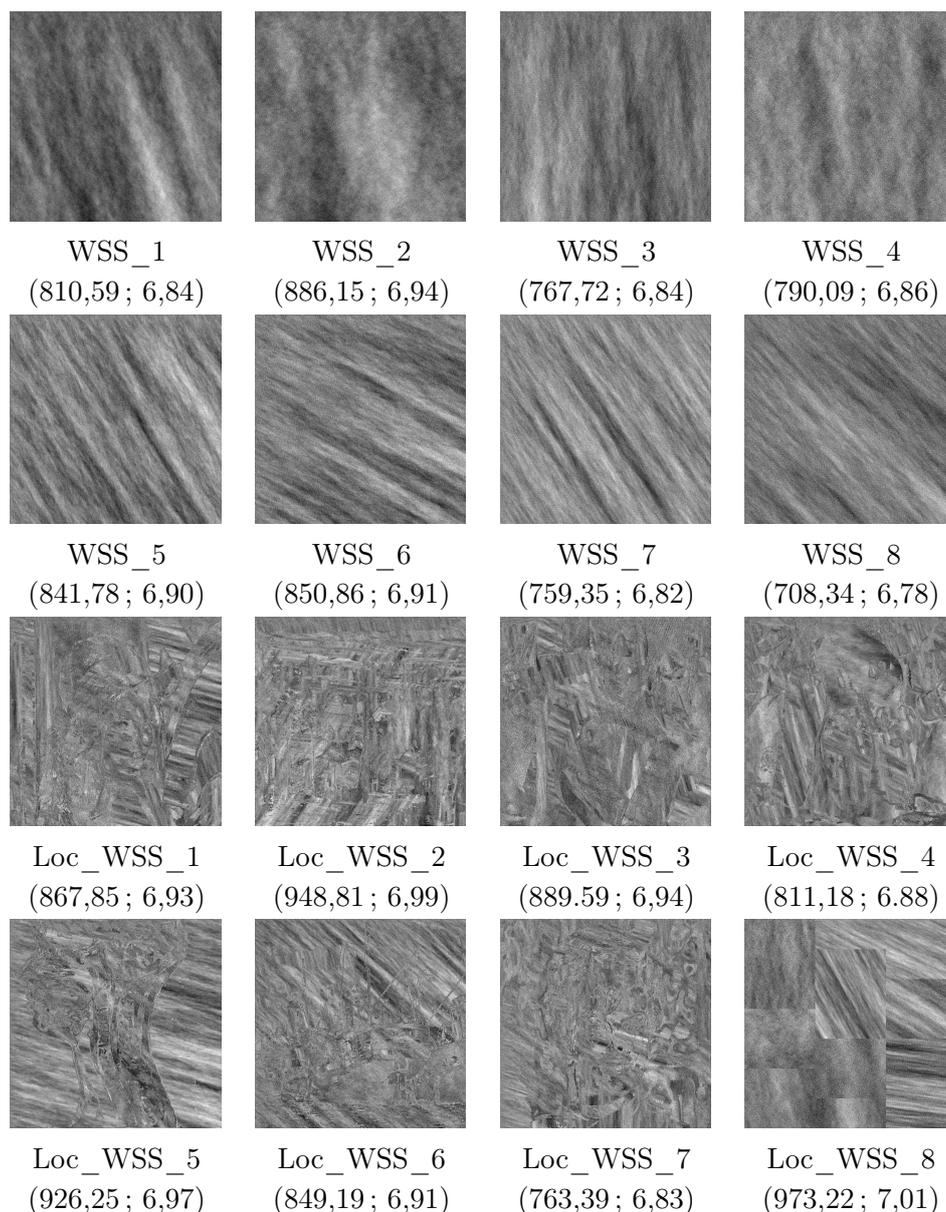


FIG. 3.7 – Signaux 2-D synthétiques, globalement stationnaires (deux lignes du haut) et localement stationnaires (deux lignes du bas). Sur la première ligne la causalité du modèle est demi-plan non symétrique et sur la seconde quart-plan. Sur les deux dernières lignes, la causalité est quart-plan ou demi plan non symétrique par zone (ce choix est fait au hasard). Les valeurs indiquées sous chaque image sont respectivement sa variance et son entropie d'ordre 1.

3.6.2 Résultats

Dans ce paragraphe, nous comparons, pour chaque signal synthétique ci-dessus, la variance et l'entropie d'ordre 1 de l'image de sous-bandes x_h , issue de la version réversible du schéma de la figure 3.4 (voir § 3.3.3), avec les méthodes GAE ($p = 3$, $q = 3$, sous-échantillonnage séparable), LAE ($r_1 = 16$, $r_2 = 8$, $\alpha = 0,9995$, sous-échantillonnage séparable), les décompositions en ondelette (9,7) et (5,3) de Daube-

chies et la méthode de Gerek & Çetin [46, 47], décrite plus haut. Les tableaux 3.1 à 3.4 présentent les performances des différentes décompositions testées.

Pour ces deux types de signaux, les performances des décompositions GAE et LAE sont bien supérieures à celles des autres décompositions et très proches entre elles. En moyenne 0,5 bpp pour les signaux globalement WSS ou 0,7 bpp pour ceux localement WSS de moins que la meilleure des autres décompositions. Nous constatons aussi, et c'est ce qui était attendu, que pour les signaux globalement stationnaires la méthode GAE donne des résultats légèrement meilleurs que LAE et que pour les signaux localement stationnaires c'est la méthode LAE qui devance légèrement GAE.

Nous constatons également que l'ajout du filtre $B(z_1, z_2)$ améliore sensiblement (de 0,6 bpp ou 0,7 bpp en moyenne) les performances du *lifting scheme* adapté, grâce à la comparaison avec la méthode de Gerek & Çetin.

	GAE	LAE	Gerek <i>et al.</i>	(9,7)	(5,3)
WSS_1	108,08	112,84	185,08	216,5	325,72
WSS_2	181,05	185,66	231,69	359,2	548,13
WSS_3	189,50	195,35	324,65	355,7	542,12
WSS_4	278,43	287,38	509,71	555,5	849,29
WSS_5	143,38	146,77	271,35	269,6	404,81
WSS_6	176,45	189,09	315,61	364,6	556,35
WSS_7	172,06	176,28	304,69	366,8	559,34
WSS_8	194,78	199,63	353,74	401,7	613,96

TAB. 3.1 – Variance du signal de sous-bande la x_h pour différentes décompositions en *lifting scheme*. Les signaux d'entrée WSS_1 à WSS_8 sont décrits au § 3.6.1.

	GAE	LAE	Gerek <i>et al.</i>	(9,7)	(5,3)
WSS_1	5,42	5,44	5,81	5,93	6,22
WSS_2	5,80	5,81	6,21	6,29	6,59
WSS_3	5,83	5,85	6,20	6,28	6,58
WSS_4	6,11	6,13	6,54	6,60	6,91
WSS_5	5,63	5,64	6,08	6,08	6,38
WSS_6	5,78	5,80	6,19	6,30	6,61
WSS_7	5,76	5,78	6,17	6,31	6,61
WSS_8	5,85	5,87	6,27	6,37	6,68
moyenne	5,77	5,79	6,18	6,27	6,57

TAB. 3.2 – Entropie d'ordre 1 du signal de la sous-bande x_h pour les différentes décompositions en *lifting scheme*. Les signaux d'entrée WSS_1 à WSS_8 sont décrits au § 3.6.1.

	GAE	LAE	Gerek <i>et al.</i>	(9,7)	(5,3)
Loc_WSS_1	129,02	127,36	337,76	418,8	639,98
Loc_WSS_2	123,97	117,76	344,49	409,9	631,33
Loc_WSS_3	141,42	141,58	356,79	471,3	719,10
Loc_WSS_4	108,58	103,03	291,49	351,1	537,72
Loc_WSS_5	79,43	77,56	236,31	248,1	380,69
Loc_WSS_6	115,54	116,45	310,67	369,7	565,62
Loc_WSS_7	112,46	110,14	303,58	357,1	546,03
Loc_WSS_8	105,94	104,10	263,10	379,5	581,45

TAB. 3.3 – Variance du signal de sous-bande x_h pour différentes décompositions en *lifting scheme*. Les signaux d'entrée Loc_WSS_1 à Loc_WSS_8 sont décrits au § 3.6.1.

	GAE	LAE	Gerek <i>et al.</i>	(9,7)	(5,3)
Loc_WSS_1	5,55	5,54	6,23	6,40	6,70
Loc_WSS_2	5,51	5,46	6,22	6,36	6,68
Loc_WSS_3	5,61	5,60	6,27	6,47	6,77
Loc_WSS_4	5,42	5,38	6,13	6,26	6,56
Loc_WSS_5	5,14	5,12	5,89	5,97	6,28
Loc_WSS_6	5,46	5,44	6,17	6,29	6,60
Loc_WSS_7	5,44	5,43	6,15	6,27	6,58
Loc_WSS_8	5,41	5,39	6,06	6,32	6,63
moyenne	5,44	5,42	6,14	6,29	6,60

TAB. 3.4 – Entropie d'ordre 1 du signal de la sous-bande x_h pour différentes décompositions en *lifting scheme*. Les signaux d'entrée Loc_WSS_1 à Loc_WSS_8 sont décrits au § 3.6.1.

3.7 Conclusion

Dans ce chapitre, nous avons généralisé la structure de bancs de filtres en *lifting scheme*, en introduisant un deuxième filtre prédicteur dans l'étape de prédiction, pour exploiter toute l'information disponible au décodeur. Le critère retenu pour ajuster les coefficients des filtres consiste à minimiser la variance de la sous-bande haute fréquence. Nous avons justifié le choix de ce critère pour des signaux gaussiens et le codage sans perte. Deux méthodes d'adaptation ont été proposées. La première (GAE) suppose une stationnarité au sens large du signal d'entrée sur la totalité de son horizon d'observation. La seconde (LAE) ne suppose qu'une stationnarité (au sens large) locale. Dans ce dernier cas, les filtres prédicteurs sont adaptatifs. Nous avons montré sur des signaux synthétiques gaussiens à stationnarités locale et globale, que les nouvelles méthodes offrent en moyenne un gain de codage de 0,5 bpp (GAE) et 0,8 bpp (LAE) par rapport à la décomposition en ondelette (9,7), de 0,8 bpp (GAE) et 1,11 bpp (LAE) par rapport à la (5,3) et de 0,41 bpp (GAE) et 0,65 bpp (LAE) par rapport à la méthode de Gerek et Çetin [46].

Au chapitre suivant nous évaluons les performances des décompositions GAE et

LAE sur des images réelles.

CHAPITRE 4

Application à la compression sans perte des images réelles

4.1 Introduction

Dans ce chapitre, nous adoptons les notations de la figure 4.1 à la place de celles de la figure 3.5. Cela revient à remplacer partout le symbole x par la lettre I (pour image). De plus, la fonction de transfert du quadripole ayant pour entrées I_1, I_2 et pour sorties I_ℓ, I_h est notée $E_1(z_1, z_2)$. Nous nous intéressons aux résultats de codage sans perte obtenus par les méthodes GAE et LAE sur des images réelles. Nous étudions quatre classes d'images, toutes codées sur 8 bits (256 niveaux de gris) : naturelles, IRM médicales, satellitaires¹, et enfin empreintes digitales et textures. L'annexe B donne le descriptif de ces images. Nous commençons par décrire l'ajustement des paramètres (ordres des filtres, facteur d'oubli,...) que nous avons choisi par famille d'images.

4.2 Ajustement des paramètres par famille d'images

4.2.1 La méthode GAE

Les performances de la décomposition dépendent de paramètres, comme le type de sous-échantillonnage utilisé et les ordres des filtres prédicteurs.

4.2.1.1 Choix du type de sous-échantillonnage

Nous avons implanté et testé les deux types de sous-échantillonnage pour la décomposition polyphase, séparable et en quinconce. Le meilleur des deux dépend

¹Parmi les images satellitaires les images *Sanfransisco, Okland et Moissac* sont à la base des images en trois composantes RVB. Chacune des ces dernières est codée sur 8 bits. Les versions niveaux de gris de ces images sont générées en calculant la moyenne des trois composantes. Les images satellitaires ont été fournies gracieusement par le Centre National d'Études Spatiales (CNES) et la société SPOT Image.

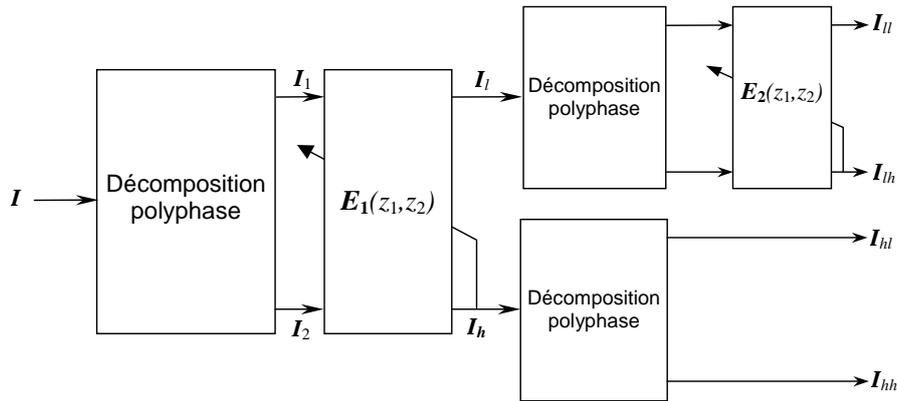


FIG. 4.1 – Un niveau de décomposition d’une image avec le lifting scheme généralisé adapté.

de l’image. Nous avons donc gardé les deux types et aucun n’est significativement plus performant que l’autre.

4.2.1.2 Choix des ordres des filtres

Le choix des ordres p et q des filtres $A(z_1, z_2)$ et $B(z_1, z_2)$ dépend de l’intercorrélation entre I_1 et I_2 d’une part, et de l’autocorrélation de I_2 d’autre part. Pour étudier le comportement de la méthode GAE, nous avons tracé les coefficients de corrélation entre les différentes sous-bandes, pour chaque niveau de décomposition, en faisant varier l’ordre du filtre.

La fonction d’intercorrélation (ou d’intercovariance)

$$\gamma_{I_1 I_2}(u, v) = E[I_1(m + u, n + v)I_2(m, n)] \quad (4.1)$$

est estimée avec l’estimateur biaisé

$$\hat{\gamma}_{I_1 I_2}(u, v) = \frac{1}{MN} \sum_{m=1}^{N-|u|} \sum_{n=1}^{M-|v|} I_1(m + |u|, n + |v|)I_2(m, n) \quad (4.2)$$

où les images I_1, I_2 sont de taille $M \times N$ (si l’une d’elles est plus petite, elle sera complétée par des zéros) et préalablement centrées. Il en est de même pour la fonction d’autocorrélation (où $I_1=I_2$). Les éléments de la matrice des coefficients de corrélation estimés

$$\hat{\rho}_{I_1 I_2}(u, v) = \frac{\hat{\gamma}_{I_1 I_2}(u, v)}{\sqrt{\hat{\gamma}_{I_1 I_2}(0, 0)\hat{\gamma}_{I_1 I_2}(0, 0)}} \quad (4.3)$$

sont tracés pour $-20 \leq u \leq 20$, et les 41 courbes obtenues quand $-20 \leq v \leq 20$ sont superposées dans un même graphe. Afin de faciliter l’analyse des courbes, nous donnons dans le schéma suivant les notations pour chaque étape d’un niveau de décomposition. Remarquons qu’ici, contrairement à la figure 4.1, une étape “predict” est appliquée entre I_{h1} et I_{h2} pour obtenir I_{hh} et I_{hl}

$$I \xrightarrow{\text{decomp.poly.}} I_1, I_2 \xrightarrow{\text{predict.}} \begin{array}{l} I_\ell \xrightarrow{\text{decomp.poly.}} I_{\ell 1}, I_{\ell 2} \xrightarrow{\text{predict.}} I_{\ell \ell}, I_{\ell h} \\ I_h \xrightarrow{\text{decomp.poly.}} I_{h1}, I_{h2} \xrightarrow{\text{predict.}} I_{hh}, I_{hl} \end{array}$$

En analysant les courbes obtenues, et comme attendu, nous constatons que l'étape de prédiction décorrèle les composantes polyphases de l'image. Cela se voit sur la première colonne de la figure 4.2 où sont tracées (du haut vers le bas) les coefficients de corrélation entre I_1 et I_2 , puis entre I_ℓ et I_h pour des filtres d'ordres croissants (1, 1), (2, 2) et (3, 3). Lorsque l'ordre du filtre augmente la corrélation commence par diminuer, puis se stabilise à partir d'un certain ordre, qui dépend de la résolution de I (généralement (3, 3) pour une image I de dimension 512×512). Il est donc inutile de choisir des ordres supérieurs à cette valeur. Il en est de même pour la corrélation entre pixel de I_2 , qui devient I_h après l'étape de prédiction. Elle commence par diminuer lorsque la prédiction est appliquée, et à partir d'une certaine valeur de l'ordre du filtre, elle ne décroît plus, comme le montre la seconde colonne de la figure 4.2.

Lors de la décomposition de I_ℓ ($I_\ell \rightarrow I_{\ell 1}, I_{\ell 2} \rightarrow I_{\ell \ell}, I_{\ell h}$), le même comportement est noté, à la différence près que l'ordre optimal est plus faible (généralement (1, 1) pour une image I de dimension 512×512). Lorsque la sous-bande I_h est décomposée ($I_h \rightarrow I_{h1}, I_{h2} \rightarrow I_{h\ell}, I_{hh}$), la diminution de corrélation due à l'étape de prédiction n'est pas significative, comme le montre la figure 4.3. Et ceci, quelle que soit la résolution de I . Cela est dû au fait qu'à l'issue de la première décomposition ($I \rightarrow I_1, I_2 \rightarrow I_\ell, I_h$) les pixels de l'images I_h sont déjà décorrélés. Par conséquent, comme nous l'avons déjà dit, on peut s'affranchir de l'étape de prédiction dans la décomposition de I_h et ne garder qu'un simple sous-échantillonnage.

Dans une décomposition multi-résolutions, au fur et à mesure que la taille de l'image d'approximation diminue, les pixels voisins dans cette dernière sont de moins en moins corrélés, car de plus en plus éloignés dans l'image d'origine. Cela se voit en comparant les premières lignes des figures 4.2 et 4.4.

Lorsque la taille de l'image d'approximation devient petite, le prédicteur ne fournit pas une bonne estimation, car il n'y a pas assez d'échantillons pour estimer les moments d'ordre deux. Dans ce cas il vaut mieux utiliser une autre technique comme, par exemple, un prédicteur à coefficients fixes ou simplement la transformation S [92]. Nous verrons plus loin quelle stratégie adopter lorsque la taille de l'image d'approximation est petite. Pour des ordres de filtre supérieurs à (2, 2), la corrélation entre I_ℓ et I_h reste presque inchangée (figure 4.4) sur l'image d'approximation de résolution 256×256 . Enfin, pour les images d'approximation de résolutions inférieures ou égales à 128×128 , un filtre d'ordre (1, 1) est suffisant comme le montre la figure 4.4. Pour exploiter ce résultat, nous prendrons des ordres de filtres de plus en plus petits au fur et à mesure que le niveau de la décomposition augmentera.

De plus, nous avons calculé l'entropie du premier ordre moyenne de l'image transformée en fonction de l'ordre du filtre, et nous avons remarqué que celui qui donne la plus petite entropie varie d'une image à l'autre. Mais pour une même classe d'images, l'ordre optimal est généralement le même. Pour les images naturelles par exemple, l'ordre (2, 2) est un bon compromis entre la performance et la complexité. Pour les images IRM, qui sont lisses, l'ordre (1, 1) est le meilleur choix.

4.2.1.3 Nombre de niveaux de décomposition

Pour disposer de suffisamment d'échantillons pour estimer les moments d'ordre deux dans les sous-bandes, nous imposons à celle de plus basse résolution d'être de

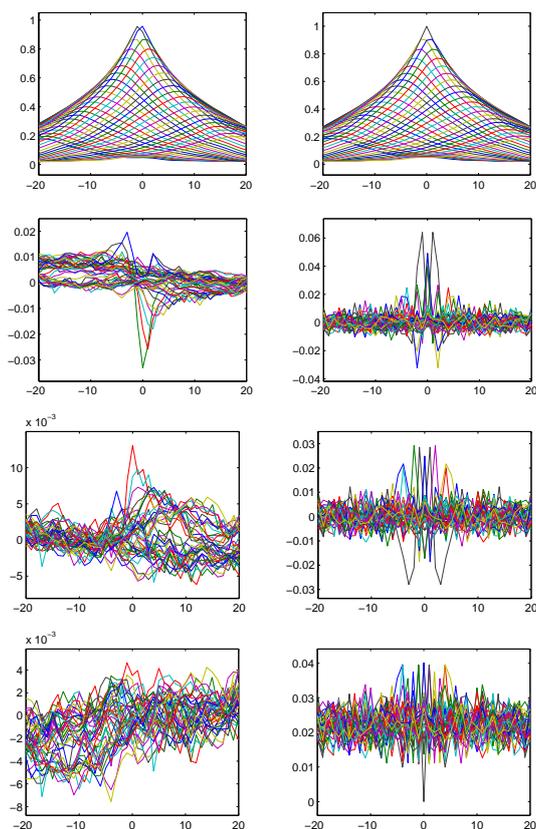


FIG. 4.2 – De gauche à droite et de haut en bas : coefficients de corrélation entre I_1 et I_2 , coefficients de corrélation entre pixels de I_2 , coefficients de corrélation entre I_ℓ et I_h , coefficients de corrélation entre pixels de I_h (la valeur 1 est omise) pour des filtres d'ordre $(1, 1)$, $(2, 2)$, $(3, 3)$ appliqués à l'image Lena 512×512 .

dimension 32×32

4.2.1.4 Utilisation de la transformation S (décomposition SGAE)

L'opération de sous-échantillonnage seule (*i.e.*, associée à aucun filtre passe-bas), pour la construction de la sous-bande d'approximation, introduit des artefacts (dus au recouvrement du spectre, les hypothèses du théorème d'échantillonnage de Shannon n'étant plus satisfaites), qui deviennent gênants quand la résolution de l'image d'approximation descend en dessous d'une valeur (de l'ordre de 128×128 à 256×256 suivant la classe d'images). Pour permettre un codage progressif en résolution, nous avons choisi d'utiliser le filtre passe bas à coefficients fixes de la transformation S comme étape "Update" du *lifting scheme* généralisé (voir la figure 4.5).

La transformation S est simple et peut être calculée d'une manière très optimisée. Elle est donnée par les équations suivantes [28]

$$I_h^{(1)}(i, j) = I_2(i, j) - I_1(i, j) \quad (4.4)$$

$$I_\ell(i, j) = I_1 + \lfloor I_h^{(1)}(i, j)/2 \rfloor. \quad (4.5)$$

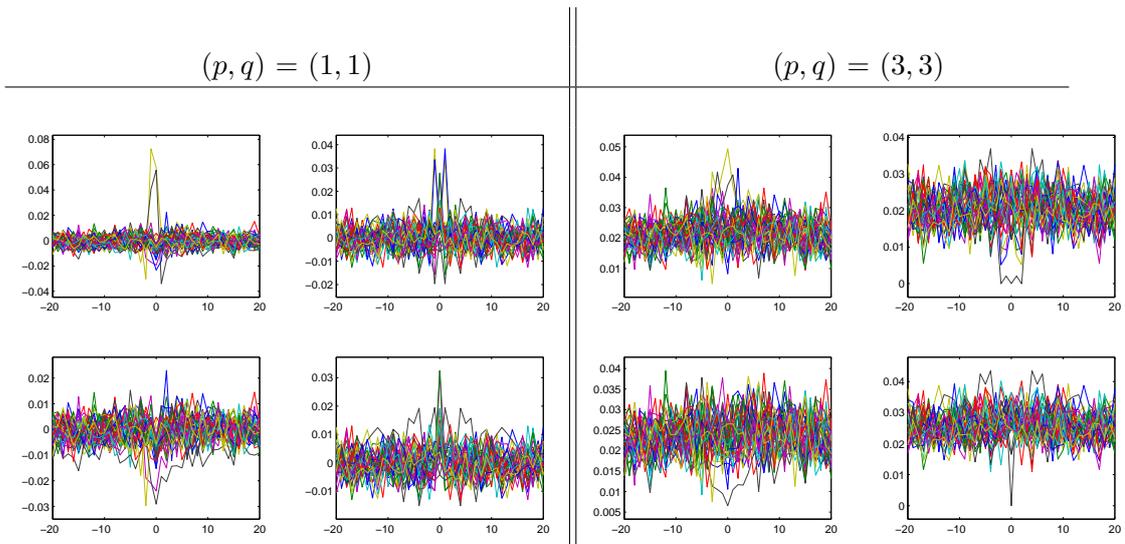


FIG. 4.3 – De gauche à droite et de haut en bas : coefficients de corrélation entre I_{h1} et I_{h2} , coefficients de corrélation entre pixels de I_{h2} (la valeur 1 est omise), coefficients de corrélation entre I_{hl} et I_{hh} et coefficients de corrélation entre pixels de I_{hh} (on a enlevé la valeur 1), pour Lena 512×512 et $(p, q) = (1, 1)$, $(3, 3)$.

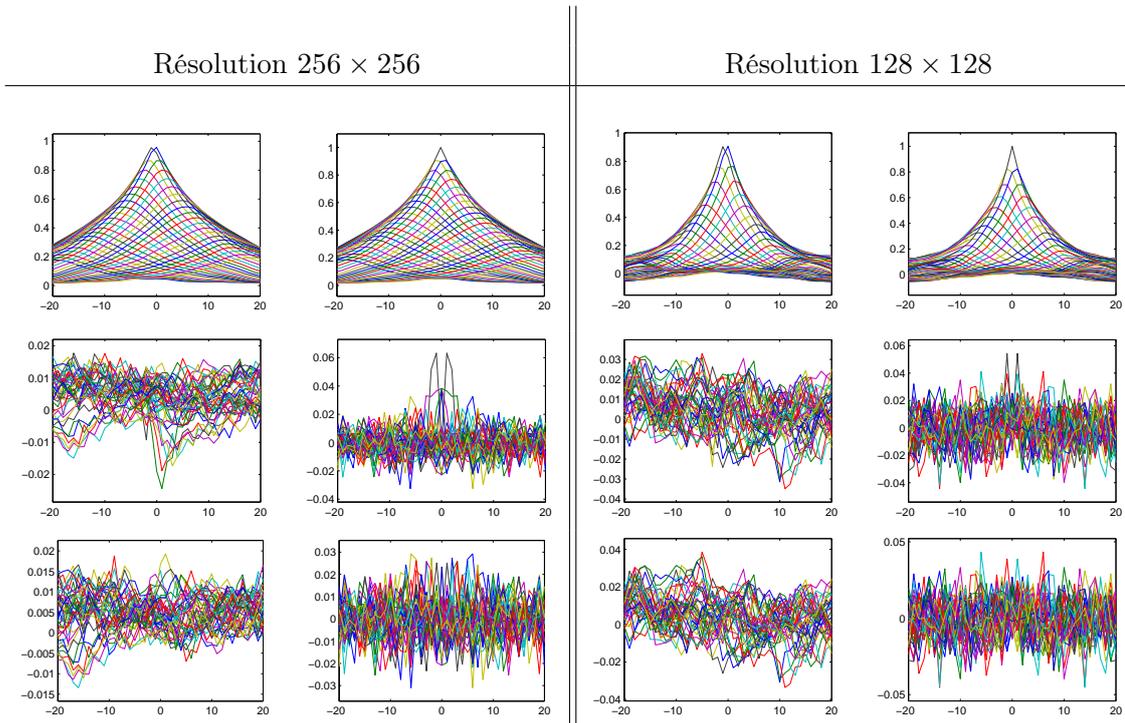
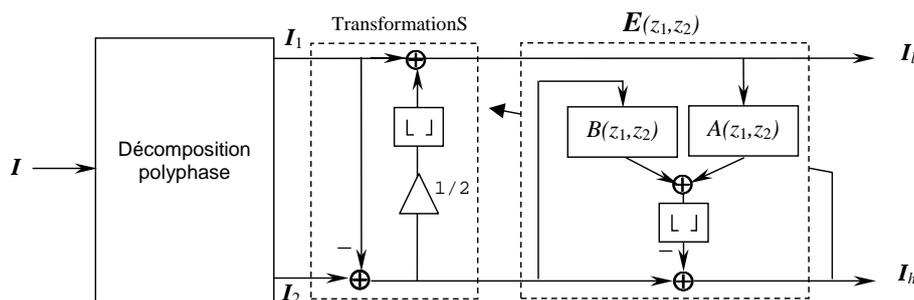


FIG. 4.4 – De gauche à droite et de haut en bas : coefficients de corrélation entre I_1 et I_2 , coefficients de corrélation entre pixels de I_2 , coefficients de corrélation entre I_ℓ et I_h , coefficients de corrélation entre pixels de I_h (la valeur 1 est omise) pour des filtres d'ordre $(1, 1)$ et $(2, 2)$ appliqués aux images de sous-bande d'approximation de résolution 256×256 et 128×128 de l'image Lena.

FIG. 4.5 – Utilisation de la transformation S .

L'image de sous-bande I_h est alors donnée par

$$I_h(i, j) = I_h^{(1)}(i, j) - \left[\sum_{(h,k) \in \Delta_1} a_{hk} I_l(i-h, j-k) + \sum_{(h,k) \in \Delta_2} b_{hk} I_h^{(1)}(i-h, j-k) + 1/2 \right] \quad (4.6)$$

et celle d'approximation par l'équation (4.5). Cette décomposition sera appelée SGAE dans la suite.

En analysant les résultats du tableau 4.1, nous constatons que la méthode SGAE donne en général une entropie moyenne légèrement inférieure à celle de la méthode GAE. Cela peut s'expliquer par le fait qu'il y a plus de corrélation entre les pixels de l'image d'approximation pour la méthode SGAE. Cette corrélation rend plus efficace l'étape de prédiction et par conséquent le codage.

4.2.1.5 La méthode GAE améliorée (GAEa)

Nous avons constaté sur les figures 4.2 et 4.4, que plus le niveau de décomposition augmente, moins les pixels de l'image de la sous-bande basse résolution sont corrélés entre eux et l'adaptation des coefficients des filtres $A(z_1, z_2)$ et $B(z_1, z_2)$ devient moins efficace. C'est également ce que nous avons observé sur les estimations de débit par l'entropie d'ordre 1 moyenne. Donc, pour améliorer la méthode GAE, nous l'avons modifiée de la façon suivante : les ordres (p, q) des filtres $A(z_1, z_2)$ et $B(z_1, z_2)$ sont adaptés à la résolution de la sous-bande et à la classe de l'image². Pour les sous-bandes d'approximation de petites dimensions (inférieures à 64×64), l'estimation \hat{I}_2 de I_2 est une approximation non-linéaire de $I_1(i, j)$, obtenue sans filtrage par un procédé voisin³ de la méthode LAR (*Locally Adaptive Resolution*) introduite par

²Plus précisément, pour chaque classe d'images et pour chaque niveau de décomposition, les valeurs des paramètres (ordres des filtres, type de sous-échantillonnage, application ou non de la transformation S) sont optimisées en choisissant dans un dictionnaire (qui dépend de la classe et du niveau de décomposition) celles qui minimisent la taille du flot de bits (obtenu avec le codeur arithmétique adaptatif proposé par Said et Pearlman sur le site [1]).

³Ce procédé consiste, après la décomposition polyphase appliquée à I (attention, ici l'image I décomposée est une sous-bande basse résolution de petite dimension de l'image à coder), à faire une partition de la sous-bande I_1 en zones homogènes. Cette partition permet de construire l'estimation \hat{I}_2 en donnant pour valeur commune aux pixels d'une même zone homogène, la valeur moyenne des pixels de I_1 appartenant à la zone en question.

Image/(p, q)	(1,1)		(2,2)		(3,3)	
	GAE	SGAE	GAE	SGAE	GAE	SGAE
Lena	4,37	4,34	4,36	4,33	4,37	4,33
Goldhill	4,85	4,84	4,83	4,82	4,82	4,81
Barbara	4,92	4,81	4,81	4,73	4,76	4,69
F16	4,19	4,18	4,19	4,18	4,20	4,18
Mandrill	6,15	6,09	6,13	6,07	6,13	6,07
Peppers	4,67	4,59	4,65	4,59	4,64	4,58
IRM1	2,73	3,23	2,93	3,16	3,03	3,23
IRM2	2,88	3,27	2,86	3,13	2,78	3,00
IRM3	5,04	5,01	4,97	4,94	4,91	4,88
IRM4	4,45	4,51	4,43	4,46	4,43	4,45
IRM5	4,29	4,30	4,40	4,35	4,40	4,35
IRM6	3,36	2,86	2,49	2,71	3,47	3,71
Pentagone	5,30	5,27	5,30	5,27	5,30	5,27
Sanfrancisco	4,89	4,86	4,88	4,85	4,88	4,85
Okland	4,38	4,36	4,37	4,36	4,37	4,35
Toul	5,09	5,10	5,05	5,06	5,05	5,06
Gènes	4,22	4,26	4,21	4,24	4,21	4,24
Airplane	4,47	4,49	4,46	4,48	4,46	4,48
Airport	5,24	5,25	5,23	5,22	5,22	5,21
Finger1	4,50	4,49	4,37	4,36	4,34	4,33
Finger2	4,00	3,99	3,93	3,91	3,91	3,89
Finger3	4,49	4,49	4,40	4,39	4,37	4,36

TAB. 4.1 – Comparaison des méthode GAE et SGAE : entropie du premier ordre moyenne de l'image transformée en fonction de l'ordre (p, q) des filtres A et B , dans le cas d'un sous-échantillonnage séparable.

Déforges et Ronsin [42]. Nous avons observé qu'en général, avec ce procédé, il est intéressant d'augmenter le nombre de niveaux de décompositions jusqu'à des sous-bandes de dimension 16×16 ou 32×32 . Par ailleurs, ayant remarqué que l'application de la transformation S n'améliore pas le codage quand la résolution de la sous-bande est supérieure à 256×256 , nous ne l'appliquons pas aux sous-bandes dont la dimension dépasse 256×256 . Dans la perspective d'un codage progressif en résolution, nous avons observé également que les artefacts dus au repliement de spectre ne sont pas gênants à l'œil, tant que la résolution de l'image d'approximation est supérieure (voire égale) à 256×256 . La version améliorée de la méthode GAE sera notée GAEa dans la suite.

4.2.1.6 En-tête du flot de bits

Au niveau du décodeur, un certain nombre de paramètres, à savoir, les dimensions N , M de l'image I , les ordres des filtres, leurs coefficients, la moyenne de l'image

originale, le nombre de niveaux de décomposition, un booléen précisant le type de sous-échantillonnage employé (séparable ou en quinconce) et un autre indiquant si la transformation S a été utilisée ou non, doivent être connus⁴ pour reconstruire l'image. Ces paramètres sont transmis comme en-tête dans le flot de bits. Pour réduire la taille mémoire nécessaire au codage de l'en-tête, nous avons tronqué les coefficients des filtres $A(z_1, z_2)$ et $B(z_1, z_2)$ (avant d'appliquer la décomposition) en ne gardant que quelques digits après la virgule : un coefficient a est remplacé par

$$\hat{a} = \lfloor a * 10^S + 0,5 \rfloor / 10^S \quad (4.7)$$

(dans nos expérimentations, S vaut 6). La taille de l'en-tête est négligeable par rapport à la taille du flot de bits total.

4.2.2 La méthode LAE

Dans ce paragraphe, nous comparons les performances en codage de l'algorithmes des moindres carrés adaptatifs à celles du gradient stochastique normalisé pour l'adaptation des filtres. De plus, pour le traitement au bord de l'image, différents prolongements sont étudiés, ainsi que l'influence de divers paramètres, comme les ordres r_1 et r_2 des filtres.

4.2.3 Choix de l'algorithme d'adaptation

Pour du codage sans perte, nous comparons l'algorithme des moindres carrés adaptatifs (MCA) à celui du gradient stochastique normalisé (GSN). Le facteur d'oubli α de l'algorithme des moindres carrés adaptatifs est fixé à 0,9995 (nous justifierons au § 4.2.3.4 le choix de cette valeur), et le paramètre μ de l'algorithme du gradient normalisé est adaptatif, sa valeur variant en fonction de la différence entre les valeurs, maximale et minimale, du vecteur observation [47]. La figure 4.6 montre la variation de l'erreur quadratique moyenne courante, calculée suivant la relation

$$\text{EQM}(i, j) = \frac{1}{M_2(i-1) + j} \left(\sum_{k=1}^{i-1} \sum_{l=1}^{M_2} |I_2(k, l) - \hat{I}_2(k, l)|^2 + \sum_{l=1}^j |I_2(i, l) - \hat{I}_2(i, l)|^2 \right) \quad (4.8)$$

où M_2 est le nombre de colonnes de I_2 . Autrement dit, en introduisant le nombre d'itérations $n = M_2(i-1) + j$ et le signal mono-dimensionnel x (resp. \hat{x}) obtenu en parcourant l'image I_2 (resp. \hat{I}_2) de gauche à droite et de haut en bas : pour tous k, ℓ tels que $1 \leq k \leq N_2$ et $1 \leq \ell \leq M_2$,

$$x[M_2(k-1) + \ell] = I_2(k, \ell) \quad (\text{resp. } \hat{x}[M_2(k-1) + \ell] = \hat{I}_2(k, \ell)), \quad (4.9)$$

l'équation (4.8) devient

$$\text{EQM}(n) = \frac{1}{n} \sum_{k=1}^n |x(k) - \hat{x}(k)|^2, \quad (4.10)$$

⁴Pour la version « améliorée » GAEEa décrite ci-dessus, d'autres paramètres sont mis dans l'en-tête, pour réduire la taille mémoire nécessaire à son codage, sachant que le décodeur dispose des mêmes dictionnaires (voir la note de bas de page 2) que le codeur.

et le graphe de la fonction $EQM(n)$ est donné à la figure 4.6. Ce graphe montre d'une part qu'en régime permanent MCA donne une EQM 60% (environ) plus petite que GSN et d'autre part que MCA converge beaucoup plus vite que GSN. Cette dernière propriété est importante, notamment quand il s'agit de signaux non stationnaires, tels que les images. Cela se traduit par un nombre minimal d'échantillons de grande énergie à proximité des contours, et donc par une réduction du coût de codage. En

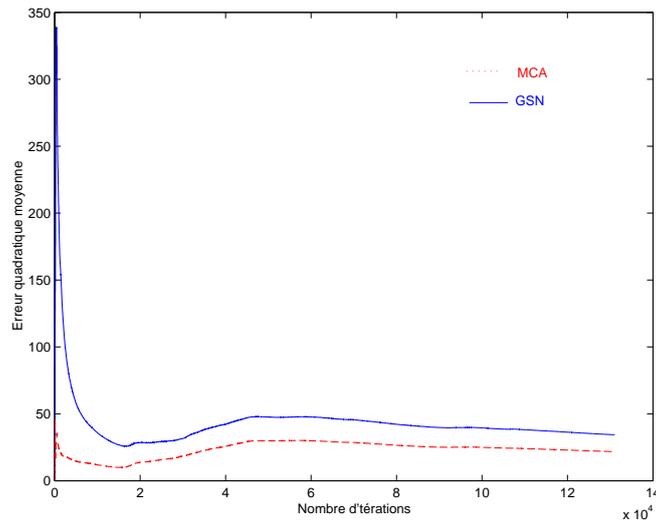


FIG. 4.6 – Erreur quadratique moyenne de prédiction, donnée à la relation (4.10) d'un signal de 2^{17} échantillons, pour l'algorithme (MCA) ($\lambda = 0,9995$) et pour l'algorithme (GSN) (μ est adaptatif). Les signaux tests sont issus de l'image Lena.

outre, les performances de codage dépendent de la variance de l'image d'erreur I_h .

La figure 4.7 montre la variance locale de l'erreur d'estimation, définie par la relation

$$\text{Var}(n) = \frac{1}{15} \sum_{k=-7}^7 |x(n+k) - \hat{x}(n+k)|^2, \quad (4.11)$$

pour les algorithmes MCA (courbe rouge) et GSN (courbe bleue). Nous remarquons que la variance locale est plus petite pour le premier algorithme et qu'autour des variations brusques (contours), elle décroît plus rapidement avec l'algorithme MCA.

Dans le tableau 4.2 nous avons reporté la variance globale (définie page vii) et l'entropie du premier ordre de l'image résiduelle I_h , pour six images naturelles. La variance et l'entropie sont plus petites avec l'algorithme MCA avec un gain d'environ 0,3 bpp quelle que soit l'image.

Une comparaison qualitative est présentée à la figure 4.8, montrant l'image à estimer I_2 et l'image de détails I_h pour les deux algorithmes. Cette image correspond au plan de bits de plus fort poids de Lena. La comparaison qualitative porte sur une image binaire pour avoir des contours nets. En faisant un zoom, on peut voir qu'il y a plus de coefficients non nuls autour des contours avec l'algorithme du gradient normalisé.

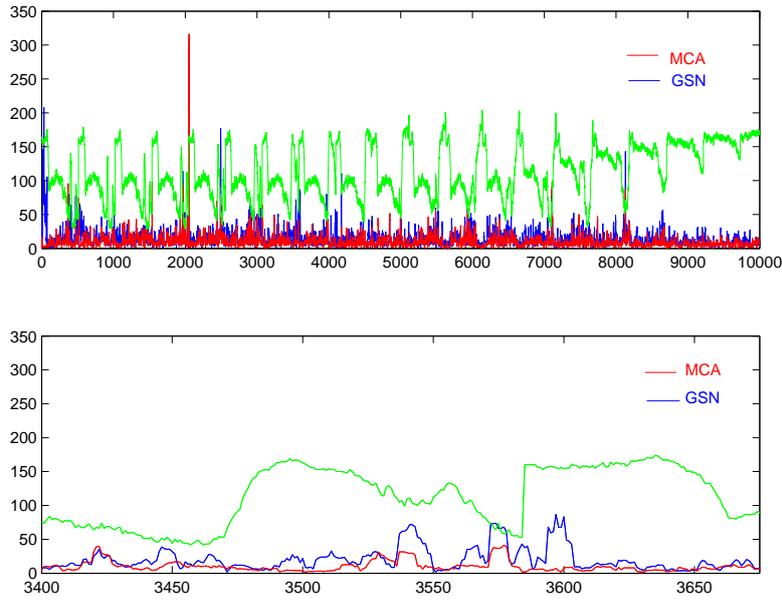


FIG. 4.7 – Haut : variance locale de l'erreur d'estimation pour l'algorithme MCA (en rouge) et l'algorithme GSN (en bleu). La courbe en vert représente le signal à estimer. Bas : zoom sur une partie de la courbe du haut, montrant le comportement des deux algorithmes autour de variations brusques du signal à estimer.

image	Entropie		Variance	
	GSN	MCA	GSN	MCA
Lena	4,45	4,17	34,42	21,79
Goldhill	4,90	4,57	61,88	36,67
Barbara	4,93	4,59	74,33	42,41
F16	4,23	3,91	37,53	21,12
Mandrill	6,02	5,78	305,69	202,34
Peppers	4,83	4,54	77,79	37,88
Library	6,37	6,14	578,11	547,45

TAB. 4.2 – Entropie du premier ordre (bpp) et variance de l'image résiduelle I_h en utilisant l'algorithme GSN et l'algorithme MCA.

4.2.3.1 Traitement des bords

Comme nous l'avons déjà dit, trois type d'extension sont possibles : prolongement par des zéros, par symétrie miroir ou par périodisation. Pour quatre des six images testées (voir le tableau 4.3), la variance et l'entropie du premier ordre de l'image résiduelle I_h sont plus petites quand on emploie un prolongement symétrique (la différence avec les autres extensions est généralement légère). L'extension par des zéros est celle qui donne les plus grandes variances et entropies du premier ordre.

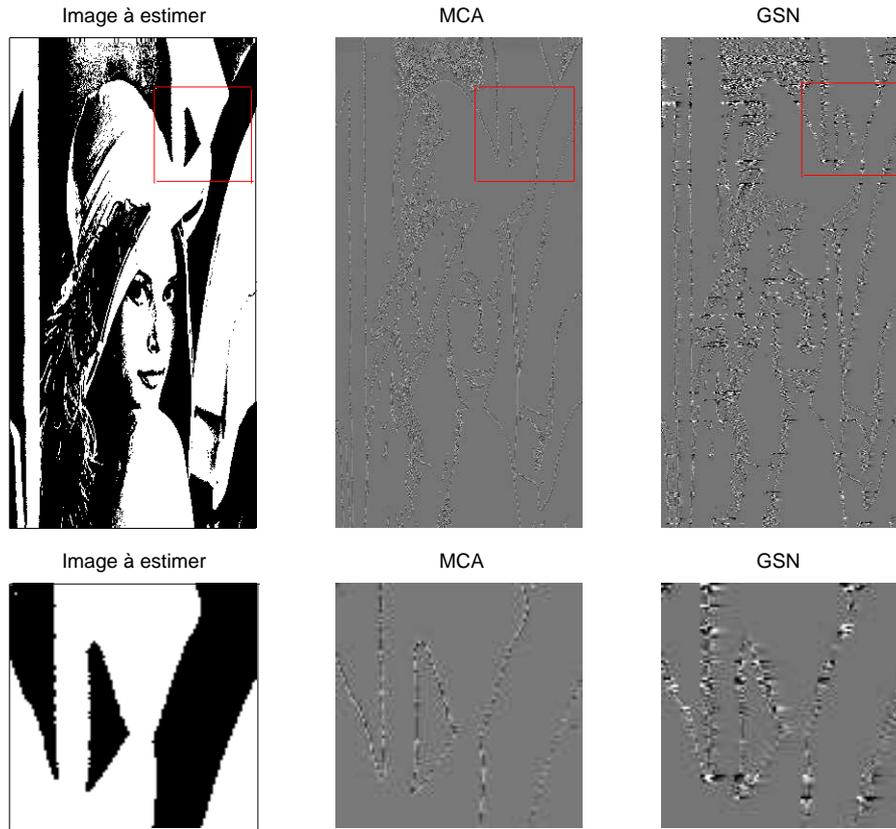


FIG. 4.8 – Gauche : image I_2 à estimer. Milieu : image de détails I_h , en utilisant l'algorithme MCA. Droite : même sous-bande I_h avec l'algorithme GSN. Ligne du bas : zoom

Image	Variance			Entropie		
	Extension			Extension		
	sym.	périod.	zéros	sym.	périod.	zéros
Lena	21,47	21,51	23,33	4,11	4,11	4,17
Goldhill	36,03	36,07	37,51	4,51	4,52	4,54
Barbara	41,13	41,17	42,62	4,56	4,56	4,59
F16	19,49	19,63	15,17	3,85	3,85	3,72
Mandrill	202,75	210,82	209,58	5,73	5,73	5,76
Peppers	38,22	38,23	33,45	4,49	4,50	4,42

TAB. 4.3 – Entropie du premier ordre et variance de l'image résiduelle I_h , avec l'algorithme MCA, pour les trois types d'extension (symétrique, périodique et par des zéros).

4.2.3.2 Choix du type de sous-échantillonnage

Dans le tableau 4.4 nous avons donné l'entropie moyenne et la variance de l'image transformée par la méthode LAE. Selon ces résultats, le sous-échantillonnage séparable est toujours plus performant que le sous-échantillonnage en quinconce avec parfois des gains supérieurs à 0,5 bpp. Nous avons donc opté pour un sous-échantillonnage toujours séparable dans la méthode LAE.

Image	Variance		Entropie moyenne	
	Sous-échantillonnage		Sous-échantillonnage	
	sép.	quinc.	sép.	quinc.
Lena	71,63	96,52	4,32	4,49
Goldhill	88,92	225,92	4,80	4,96
Barbara	128,77	178,23	4,71	4,85
F16	110,71	163,89	4,20	4,39
Mandrill	439,18	502,38	6,10	6,25
Peppers	104,81	304,56	4,66	4,75
IRM1	396,22	2626,96	3,05	3,51
IRM2	101,79	138,21	2,98	3,56
IRM3	143,68	189,29	5,05	5,08
IRM4	282,81	812,39	4,37	4,42
IRM5	356,18	1590,02	4,00	4,59
IRM6	186,91	525,43	2,35	2,65
Pentagone	126,53	160,98	5,26	5,39
Sanfrancisco	74,97	133,93	4,84	4,99
Okland	40,98	48,31	4,36	4,49
Toulouse	177,53	204,34	5,08	5,38
Genes	81,56	114,68	4,22	4,55
Airplane	37,11	38,80	4,49	4,52
Airport	174,49	168,22	5,29	5,41
Finger1	145,85	213,10	4,33	4,87
Finger2	91,57	129,51	3,89	4,28
Finger3	119,35	213,16	4,36	4,81

TAB. 4.4 – Entropie moyenne (en bpp) et variance de l'image décomposée (méthode LAE), pour les deux types de sous-échantillonnage séparable (sep.) et quinconce (quinc.). Les paramètres de la décomposition sont $\alpha = 0,9995$ et $(r_1, r_2) = (16, 8)$.

4.2.3.3 Choix de l'ordre des filtres prédicteurs

Nous commençons par étudier l'influence du 2^{ème} filtre B dans l'étape de prédiction, puis nous recherchons les ordres (r_1, r_2) optimaux. L'introduction du deuxième filtre $B(z_1, z_2)$ améliore les performances de la prédiction. La variance et l'entropie du premier ordre de l'image résiduelle I_h , avec ou sans le filtre $B(z_1, z_2)$ sont données au tableau 4.5, pour quelques images tests. Voir également la figure 4.9 qui donne la taille du flot de bits (en utilisant le codeur entropique⁵ de Said et Pearlman) pour l'image *Finger1* avec ou sans le filtre $B(z_1, z_2)$.

Les valeurs optimales des ordres dépendent de l'intercorrélation entre le paramètre à estimer ($I_2(i, j)$) et le vecteur observation ($\underline{Y}(i, j)$). Plusieurs travaux ont été effectués pour la sélection de l'ordre optimal du modèle d'estimation [10, 11, 65]. Les techniques utilisées sont très gourmandes en temps de calcul [13]. Dans [13], Ak-

⁵Il s'agit du codeur arithmétique adaptatif de Said et Pearlman, disponible sur le site [1].

image/filtres	Variance		Entropie	
	A et B	A	A et B	A
Lena	21,47	36,12	4,11	4,30
Goldhill	36,03	66,42	4,51	4,80
Barbara	41,13	173,17	4,56	5,40
F16	19,49	80,74	3,85	4,01
Mandrill	202,75	230,45	5,73	5,82
Peppers	38,22	81,69	4,49	4,48

TAB. 4.5 – Entropie du premier ordre (en bpp) et variance de l'image résiduelle I_h avec l'algorithme MCA, dans les cas de l'introduction du 2^{ème} filtre $B(z_1, z_2)$ ou non dans la prédiction.

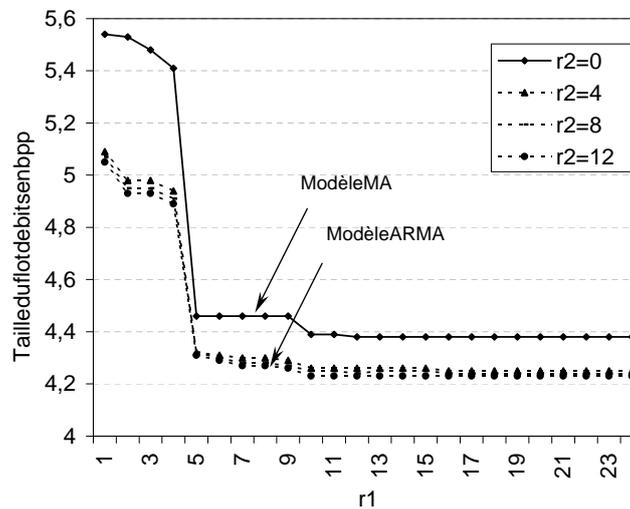


FIG. 4.9 – Taille du flot de bits (en bpp) après codage entropique en fonction de l'ordre du filtre, pour l'image Finger1. Modèle MA : sans filtre B, modèle ARMA : avec filtre B.

sasse *et al.* proposent une technique d'estimation de l'ordre d'un modèle AR-2D et traitent le cas d'une erreur d'estimation gaussienne. Suite aux expérimentations que nous avons faites sur des données réelles, nous avons constaté que l'ordre du filtre qui minimise la variance de l'erreur d'estimation varie sensiblement d'une classe d'images à l'autre, et légèrement entre les images d'une même classe. Nous avons mesuré la variance (définie, page vii) et l'entropie du premier ordre de l'image résiduelle I_h en faisant varier les ordres respectifs r_1 et r_2 des filtres $A(z_1, z_2)$ et $B(z_1, z_2)$. Les images testées sont des images naturelles (voir l'annexe B pour la description des images) et IRM médicales.

Les figures 4.10 et 4.11 montrent les graphes des grandeurs mesurées en fonction de r_1 et r_2 . Pour l'ensemble des images testées les comportements de la variance et de l'entropie d'ordre 1 de I_h en fonction de (r_1, r_2) sont similaires. Ces grandeurs décroissent lorsque r_1 et r_2 augmentent, jusqu'à atteindre des valeurs optimales, à partir de lesquelles les variations de la variance et de l'entropie sont négligeables.

Ces valeurs sont $(r_1, r_2) = (8, 4)$ pour les images naturelles et $(r_1, r_2) = (6, 2)$ pour les images IRM médicales.

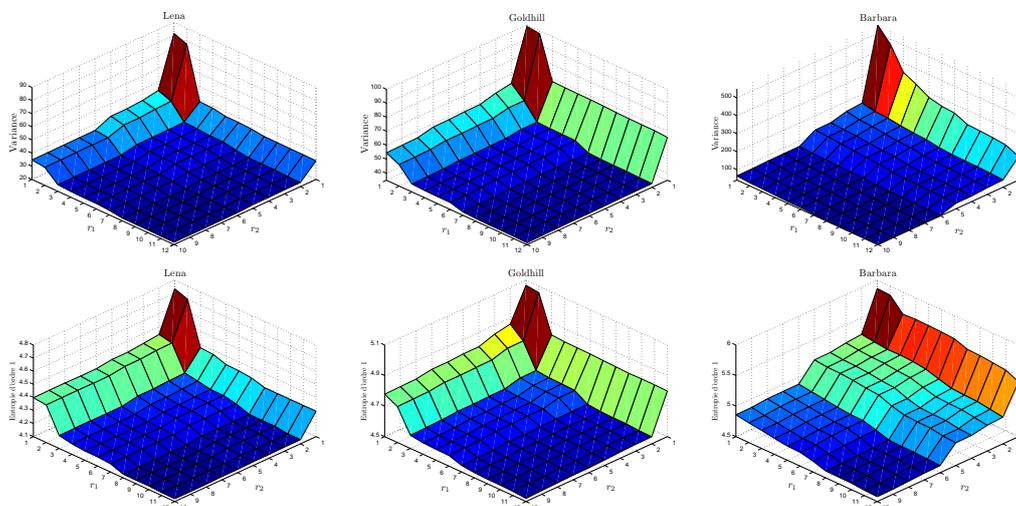


FIG. 4.10 – Variance et entropie d'ordre 1 de l'image résiduelle I_h en fonction des nombres de coefficients r_1 et r_2 des filtres A et B pour des images naturelles. Le facteur d'oubli vaut $\alpha = 0,9995$, l'extension est symétrique et le sous-échantillonnage séparable.

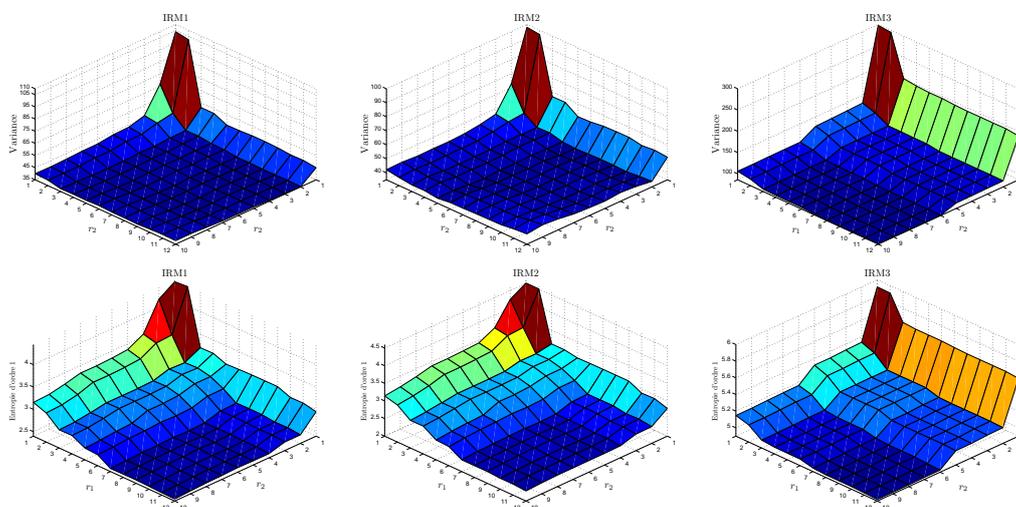


FIG. 4.11 – Même légende que la figure 4.10, pour des images médicales IRM.

La figure 4.9 montre les variations de la taille du flot de bits, obtenu avec un codeur arithmétique adaptatif, en fonction des ordres r_1 et r_2 pour l'image *Finger1*.

4.2.3.4 Choix du facteur d'oubli

L'introduction d'un facteur d'oubli α permet d'accorder un poids plus important aux échantillons plus proches, et ainsi de faire face à la dynamique du signal. Les tableaux 4.6 et 4.7 donnent l'entropie du premier ordre et la variance de l'image résiduelle I_h pour différentes images tests. Pour toutes ces images, la valeur optimale

du facteur d'oubli (celle qui donne la plus petite entropie du premier ordre de l'image résiduelle I_h) se situe entre 0,999 et 1. La figure 4.12 donne un exemple de variation de la variance et de l'entropie d'ordre 1 de I_h en fonction de α pour l'image Lena. Dans nos simulations nous utilisons la valeur de $\alpha = 0,9995$, qui est un bon compromis.

En posant $\alpha = 1 - \varepsilon$ avec $0 < \varepsilon \ll 1$, un développement limité à l'ordre 1 donne

$$\alpha^n = (1 - \varepsilon)^n \simeq 1 - n\varepsilon, \quad (4.12)$$

on peut donc considérer qu'après $n = 1/\varepsilon$ itérations, l'erreur de prédiction est complètement oubliée. Pour $\alpha = 0,9995$, on trouve $n = 2000$, ce qui correspond à quelques lignes (de 4 à 8) de l'image I_2 .

Par ailleurs, puisque les images sont généralement constituées de larges zones de textures séparées par des contours, le nombre de pixels appartenant aux contours, pour lesquels α doit avoir une faible valeur, est largement inférieur au nombre de pixels appartenant aux zones de textures, pour lesquelles α doit être proche de 1. Par exemple, dans l'image Lena, seulement 3,15% des pixels appartiennent aux contours. Il pourrait être intéressant d'adopter une stratégie d'adaptation du facteur d'oubli en fonction du fait que le pixel à estimer est dans une zone de contour ou une zone de texture. Il existe dans la littérature des travaux concernant l'utilisation d'un facteur d'oubli adaptatif [129], mais nous avons choisi un facteur fixe $\alpha = 0,9995$.

Image/ α	0,990	0,991	0,992	0,993	0,994	0,995	0,996	0,997	0,998	0,999	1
Lena	4,23	4,22	4,21	4,20	4,19	4,17	4,16	4,15	4,14	4,12	4,18
Goldhill	4,67	4,61	4,60	4,59	4,58	4,57	4,56	4,58	4,54	4,53	4,64
Barbara	4,63	4,63	4,62	4,61	4,6	4,59	4,58	4,57	4,56	4,56	4,92
F16	3,98	3,96	3,95	3,94	3,92	3,91	3,89	3,88	3,87	3,86	4,08
Mandrill	5,83	5,82	5,81	5,80	5,79	5,78	5,77	5,76	5,75	5,74	5,79
Peppers	4,61	4,59	4,58	4,57	4,56	4,54	4,52	4,51	4,50	4,49	4,55
IRM1	2,52	2,52	2,50	2,50	2,50	2,49	2,49	2,41	2,47	2,46	2,71
IRM2	2,38	2,37	2,36	2,35	2,34	2,33	2,32	2,31	2,29	2,28	2,51
IRM3	4,99	4,99	4,98	4,97	4,96	4,95	4,94	4,93	4,91	4,92	4,96
IRM4	4,40	4,31	4,38	4,36	4,34	4,33	4,31	4,29	4,26	4,23	4,34

TAB. 4.6 – Entropie du premier ordre de l'image résiduelle I_h en fonction du facteur d'oubli α . Le reste des paramètres de la décomposition est fixé à $r_1 = 16$, $r_2 = 8$, une extension symétrique et un sous-échantillonnage séparable.

Image/ α	0,990	0,991	0,992	0,993	0,994	0,995	0,996	0,997	0,998	0,999	1,000
Lena	26,39	25,82	25,25	24,73	24,21	23,71	23,21	22,71	22,25	21,87	26,69
Goldhill	41,91	41,31	40,74	40,14	39,57	38,99	38,41	37,86	37,32	36,82	46,59
Barbara	47,93	47,10	46,22	45,37	44,55	43,72	42,88	42,12	41,46	41,32	88,29
F16	26,38	25,57	24,88	24,21	23,56	22,99	22,44	21,93	21,48	21,15	33,40
Mandrill	229,86	226,76	223,62	220,52	217,47	314,36	211,31	208,30	205,33	202,84	217,13
Peppers	49,91	48,20	46,58	45,02	43,58	42,20	40,90	39,66	38,53	37,73	43,67
IRM1	57,68	56,16	54,48	52,91	50,99	49,03	46,86	44,55	42,10	39,54	45,53
IRM2	57,41	55,14	53,06	51,16	49,41	47,86	46,48	45,39	44,70	44,29	46,98
IRM3	101,09	99,16	97,20	95,40	93,62	92,01	90,55	89,24	88,23	87,46	98,71
IRM4	110,23	107,71	105,29	102,89	100,56	98,38	96,24	94,27	92,55	91,05	113,41

TAB. 4.7 – Variance de l'image résiduelle I_h en fonction du facteur d'oubli α . Les paramètres de la décomposition sont fixés comme pour le tableau 4.6.

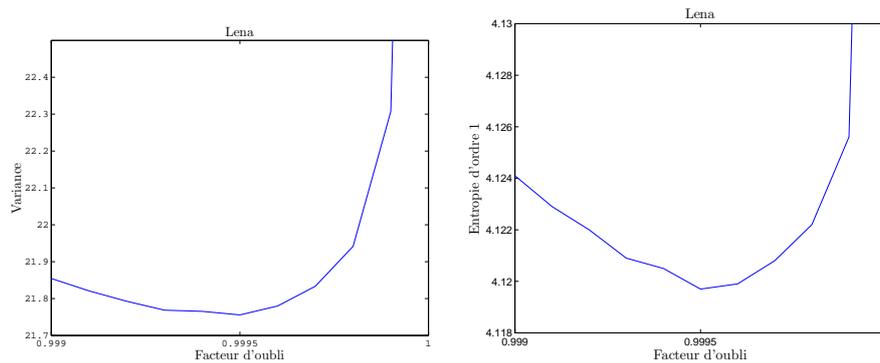


FIG. 4.12 – Entropie d'ordre 1 et de variance en fonction du facteur d'oubli α , pour l'image Lena. La valeur optimale se situe autour de $\alpha = 0,9995$.

4.2.3.5 En-tête du flot de bits

Contrairement à la méthode GAE, dans la méthode LAE, les coefficients des filtres A et B sont recalculés au niveau du décodeur et il n'est donc pas nécessaire de les transmettre, via le flot de bits, du codeur vers le décodeur. Seuls la dimension de l'image, sa valeur moyenne et le nombre de niveaux de décomposition sont écrits dans l'en-tête.

4.2.4 Complexité

Le tableau 4.8 donne les temps de codage et de décodage des différents codeurs pour une image de résolution 512×512 . On peut remarquer que les méthodes adap-

temps	LOCO	CALIC	JASP	S+P	GAE	LAE
codage	0,12	0,21	0,40	0,36	3,19	3,27
décodage	0,11	0,26	0,37	0,35	0,46	3,20

TAB. 4.8 – Temps moyen, exprimé en secondes, de codage et de décodage. Les calculs ont été faits sur un PC PIII 700 MHz, ayant 256 Mb RAM pour des images de dimension $512 \times 512 \times 8$.

tatives ont un temps de codage significativement plus élevé que les autres méthodes et que la méthode GAE a grosso modo le même temps de décodage que les autres codecs.

4.3 Performances en compression sans perte

4.3.1 Avec mesure de débit par entropie d'ordre 1

Nous commençons par analyser les méthodes proposées LAE, GAE et SGAE (variante de la méthode GAE, cf. §4.2.1.4) seules, c'est-à-dire sans appliquer de codeur entropique. Pour cela nous prenons l'entropie du premier ordre moyenne (définie page viii) comme mesure de débit et nous le comparons à celui obtenu avec

d'autres transformations en ondelette d'entiers en entiers [28] : la transformation S [92], la décomposition en ondelettes (2,2), les transformations SPA, SPB et SPC, décrites dans [92] qui correspondent respectivement aux prédicteurs A, B et C de la transformation S+P et les autres transformations apparaissant dans nos tests sont décrites dans [28] et dans l'annexe C.

Nos évaluations portent sur quatre classes d'images : naturelles, médicales (IRM), satellitaires et enfin empreintes digitales (désignées par *Finger*) et textures. Dans les expérimentations, nous décomposons les images de dimension 512×512 ou 768×768 en 5 niveaux et les images de dimension 1024×1024 en 6 niveaux. Les dimensions des images «Moissac» et «Montpellier» ont été changées en les complétant par des zéros. Pour «Moissac» de 802×212 à 832×224 et pour «Montpellier» de 856×1336 à 896×1408 .

L'entropie du premier ordre moyenne est présentée au tableau 4.9. Nous remarquons que

- la transformation S donne généralement l'entropie moyenne la plus élevée (cela est dû au fait qu'elle ne décorrèle que faiblement les signaux) ;
- pour les images relativement lisses (contenant de larges zones de texture) telles que Lena, Peppers, F16, l'entropie moyenne de l'image transformée par la méthode proposée n'est que légèrement inférieure à celle obtenue par la transformation S+P ;
- pour les images contenant plus de contours, comme Goldhill, Barbara, et les images *Finger1-3* (empreintes digitales), les méthodes proposées donnent la plus petite entropie moyenne, l'amélioration étant d'environ 0,15 bpp.

	(1,1)	(2,2)	(4,2)	(2,4)	(2+2,2)	(4,4)	(6,2)	(5,3)	SPA	SPB	SPC	GAE	SGAE	LAE
Lena	4,77	4,35	4,30	4,36	4,33	4,30	4,31	4,46	4,43	4,35	4,39	4,37	4,33	4,33
Goldhill	5,16	4,84	4,83	4,86	4,84	4,83	4,85	4,98	4,97	4,90	4,95	4,82	4,81	4,80
Barbara	5,45	4,99	4,86	4,97	4,87	4,82	4,81	5,08	5,06	4,91	4,83	4,76	4,69	4,68
F16	4,61	4,18	4,14	4,21	4,15	4,14	4,15	4,33	4,32	4,19	4,20	4,20	4,18	4,22
Mandrill	6,36	6,11	6,09	6,12	6,09	6,08	6,10	6,18	6,11	6,11	6,15	6,13	6,07	6,07
Peppers	4,89	4,58	4,58	4,59	4,60	4,58	4,60	4,64	4,62	4,63	4,70	4,64	4,58	4,67
IRM1	4,44	2,31	3,05	2,42	3,10	3,04	3,21	3,66	3,65	3,51	3,39	3,03	3,23	3,10
IRM2	4,58	2,35	3,10	2,46	3,17	3,10	3,28	3,80	3,79	3,62	3,49	2,78	3,00	3,05
IRM3	5,56	5,18	5,11	5,20	5,11	5,11	5,10	5,31	5,30	5,12	5,06	4,91	4,88	5,01
IRM4	4,65	4,54	4,47	4,58	4,46	4,48	4,46	4,61	4,63	4,42	4,36	4,43	4,45	4,38
IRM5	4,43	4,24	4,11	4,29	4,11	4,12	4,09	4,34	4,34	4,15	4,04	4,40	4,35	4,03
IRM6	3,10	1,84	2,36	1,92	2,40	2,36	2,49	2,75	2,76	2,64	2,60	3,47	3,71	2,39
Pentagone	5,44	5,27	5,27	5,28	5,28	5,26	5,28	5,32	5,32	5,53	5,72	5,30	5,27	5,28
Sanfrancisco	5,11	4,88	4,86	4,89	4,87	4,85	4,86	4,95	4,88	4,95	5,35	4,88	4,85	4,85
Okland	4,58	4,38	4,37	4,39	4,38	4,36	4,38	4,47	4,38	4,47	4,84	4,37	4,35	4,38
Toulouse	5,77	5,23	5,15	5,26	5,16	5,15	5,14	5,40	5,23	5,40	5,92	5,05	5,06	5,11
Genes	4,75	4,28	4,25	4,31	4,25	4,25	4,26	4,44	4,28	4,46	5,01	4,21	4,24	4,24
Airplane*	4,59	4,51	4,52	4,51	4,53	4,51	4,54	4,52	4,51	4,66	4,80	4,46	4,48	4,50
Airport*	5,60	5,40	5,38	5,42	5,39	5,38	5,38	5,46	5,40	5,66	5,82	5,22	5,21	5,31
Finger1	5,70	4,85	4,61	4,88	4,63	4,62	4,57	5,00	4,85	4,99	4,53	4,34	4,33	4,40
Finger2	4,93	4,25	4,09	4,27	4,11	4,10	4,07	4,40	4,25	4,40	4,09	3,91	3,89	3,94
Finger3	5,55	4,78	4,56	4,80	4,60	4,60	4,56	4,93	4,78	4,92	4,54	4,37	4,36	4,40

TAB. 4.9 – Entropie moyenne de l'image décomposée avec 5 ou 6 (images ayant une astérisque) niveaux de décomposition. Pour les méthodes GAE et SGAE, les valeurs des paramètres sont $(p, q) = (3, 3)$ et un sous-échantillonnage séparable. Pour la méthode LAE, les valeurs des paramètres sont, $\alpha = 0, 9995$, une extension symétrique, un sous-échantillonnage séparable. Le choix de (r_1, r_2) est indiqué au § 4.2.3.3.

4.3.2 Avec débit réel

Analysons maintenant le système de compression complet en associant aux décompositions GAEa et LAE un codeur arithmétique adaptatif. Le tableau 4.10 compare la taille du flot de bits obtenu à d'autres systèmes de compression sans perte : LOCO-I [4] (standard JPEG-LS), CALIC [2], S+P avec codage arithmétique [1] et JASPER [3] (une implantation du standard JPEG2000-part1). Des valeurs moyennes sont données par classes d'images. La ligne « Moyenne » associée aux images naturelles a été obtenue avec un jeu de 14 images de dimensions 512×512 et 1024×1024 . Celle associée aux images médicales IRM a été obtenue avec un jeu de 16 images, toutes de dimension 512×512 . Pour les images satellitaires, la moyenne a été calculée à partir de 14 images rectangulaires de dimensions variées. Pour les images de texture et d'empreintes digitales, la moyenne a été calculée à partir de 11 images de dimensions 512×512 et 768×768 .

	pas de codage progr.		codage progressif possible				
Image	LOCO	CALIC	JASP	S+P	GAEa	LAE	
Images naturelles	Lena	4,24	4,13	4,32	4,17	4,26	4,21
	Goldhill	4,71	4,65	4,84	4,75	4,78	4,75
	Barbara	4,74	4,51	4,66	4,53	4,65	4,59
	Mandrill	6,04	5,90	6,11	5,93	5,99	5,98
	Peppers	4,49	4,39	4,62	4,54	4,58	4,56
	Airplane	4,61	4,47	4,62	4,50	4,45	4,46
	Airport	5,32	5,22	5,48	5,32	5,27	5,24
	Moyenne	4,82	4,70	4,91	4,78	4,78	4,77
	Images médicales	IRM1	2,27	2,20	2,58	2,41	2,60
IRM2		2,54	2,34	1,69	2,59	2,31	2,13
IRM3		5,27	5,09	5,22	5,04	5,02	5,17
Moyenne		2,81	2,65	2,40	2,86	2,71	2,54
Images satellitaires	Genes	3,81	3,72	4,01	3,89	3,88	3,85
	Mars	4,26	4,03	4,24	3,85	3,76	4,01
	Okland3	4,41	4,28	4,44	4,31	4,26	4,28
	Moyenne	4,70	4,53	4,85	4,69	4,67	4,61
Images de textures	Fing1	4,57	4,44	4,46	4,33	4,25	4,24
	Text1	6,71	6,62	6,79	6,53	6,48	6,46
	Text2	5,97	5,88	6,18	5,95	5,91	5,93
	Moyenne	4,69	4,58	4,75	4,62	4,59	4,57

TAB. 4.10 – Taille du flot de bits (en bpp) pour les codeurs S+P, LOCO-I (JPEG sans perte), CALIC, JASPER (JPEG 2000) et les méthodes proposées avec une adaptation globale (GAEa) ou locale (LAE). Seuls les quatre codeurs GAEa, LAE, JASPER et S+P permettent un codage progressif. La ligne “Moyenne” est calculée pour chaque classe d'images : médicales IRM, naturelles, satellitaires et un lot d'empreintes digitales et de textures.

Pour les images naturelles, CALIC donne la plus petite taille du flot de bits, et

parmi les codeurs qui permettent un codage progressif, la méthode LAE donne le plus petit nombre de bits par pixel, légèrement inférieur à celui obtenu avec S+P ou LOCO-I. Quand à JASPER, il est plus élevé d'environ 0,1 bpp que les autres codeurs.

En revanche pour les images médicales IRM, il est significativement plus performant que tous les autres codeurs. Cela est dû au fait que ces images sont lisses et que la décomposition en ondelettes (5,3) réversible utilisée dans JASPER, ayant plus de moments nuls, est bien adaptée aux images lisses [28]. L'efficacité de JASPER diminue avec des images contenant des contours raides. Pour la famille de textures et d'empreintes digitales, le codeur CALIC et la méthode LAE sont similaires et plus performants que les autres codeurs.

4.3.3 Codage progressif en résolution

Pour évaluer l'efficacité des méthodes GAE et LAE en codage progressif, nous utilisons l'entropie du premier ordre moyenne aux différentes résolutions. Une comparaison des performances est donnée à la figure 4.13. Les méthodes LAE et GAE donnent des taux de compression comparables pour les différentes résolutions. L'entropie du premier ordre moyenne des images aux différentes résolutions est plus petite que celles obtenues par les transformations S+P et 5/3.

4.3.4 Analyse des résultats

En conclusion, sur des images naturelles ou IRM médicales, les performances du codage par prédiction linéaire associé à des décompositions hiérarchiques pyramidales ne dépassent pas celles obtenues avec des décompositions à coefficients fixes. Cela n'est pas en contradiction avec l'étude théorique du chapitre 3. En effet, l'hypothèse de gaussianité qui permet de remplacer le "bon" critère en codage sans perte (par transformée) par celui de minimisation du produit des variances (voir le dernier paragraphe de la section 3.2) n'est pas satisfaite par les images naturelles ou IRM médicales.

Pour les classes d'images satellitaires et de textures, les performances ultimes (dans le sens où toute l'information disponible au décodeur est exploitée) du codage par prédiction linéaire associé à des décompositions hiérarchiques donnent un gain en codage faible (de l'ordre de 0,05 à 0,08 bpp) par rapport aux décompositions à coefficients fixes. Ce gain est largement en dessous de ceux que nous avons obtenus sur les signaux synthétiques à la fin du chapitre précédent.

Pour terminer ce chapitre, nous présentons des travaux réalisés pour adapter les structures en *lifting scheme* généralisé à du codage avec pertes. Notre objectif, au départ, était de réaliser un codage progressif en qualité.

4.4 Peux-on faire du codage progressif en qualité ?

4.4.1 Schéma LAE modifié

Dans la méthode LAE, pendant la reconstruction de l'image, la procédure de mise à jour des coefficients des filtres doit être la réplique exacte de celle utilisée lors de

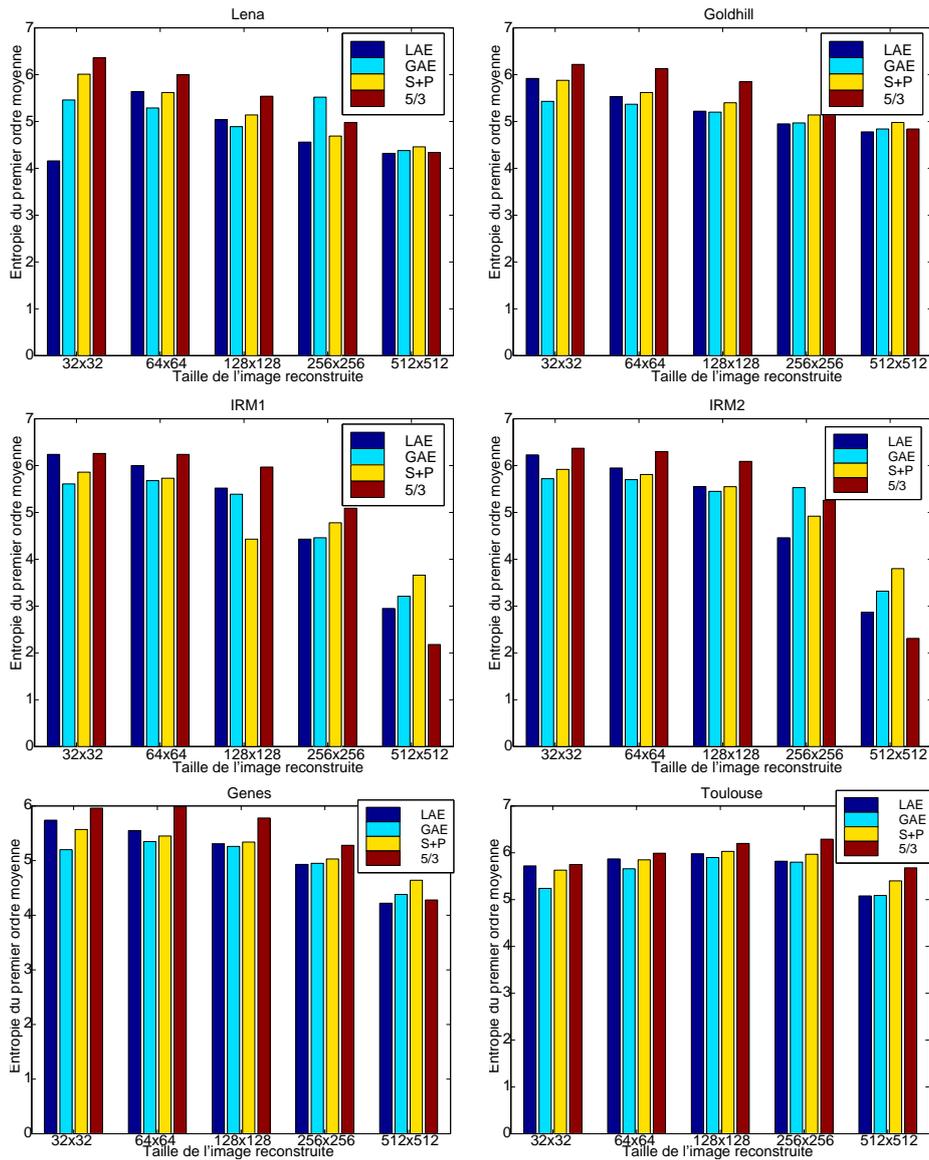


FIG. 4.13 – Entropies moyennes des images d'approximation suivant leur résolution, pour différentes décompositions (S+P, GAE, LAE et ondelette bi-orthogonale 5/3). Cela permet de comparer les performances des décompositions pour du codage progressif en résolution.

la décomposition de l'image. C'est la condition *sine qua non* pour pouvoir reconstruire l'image, même approximativement. Sans quoi une divergence de l'algorithme de reconstruction est observée.

Pour être utilisée en codage avec pertes, la décomposition associée à la méthode LAE doit être suivie d'une étape de quantification. Cette dernière entraîne une distorsion des coefficients transformés, et par conséquent l'impossibilité de la reconstruction d'une image proche de l'image originale, à cause de la propagation de l'erreur de reconstruction qui s'amplifie d'un pixel à l'autre jusqu'à la divergence de l'algorithme de reconstruction. La figure 4.14 illustre cette divergence. Le premier coefficient de l'image décomposée a été modifié en ajoutant 1 à sa valeur réelle. Nous avons représenté la première colonne de l'image originale ainsi que celle de l'image reconstruite

où le phénomène de divergence apparaît clairement.

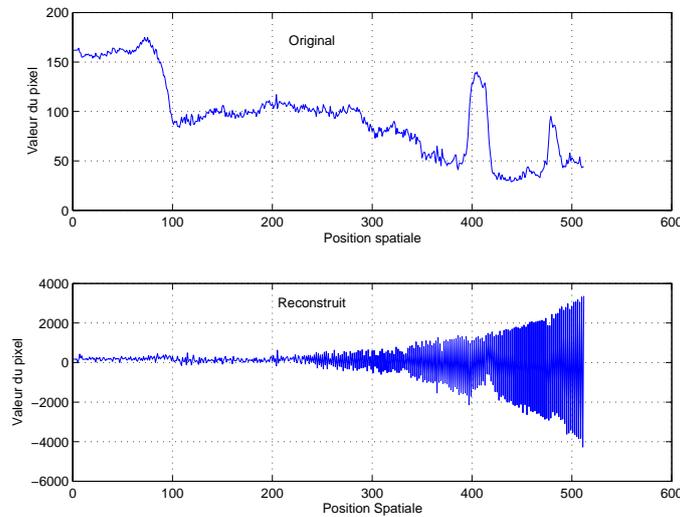


FIG. 4.14 – Illustration du phénomène de divergence de l'algorithme de reconstruction dans le cas où les coefficients transformés sont modifiés.

Deux sources de propagation d'erreur sont en cause. La première est *intra-résolution* : l'erreur de reconstruction des pixels de l'image à une résolution donnée se propage en s'amplifiant jusqu'à la divergence. La seconde est *inter-résolution* puisque l'image reconstruite à la résolution Res donnée est utilisée pour la reconstruction de l'image à la résolution $Res + 1$. Notons que ces erreurs sont dues à l'indisponibilité au niveau du décodeur des informations utilisées pour mettre à jour les coefficients des filtres du codeur. Pour y remédier, le schéma de base de la méthode LAE a été modifié pour que la mise à jour des filtres du codeur utilise uniquement l'information disponible au décodeur.

4.4.1.1 Un niveau de décomposition

Considérons d'abord le cas d'un seul niveau de décomposition. La solution au problème de la propagation *intra-résolution* des erreurs est l'introduction du quantificateur dans la boucle de prédiction, d'une manière similaire à une quantification prédictive en boucle fermée [48, 77], comme le montre la figure 4.15.a. L'opération inverse est obtenue en utilisant la structure de la figure 4.15.b. Remarquons que pour la mise à jour des coefficients des filtres $A(z_1, z_2)$ et $B(z_1, z_2)$, les mêmes quantités sont utilisées à la décomposition aussi bien qu'à la reconstruction, ce qui permet d'éviter la propagation des erreurs intra-résolution pendant la reconstruction.

Il reste à traiter le cas des propagations inter-résolutions où les erreurs de quantification se propagent des résolutions inférieures vers les résolutions supérieures.

4.4.1.2 Plusieurs niveaux de décompositions

Dans le cas où l'image est décomposée sur plusieurs niveaux, la propagation des erreurs de quantification peut être *inter-résolutions*. Le schéma proposé pour

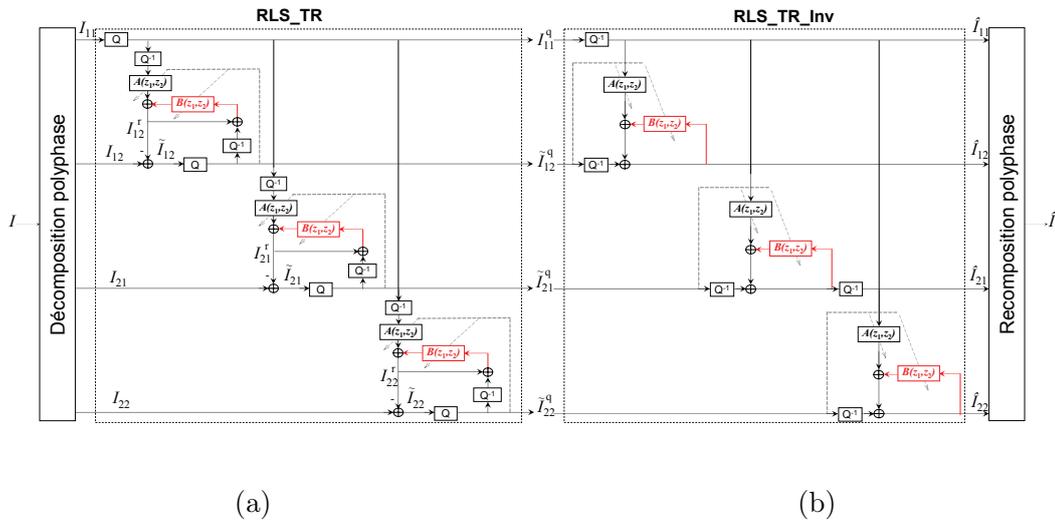


FIG. 4.15 – Un niveau de décomposition et de reconstruction d’une image par la méthode LAE modifiée.

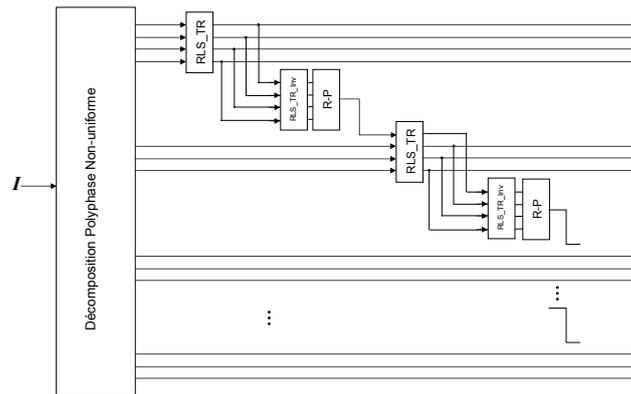


FIG. 4.16 – Schéma de décomposition proposé, pour plusieurs niveaux de décomposition.

et remédier est donné à la figure 4.16. Cette solution consiste à utiliser un sous-échantillonnage non uniforme avec N_{sb} sous-bandes, représenté à la figure 4.17. Les sous-bandes de détails sont par la suite générées en partant de la résolution la plus petite, contrairement aux décompositions multi-résolutions classiques où on procède en partant de la plus grande résolution vers la plus petite. Puisque pendant la reconstruction une approximation \hat{I}_1 de l’image à la résolution courante I_1 est disponible, le schéma de décomposition proposé reproduit cette approximation \hat{I}_1 et fait la mise à jour des filtres à partir de cette dernière (cf. Fig. 4.16).

4.4.2 Description du codec

Dans cette étude nous considérons un quantificateur scalaire uniforme suivi d’un codeur entropique par sous-bande. Cette structure offre de bonnes performances [48] et est simple à mettre en œuvre. Nous nous plaçons dans le cas d’une compression à

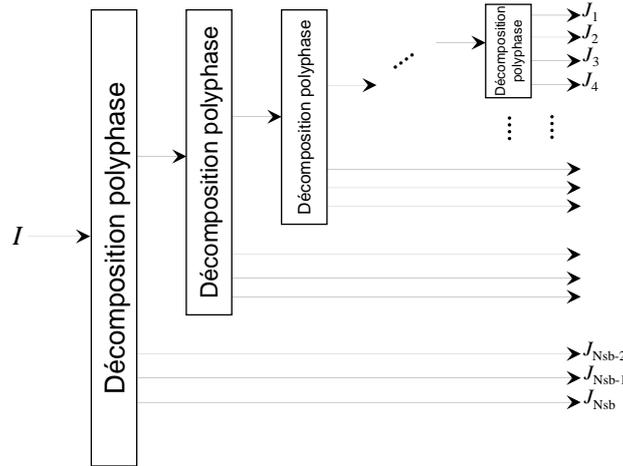


FIG. 4.17 – *Décomposition polyphase avec sous-échantillonnage non uniforme, pour plusieurs niveaux de décomposition.*

hauts débits et faisons l’hypothèse que l’approximation asymptotique d’une quantification à haute résolution est valable dans chaque sous-bande de l’image transformée. C’est-à-dire que les bruits de quantification sont blancs et décorrélés entre eux.

Nous avons vu au paragraphe 2.2.3 page 36 que la distorsion D_{sortie} entre l’image originale et l’image reconstruite se met sous la forme

$$D_{sortie} \approx \sum_{k=1}^{N_{sb}} a_k w_k D_k \quad (4.13)$$

où $D_k = \frac{q^2}{12}$ est la distorsion moyenne du quantificateur de la $k^{\text{ème}}$ sous-bande, a_k la proportion des coefficients transformés appartenant à la $k^{\text{ème}}$ sous-bande et les facteurs de pondération w_k dépendent du banc de filtres de synthèse utilisé pour reconstruire l’image. Le débit moyen vaut

$$R_{sortie} = \sum_{k=1}^{N_{sb}} a_k R_k, \quad (4.14)$$

avec R_k l’entropie d’ordre 1 des coefficients quantifiés de la $k^{\text{ème}}$ sous-bande.

Sous ces conditions, nous avons vu que l’allocation optimale entre quantificateurs est assurée si et seulement si $D_k = D_{cible}/w_k$ pour la $k^{\text{ème}}$ sous-bande. Ce qui donne comme pas de quantification $q_k = \sqrt{12D_{cible}/w_k}$ pour la $k^{\text{ème}}$ sous-bande. Ainsi, un coefficient transformé x_k dans la $k^{\text{ème}}$ sous-bande sera quantifié en

$$Q[x_k] = \left\lfloor \frac{x_k \sqrt{w_k}}{\sqrt{12D_{cible}}} + 0,5 \right\rfloor. \quad (4.15)$$

D’après cette relation, pour une distorsion cible D_{cible} donnée, quantifier les différentes sous-bandes avec les pas de quantification q_k optimaux est équivalent à :

- multiplier chaque sous-bande par la racine carrée du coefficient de pondération w_k correspondant,

- quantifier par la suite toutes les sous-bandes avec un même pas de quantification.

C'est de ce que nous faisons dans nos tests.

Les résultats sont présentés sous la forme de courbes débit/distorsion. La distorsion est mesurée par le PSNR (en décibels) entre l'image originale et l'image reconstruite, et le débit (en bits par pixel) est estimé par la moyenne pondérée des entropies d'ordre 1 des différentes sous-bandes de coefficients.

4.4.3 Estimation des coefficients de pondération

En connaissant les valeurs des coefficients des filtres de la décomposition, les coefficients de pondération peuvent être calculés d'une manière exacte. Dans [123], Woods et Naven proposent une méthode de calcul des valeurs exactes de ces coefficients dans le cas de bancs de filtres de synthèses constitués de filtres RIF. Pour nos transformations, une estimation de leurs valeurs est utilisée, en raison du fait que nos filtres de synthèse sont de type ARMA. Les bruits de quantification dans les sous-bandes sont supposés blancs décorrélés entre eux et de moyenne nulle. Supposons que seule la sous-bande k soit quantifiée, nous avons donc

$$D_\ell = 0, \quad \forall \ell \neq k$$

et

$$D = a_k w_k D_k$$

où D est la puissance du bruit de quantification en sortie et D_k la puissance du bruit de quantification dans la sous-bande k .

Soit b_k le bruit de quantification dans la sous-bande k et I_k^{bruit} le bruit de quantification en sortie (après reconstruction de l'image) nous avons

$$\frac{1}{NM} \sum_{(i,j) \in S} |I_k^{bruit}(i,j)|^2 = w_k a_k \frac{1}{N_k M_k} \sum_{(i,j) \in S_k} |b_k(i,j)|^2 \quad (4.16)$$

où $S = \{(i,j) \in \mathbb{N}^2 : 1 \leq i \leq N \text{ et } 1 \leq j \leq M\}$ et $S_k = \{(i,j) \in \mathbb{N}^2 : 1 \leq i \leq N_k \text{ et } 1 \leq j \leq M_k\}$. Sachant que $a_k = N_k M_k / NM$, on déduit la relation donnant un estimateur du facteur de pondération

$$\hat{w}_k = \frac{\sum_{(i,j) \in S} |I_k^{bruit}(i,j)|^2}{\sum_{(i,j) \in S_k} |b_k(i,j)|^2} \quad (4.17)$$

Maintenant, le facteur de pondération w_k correspondant à la sous-bande k peut être estimé en procédant comme suit :

- Faire coïncider cette dernière avec un bruit blanc de densité de probabilité uniforme et de moyenne nulle en annulant le reste des sous-bandes.
- Reconstruire une image en appliquant la décomposition inverse.
- Appliquer la relation (4.17) pour estimer le facteur de pondération w_k .

Quand la dimension de la sous-bande est petite, plusieurs estimations indépendantes \hat{w}_k peuvent être réalisées et moyennées pour améliorer l'estimation de w_k .

Afin de valider cette méthode d'estimation des coefficients de pondération, nous l'avons appliquée à la décomposition en ondelettes de Daubechies (unidimensionnelle)

et comparé les valeurs estimées avec les valeurs exactes données par B. Usevitch dans [111]. Comme le montre le tableau 4.11 les valeurs estimées sont suffisamment proches des valeurs exactes calculées à partir des coefficients des filtres [123].

Poids	Exacts	Estimés
w_1	0,98295	0,98299
w_2	1,04043	1,04040
w_3	1,03060	1,03066
w_4	0,96721	0,96736
w_5	1,05209	1,05214
w_6	1,03963	1,03973
w_7	1,05848	1,05839
w_8	1,07512	1,07488

TAB. 4.11 – Tous les coefficients de pondération possibles pour 1, 2, 3 et 4 niveaux de décomposition de la décomposition (9,7) de Daubechies unidimensionnelle. Les valeurs exactes sont données par B. Usevitch [111] et celles estimées par la méthode décrite plus haut. Pour un niveau de décomposition les coefficients sont (w_1, w_2) , (w_1, w_3, w_4) pour deux niveaux...

Le tableau 4.12 donne les coefficients de pondération estimés pour quelques transformations ainsi que la transformation GAE. Pour cette dernière, les facteurs de pondération dépendent de l'image (Lena pour les coefficients donnés dans le tableau 4.12). Pour une image donnée, les coefficients des filtres A et B sont calculés (avec la méthode GAE) pour chaque niveau de décomposition, les facteurs de pondération sont ensuite estimés en utilisant ces coefficients.

Poids	Haar	9/7	5/3	GAE
\hat{w}_1	1,0000	1,1135	30,0179	24,1474
\hat{w}_2	1,0000	1,0961	8,6025	11,2719
\hat{w}_3	1,0000	1,0961	2,5536	11,2401
\hat{w}_4	1,0000	1,1098	8,7347	14,8845
\hat{w}_5	1,0000	0,9990	2,5695	3,5901
\hat{w}_6	1,0000	0,9406	0,8649	3,5391
\hat{w}_7	1,0000	1,0051	2,5739	4,5139
\hat{w}_8	1,0000	1,0238	1,0823	1,8372
\hat{w}_9	1,0000	1,0843	0,5151	1,8336
\hat{w}_{10}	1,0000	1,0242	1,0847	1,7375

TAB. 4.12 – Estimation des poids des distorsions correspondants aux différentes sous-bandes pour quelques décompositions avec trois niveaux de décomposition.

Dans le cas de la décomposition par la méthode LAE modifiée. Les facteurs de pondération w_k sont égaux à 1. Nous allons le vérifier dans le cas d'un niveau de décomposition, le cas général s'en déduisant.

En appelant $Q[.]$ l'opérateur de quantification, $Q^{-1}[.]$ son opérateur inverse et

$A[\cdot]$, $B[\cdot]$ les opérateurs de filtrage avec les filtres $A(z_1, z_2)$ et $B(z_1, z_2)$. Au niveau du banc de filtres d'analyse (voir figure 4.15.a) nous pouvons écrire

$$\tilde{I}_{12} = I_{12} - I_{12}^r \quad (4.18)$$

et au niveau du banc de filtres de synthèse

$$\hat{I}_{12} = Q^{-1}[Q[\tilde{I}_{12}]] + I_{12}^r. \quad (4.19)$$

En combinant ces deux équations nous trouvons

$$I_{12} - \hat{I}_{12} = \tilde{I}_{12} - Q^{-1}[Q[\tilde{I}_{12}]]. \quad (4.20)$$

À partir de cette relation, nous remarquons que la distorsion entre l'image originale et celle reconstruite due à la quantification est égale à celle des images de sous-bandes quantifiées : la formule (4.13) est valable pour des coefficients w_k égaux à 1.

4.4.4 Évaluation des performances à hauts débits

4.4.4.1 La méthode GAEEa

Nous avons comparé la décomposition de la méthode GAEEa aux décompositions en ondelettes (5,3) et (9,7) de Daubechies et observé les écarts de PSNR à hauts débits (voir le tableau 4.13). Pour toutes les images, testées sauf les textures, la méthode GAEEa présente des performances inférieures à celles des ondelettes de Daubechies. Remarquons que la méthode GAEEa pourrait permettre un codage progressif en qualité si ses performances en codage avec pertes étaient meilleures.

4.4.4.2 La méthode LAE modifiée

De même que pour la méthode GAEEa, nous avons comparé la décomposition de la méthode LAE modifiée aux décompositions en ondelettes de Daubechies et observé des écarts de PSNR à hauts débits comme le montre le tableau 4.13). Pour toutes les familles d'images testées, LAE modifiée présente des performances bien supérieures à celles de la (9,7) et idem par rapport à la (5,3) (sauf pour les images IRM). Les courbes expérimentales PSNR versus entropie moyenne sont présentées à l'annexe A. Remarquons toutefois que la méthode LAE modifiée ne permet pas un codage progressif en qualité car les filtres dépendent de la qualité visée.

4.5 Conclusion

Dans ce chapitre nous avons évalué les performances en codage d'images de la structure en *lifting scheme* généralisé, que nous avons introduite au chapitre 3 pour exploiter toute l'information disponible au décodeur en codage par prédiction linéaire associé à des décompositions multi-résolutions.

Nous avons constaté que pour les images naturelles ou IRM médicales, les décompositions à coefficients fixes ont des performances très proches que celles que nous avons introduites, parfois même meilleures. Cela montre que le critère que nous avons retenu (produit des variances des sous-bandes) pour adapter les filtres prédictifs

Δ PSNR(dB)	(GAEa,9/7)	(GAEa,5/3)	(LAE _{modifiée} ,9/7)	(LAE _{modifiée} ,5/3)
Lena	-3,03	-2,55	0,33	0,81
Goldhill	-2,31	-2,24	0,96	1,04
Barbara	-2,99	-1,89	1,42	2,52
F16	-2,77	-2,62	0,53	0,68
Mandrill	-1,99	-1,67	0,60	0,92
Peppers	-2,90	-2,54	0,21	0,57
Irm1	-2,00	-6,17	1,79	-2,38
Irm2	-2,30	-6,93	2,83	-1,81
Irm3	-1,78	-1,41	1,93	2,30
Irm4	-2,00	-6,17	1,79	-2,38
Texture1	0,29	0,97	0,13	0,81
Texture2	0,28	0,82	0,12	0,66
Texture3	0,59	1,22	0,15	0,79
Texture4	3,88	4,44	0,12	0,69
Texture5	1,12	1,81	0,13	0,82
Texture6	0,18	0,85	0,10	0,77
Finger1	-1,92	-1,02	2,74	3,64
Finger2	-2,29	-1,76	2,22	2,75
Finger3	-2,00	-1,36	2,45	3,08

TAB. 4.13 – Comparaison des méthodes à hauts débits.

ne convient pas pour ces classes d'images. En effet, il est bien connu que l'hypothèse de gaussianité (nécessaire pour justifier le critère que nous avons utilisé) n'est pas satisfaite pour ces images.

Toutefois, parmi les codeurs permettant un codage progressif en résolution, ceux introduits dans cette thèse ont les meilleures performances (Comparées à JPEG 2000 et S+P) pour les images satellitaires et de texture.

Enfin, nous avons modifié la méthode LAE pour l'appliquer en compression avec pertes et constaté qu'elle offrait des meilleures performances en codage à hauts débits que la décomposition en ondelettes (9,7) de Daubechies.

Une suite possible de cette étude consiste en la modification du critère d'adaptation des filtres : choisir le bon critère ; $I(Y_1, \dots, Y_N)$ ou de façon équivalente $\sum_{i=1}^N h(Y_i)$ (avec les notations du paragraphe 3.2).

Conclusion générale

Dans ce rapport, nous avons vu, à l'issue de l'état de l'art sur les structures en *liftings schemes* adaptatifs utilisées en compression sans perte des images, que toute l'information disponible au décodeur pour estimer la valeur du pixel courant n'était pas exploitée jusqu'à ce jour, et que par conséquent les performances limites atteignables en codage par prédiction linéaire associé à des décompositions multi-résolutions n'étaient pas connues.

Après avoir justifié le critère de minimisation de la variance pour coder sans perte des signaux gaussiens, nous avons proposé une structure en *lifting scheme* généralisé qui permet d'exploiter toute l'information disponible au décodeur pour estimer la valeur du pixel courant. Les filtres prédicteurs apparaissant dans cette structure sont adaptés suivant deux méthodes : les moindres carrés (l'image à coder est alors supposée globalement stationnaire) et les moindres carrés adaptatifs (seule la stationnarité locale est supposée ici). Nous avons appelé respectivement GAE et LAE les décompositions et les méthodes associées. Nous avons comparé les performances en compression sans perte de cette nouvelle structure à celle de Gerek et Çetin et aux décompositions en ondelettes (5,3) et (9,7) de Daubechies, utilisées dans le standard JPEG 2000, et montré sur des signaux de textures synthétiques gaussiens que le gain moyen (sur l'entropie d'ordre 1 des coefficients transformés) de la structure en *lifting scheme* généralisé est de 0,5 bpp par rapport à la meilleure des ondelettes (pour ces textures, c'est la (9,7) réversible) et de 0,4 bpp par rapport à la structure adaptative de Gerek et Çetin.

Nous avons ensuite réglé, pour quatre familles d'images réelles (naturelles, satellitaires, IRM médicales et enfin textures et empreintes), les paramètres de la structure en *lifting scheme* généralisé pour optimiser les performances en codage sans perte, avec la possibilité d'un codage progressif en résolution. En comparant les performances obtenues avec les codecs les plus usités aujourd'hui (et les plus performants), nous avons constaté que pour les images naturelles ou IRM médicales, les décompositions à coefficients fixes ont des performances très proches de celles que nous avons introduites, parfois même meilleures. Cela montre que le critère que nous avons retenu (produit des variances des sous-bandes) pour adapter les filtres prédicteurs ne convient pas pour ces classes d'images. Et effectivement, il est bien connu que l'hypothèse de gaussianité (nécessaire pour justifier le critère que nous avons utilisé) n'est pas satisfaite pour ces images. Toutefois, parmi les codeurs permettant un codage progressif en résolution, ceux introduits dans cette thèse ont les meilleures performances (en comparaison à JPEG 2000 et S+P) pour les images satellitaires et de texture. Nous avons également observé que, pour toutes les classes d'images étudiées, l'ajout du deuxième filtre prédicteur dans le *lifting scheme* améliore significativement

les performances en compression sans perte, autrement dit utiliser toute l'information disponible au décodeur pour prédire la valeur du pixel courant améliore le codage.

Enfin, nous avons évalué les performances en codage avec pertes de la méthode GAE, c'est la seule qui permet un codage progressif en qualité parmi les deux que nous avons introduites. Pour cela, nous avons estimé les facteurs de pondération qui doivent s'appliquer aux différentes sous-bandes pour assurer une allocation optimale entre quantificateurs scalaires uniformes, et nous avons estimé le débit par l'entropie d'ordre 1. Les performances obtenues sont significativement en-dessous de l'état de l'art pour les images naturelles, IRM médicales et les empreintes digitales, seules les textures sont mieux codées à haute résolution avec la méthode GAE. Ces faibles performances sont dues, d'une part au fait que le critère retenu pour ajuster la structure n'est justifié qu'en compression sans perte, en codage avec pertes il faudrait tenir compte de la distorsion. D'autre part, les filtres sont adaptés sans aucune contrainte sur la dynamique des coefficients obtenus après filtrage, autrement dit le gain du filtre à la fréquence de Nyquist n'est pas limité à 1, (voir les travaux de Gouze et *al.* sur ce sujet). Nous avons également modifié la méthode LAE pour qu'elle puisse être utilisée en compression avec pertes. Le codeur obtenu ne permet pas un codage progressif en qualité, mais pour de hauts débits la méthode LAE modifiée a un gain de codage toujours positif et parfois significatif par rapport à la décomposition en ondelettes (9,7), ceci pour les quatre familles d'images testées.

En conclusion, nous avons montré que sur les images naturelles, IRM médicales ou satellitaires, le codage par prédiction linéaire associé à des décompositions multi-résolutions atteint des performances en compression sans perte qui ne sont pas significativement meilleures que les décompositions à coefficients fixes. Une suite logique à cette étude serait de remplacer le critère d'adaptation de la structure en *lifting scheme* généralisé par la minimisation de l'information mutuelle entre x_ℓ et x_h . Il faudrait aussi tenir compte de la compression avec (de faibles) pertes et modifier le critère en conséquence.

BIBLIOGRAPHIE

- [1] <http://www.cipr.rpi.edu/research/spiht/>
- [2] <http://www.csd.uwo.ca/faculty/wu/>
- [3] <http://www.ece.ubc.ca/mdadams/jasper/>
- [4] <http://www.hpl.hp.com/loco/software.htm>
- [5] Aas K. C. et Mullis T., “Minimum mean-squared error transform coding and subband coding”, *IEEE Trans. on Info. Theory*, 42 :1179–1192, juil. 1996.
- [6] Aase K. C. et Ramstad T., “On the optimality of nonuniform filter banks in subband coders”, *IEEE Trans. on Image Proc.*, 44 :1595–1591, déc. 1995.
- [7] Adams M. D., Reversible Integer-to-Integer Wavelet Transforms for Image Coding, Thèse de doctorat, The University Of British Columbia, septembre 2002.
- [8] Adams M. et Kossentini F. *A flexible computation engine for subband transforms*, ISO/IEC JTC 1/SC 29/WG1, juil. 1998.
- [9] Adams M. et Kossentini F., “Reversible integer-to-integer wavelet transforms for image compression : performance evaluation and analysis”, *IEEE Trans. on Image Proc.*, 9(6) :1010–1024, juin 2000.
- [10] Akaike B., “A new look at the statistical model identification”, *IEEE Trans. on Automatic Control*, 19 :716–723, 1974.
- [11] Akaike B., “Modeling by shortest data description”, *Automatica*, 14 :465–471, 1978.
- [12] Akansu A. N. et Liu Y., “On signal decomposition techniques”, *Optical Engineer*, 30 :912–920, juil. 1991.
- [13] Aksasse B. et Radouane L., “Two-Dimensional Autoregressive (2D-AR) Model Order Estimation”, *IEEE Trans. on Signal Proc.*, 47(7) :2072–2077, juil. 1999.
- [14] Antonini M., Barlaud M., Mathieu P. et Daubechies I., “Image coding using wavelet transform”, *IEEE Trans. on Image Proc.*, 1(2) :205–219, avril 1992.
- [15] Barret M., *Traitement statistique du signal*, photocopié de Supélec, 2005.
- [16] Barret M. et Bekkouche H., “Adapted nonlinear multiresolution decomposition with applications in progressive lossless image coding”, *Proc. of Int. Symposium on Image and Signal Processing and Analysis*, pp. 609–612, Pula, Croatia, juin 2001.
- [17] Barret M. et Narozny M., “Application of ICA to lossless image coding”, *Proc. of Int. Symposium on Independent Component Analysis and Blind Source Separation*, pp. 855–859, Nara, Japon, avril 2003.

- [18] Battu D., *Initiation aux télécoms, technologies et applications*, Dunod, 2002.
- [19] H. Bekkouche et M. Barret, “Décomposition multi-résolution adaptative. Application à la compression sans perte des images”, *Actes de la Réunion des Théoriciens des Circuits de Langue Française*, Paris, 2001.
- [20] Bekkouche H. et Barret M., “Adaptive multiresolution decomposition : applications to lossless image compression”, *Proc. of Int. Conf. on Acoust., Speech and Signal Processing*, pp. 3533–3536, Orlando, Floride, mai 2002.
- [21] Bekkouche H. et Barret M., “Comparison of lossless codecs for satellite and MRI images”, *Proc. of EUSIPCO-02*, pp. 475–478, Toulouse (France), sept. 2002.
- [22] M. Barret, H. Bekkouche, J.-L. Collette et J. Oksman, “Adapted lifting schemes for lossless image coding”, *Workshop Transmitting, processing and watermarking multimedia contents*, Bordeaux, pp. 49–54, avril 2003.
- [23] Bell T. C., Cleary J. C. et Witten I. H. “Text Compression”, Advance reference series. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1990.
- [24] Bellanger M., Bonnerot G. et Coudreuse M., “Digital filtering by polyphase network : application to sample rate alteration and filter banks”, *IEEE Trans. on Acoust. Speech and Signal Proc.*, 24(2) :109–114, avril 1976.
- [25] Benidir M. et Barret M., *Stabilité des filtres et des systèmes linéaires*, Dunod, septembre 1999.
- [26] Blume H. et Fand A., “Reversible and irreversible image data compression using S-transform and Lempel-Ziv coding”, *Proc. SPIE*, 1091 :2–18, 1989.
- [27] Boulgouris N. V., Tzovaras D. et Strintzis M. G., “Lossless image compression based on optimal prediction, adaptive lifting, and conditional arithmetic coding”, *IEEE Trans. on Image Proc.*, 10(1) :1–14, jan. 2001.
- [28] Calderbank R. C., Daubechies I., Sweldens, W. et Yeo B. L., “Wavelet Transforms that Map Integers to Integers”, *Applied and Computational Harmonic Analysis*, 5(3) :332–369, 1998.
- [29] Cardoso J-F., “Infomax and maximum likelihood for blind source separation”, *IEEE Letters on Signal Processing*, Vol. 4, No. 4, pp. 112-114, April 1997.
- [30] Chi C-Y. and Chen W-T., “Linear Prediction Based on Higher Order Statistics”, *Proceedings of the International Symposium on Signal Processing and its Applications*, pp. 230-233, August 1992.
- [31] Chi C-Y., “Linear Prediction, Maximum Flatness, Maximum Entropy, and AR Polyspectral Estimation”, *IEEE Transactions on Signal Processing*, Vol. 41, No. 6, pp. 2155-2164, June 1993.
- [32] Christopoulos C., Skodras A., Ebrahimi T., “The JPEG2000 still image coding system : An Overview”, *IEEE Trans. on Consumer Electronics*, 46(4) :1103–1127, Nov. 2000.
- [33] Chung Y. S. et Kanefsky M., “On 2D recursive LMS algorithms using ARMA prediction for ADPCM encoding of images”, *IEEE Trans. on Image Proc.*, 1(3) :416–422, juil. 1992.

- [34] Claypoole Jr. R. L., Davis G. M., Sweldens W. et Baraniuk R., “Nonlinear wavelet transforms for image coding via lifting”, *IEEE Trans. on Image Proc.*, 12(12) :1449–1450, dec. 2003.
- [35] Cohen A., *Ondelettes et traitement numérique du signal*, Masson, Paris, 1992.
- [36] Comon P., “Independent Component Analysis—a new concept,” *Signal Processing*, Vol. 36, pp. 287–314, 1994.
- [37] Cover T. M. et Thomas J. A., *Elements of Information Theory*, Wiley Series in Communications, 1991.
- [38] Daubechies I., “Orthonormal Bases of Compctly Supported Wavelets”, *Communications on Pure and Applied Mathematics*, vol. XLI, pp. 909–996, 1988.
- [39] Daubechies I. et Sweldens W., “Factoring Wavelet Transforms into Lifting Steps”, *J. Fourier Analysis and Applications*, 4(3) :245–267, 1998.
- [40] Deng G., “A new interpolative subband coding algorithm for lossless image compression”, *Proc. Int. Conf. on Image Processing*, 1 :93–96, Lausanne, sept. 1996.
- [41] Desarte P., Macq B. et Slock D. T. M., “Signal-adapted multiresolution transform for image coding”, *IEEE Trans. on Info. Theory*, 38(2) :897–904, mars 1992.
- [42] Déforges O. et Ronsin J., “Locally Adaptive Resolution method for progressive still image coding”, *Proc. Int. Symp. Signal Processing and its Applications*, Brisbane, Australia, août 1999.
- [43] Djokovic I. et Vaidyanathan P. P., “On optimal analysis/synthesis filters for coding gain maximization”, *IEEE Trans. on Signal Proc.*, 44 :1276–1279, mai 1996.
- [44] O. Egger, W. Li and M. Kunt, “High compression image coding using an adaptive morphological subband decomposition”, *Proceedings of the IEEE*, vol. 83, no. 2, Feb. 1995.
- [45] Friedlander B., Morf M., Kailath T. et Ljung L., “New inversion formulas for matrices classified in terms of their distance from Toeplitz matrices”, *Linear Algebra and its Applications*, 27 :31–60, 1979.
- [46] Gerek Ö, N. et Çetin A. E., “Polyphase adaptive filter banks for fingerprint image compression”, *Proc. 9^{ème} European Signal Processing Conference*, pp. 45–48, Rhodes, Grèce, sept. 1998.
- [47] Gerek Ö, N. et Çetin A. E., “Adaptive polyphase subband decomposition structures for image compression”, *IEEE Trans. on Image Proc.*, 9(10) :1649–1660, oct. 2000.
- [48] Gercho A. et Gray Robert M., *Vector quantization and signal compression*, Kluwers Academic Publishers, Boston, 1993.
- [49] Gosse K. et Duhamel P., “Perfect reconstruction versus MMSE filter banks in source coding”, *IEEE Trans. on Signal Proc.*, 45(9) :2188–2202, sept. 1997.
- [50] Gouze A., Schéma lifting quinconce pour la compression d’images, Thèse de doctorat, Université de Nice-Sophia Antipolis, décembre 2002

- [51] Gouze A., Antonini A., Barlaud M. et Macq B., “Optimized lifting scheme for two-dimensional quincunx sampling”, *Proc. Int. Conf. on Image Processing*, 2 :253–256, Thessaloniki, Grèce, oct. 2001.
- [52] Hall T. E. and Wilson S. G., “Stochastic Image Modelling using Cumulants with Application to Predictive Image Coding”, *Workshop on Higher Order Spectral Analysis*, June 28-30, Vail, Colorado, pp. 239-244, 1989.
- [53] Haykin S., *Adaptive filter theory*, Engelwood Cliffs, NJ : Prentice-Hall, 1991.
- [54] Hong G., Hall G. and Terrell T., “Prediction by Back-propagation Neural Network for Lossless Image Compression”, *Proceedings of ICSP’96 International Conference on Signal Processing*, Vol. 2, pp. 1026-1030, 1996.
- [55] Howard P. G. et Vitter J. S., “Analysis of Arithmetic Coding for Data Compression”, *Information Processing and Management*, 28(7) :pp. 749–763, 1992.
- [56] Herley C. et Vetterli M., “Orthogonal time-varying filter banks and wavelet packets”, *IEEE Trans. on Signal Proc.*, 42(10) :2650–2662, oct. 1994.
- [57] Howard P. G. et Vitter J. S., “Fast and efficient lossless image compression”, *in Proc. of Data Compression Conference*, pp. 351–360, Snowbird, Utah, mars 1993.
- [58] Huang J. et Shultheiss P., “Bloc Quantization of Correlated Gaussian Random Variables”, *IEEE Trans. on Communications*, 11 :289–296, 1963.
- [59] Jain A. K., *Fundamentals of Digital Image Processing*, Englewood Cliffs, New Jersey, Prentice-Hall, 1989.
- [60] Jung H., Choi T. et Prost R., “Rounding transform and its application for pyramid structured coding”, *IEEE Trans. on Image Proc.*, 7(2) :234–237, 1998.
- [61] Kailath T., Kung S. Y. et Morf M., “Displacement ranks of a matrix”, *Bull. Amer. Math. Soc.*, 1(5) :769–773, sept. 1979.
- [62] Kiat Y. C., Yann S. I. and Tong L. C., “Lossless Image Compression Using Neural Networks and Arithmetic Coding”, *Proceedings Vision Interface ’98*, Vancouver, British Columbia, pp. 115-119, June 1998.
- [63] Kirac A. et Vaidyanathan P. P., “On the existence of FIR principal component filter banks”, *IEEE Int. Conf. Acoust. Speech and Signal Processing*, Seattle, avril 1998.
- [64] G. Langdon, A. Gulati et E. Seiler. “On the JPEG model for lossless image compression”. *In Proc. 1992 Data compression conference*, 1992, pp. 172-180.
- [65] Liang G., Wilkes D. M. et Cadzow J. A., “ARMA model order estimation based on eigenvalues of the covariance matrix”, *IEEE Trans. on Signal Proc.*, 41 :3003–3009, oct. 1993.
- [66] Liu X. et Najim M., “A two-dimensional fast lattice recursive least squares algorithm”, *IEEE Trans. on Signal Proc.*, 44(10), oct. 1996.
- [67] Lloyd S. P., “Least Squares Quantization in PCM”, *IEEE Trans. on Info. Theory*, 28(2) :129–137, mars 1982.
- [68] Mallat S., “A theory for multiresolution signal decomposition : the wavelet representation”, *IEEE Trans. on Pattern Anal. and Machine Intelligence*, 11 :674–693, juil 1989.

- [69] Mallat S., “Multifrequency channel decompositions of images and wavelet models”, *IEEE Trans. on Acoust. Speech and Signal Proc.*, 37 :2091–2110, 1989.
- [70] Mallat S., *Une exploration des signaux en ondelettes*, Les Éditions de l’École Polytechnique, 2000.
- [71] Malvar H. S., *Signal processing with lapped transforms*, Artech House, 1992.
- [72] Manikopoulos C. N., “Neural Network Approach to DPCM System Design for Image Coding”, *IEE Proceedings-I*, Vol. 139, No. 5, pp. 501-507, October 1992.
- [73] Memon N. D. and Sayood K., “Lossless image compression : A comparative study”, *Proceedings SPIE*, Vol. 2418, pp. 8-20, 1995.
- [74] Meyer Y., *Ondelettes et Opérateurs I : Ondelettes*, Ed. Hermann, 1990.
- [75] Meyer Y. et Wickerhauser M. V. Coifman R. R. *Wavelet analysis and signal processing, Wavelets and their applications*. Jones and Barlett B. Ruskai et al., pp. 135–178, 1992.
- [76] Moreau de Saint Martin F., *Bancs de filtres et ondelettes : synthèse, adaptation et application*, Thèse de doctorat, Université Paris IX, 1997.
- [77] Moreau N., *Techniques de compression des signaux*, Masson, 1995.
- [78] Moulin P., “A new look at signal-adapted QMF bank design”, in *Proc. of the 20th IEEE Int. Conf. Acoust. Speech Signal Process. ICASSP-95.*, Detroit, Michigan, 8-12 May 1995, pp. 1312–1315.
- [79] M. Narozny, “Analyse en composantes indépendantes et compression de données”, Thèse de doctorat de l’Université Paris XI, Orsay, 2005.
- [80] Narozny M., Michel Barret M., Pham D. T. et Akam Bitu I. P., “Modified ICA algorithms for finding optimal transforms in transform coding”, *Proc. of Int. Symposium on Image and Signal Processing and Analysis*, pp. 15–17, Zagreb, Croatia, Septembre 2005.
- [81] Nijim Y. W., Stearns S. D. and Mikhael W. B., “Lossless Compression of Images Employing a Linear IIR Model”, *IEEE International Symposium on Circuits and Systems*, Vol. 2, pp. 305-308, 1996.
- [82] Oktem R., Gerek Ö. N. et Çetin A. E., “adaptive filter Banks for lossless image compression”, *Proc. of Int. Conf. on Acoust., Speech and Signal Processing*, pp. 1809–1812, Salt Lake City, Utah, mai 2001.
- [83] Parisot C., *Allocations basées modèles et transformée en ondelettes au fil de l’eau pour le codage d’images et de vidéos*, Thèse de doctorat, Université de Nice-Sophia Antipolis, juin 2003.
- [84] Qiu G., “A Progressively Predictive Image Pyramid for Efficient Lossless Coding”, *IEEE Transactions on Image Processing*, Vol. 8, No. 1, pp. 109-115, January 1999
- [85] Ramchandran K. et Vetterli M., “Best wavelet packet in a rate distortion sense”, *IEEE Trans. on Image Proc.*, 2(2) :160–175, avril 1993.
- [86] Ramchandra K., Xiong Z., Asai K. et Vetterli M., “Adaptive transforms for image coding using spatially varying wavelet packets”, *IEEE Trans. on Image Proc.*, 5(7) :1197–1204, juillet 1996.

- [87] J. Rissanen, "A Universal Data Compression System", *IEEE Transactions on Information Theory*, 30(4) :656-664, 1983.
- [88] J. Rissanen. "Universal Coding, Information, Prediction and Estimation", *IEEE Transactions on Information Theory*, 29(5) :629-636, Juillet 1984.
- [89] Roos P., Viergever M. A., van Dijke M. C. A. et Peters J. H., "Reversible Intraframe Compression of Medical Images", *IEEE Trans. on Medical Image*, 7(4) :328-336, 1988.
- [90] Saha S. et Vemuri R., "Multimedia Image Coding using Adaptive Integer-to-Integer Wavelet Transforms", *Proc. of the IEEE International Conference on Multimedia and Expo*, pp. 99-102, New-York, juil. 2000.
- [91] Said A. et Pearlman W. A., "A new, fast and efficient image codec based on Set Partitioning In Hierarchical Trees", *IEEE Trans. on Circuits and Systems for Video Technology*, 6(3) :243-250, juin 1996.
- [92] Said A. et Pearlman W. A., "An image multiresolution representation for lossless and lossy compression", *IEEE Trans. on Image Proc.*, 5 :1303-1310, 1996.
- [93] Santa Cruz D. et Ebrahimi T., "An analytical study of JPEG 2000 functionalities", *rapport ISO/IEC JTC1/SC29/WG1 N1815*, pages 1010-1024, juil. 2000.
- [94] Santa-Cruz D., Ebrahimi T., "A study of JPEG2000 still image coding versus other standards", *Proc. EUSIPCO*, 2 :673-676, Tampere, Finland, Sept. 2000.
- [95] Sayood K. and Anderson K., "A differential lossless image compression algorithm", *IEEE Transactions on Signal Processing*, Vol. 40, No. 1, pp. 236-241, January 1992.
- [96] Segall, A., "Bit allocation and encoding for vector sources", *IEEE Trans. IT.*, 162-169, Mars 1971.
- [97] Shannon C. E., "A Mathematical Theory of Communication", *Bell Sys. Tech. Journ.*, Vol. 27, pp. 379-423 and 623-656, 1948.
- [98] Shapiro J. M., "Embedded image coding using zerotrees of wavelet coefficients", *IEEE Trans. on Signal Proc.*, 41(12) :3445-3462, déc 1993.
- [99] Shoham Y. et Gersho A., "Efficient bit allocation for an arbitrary set of quantizer", *IEEE Trans. on Acoust. Speech and Signal Proc.*, 36(9) :1445-1453, septembre 1988.
- [100] Shynk J. J., "Adaptive IIR Filtering", *IEEE Acoust. Speech and Signal Proc. Magazine*, pp. 4-21, April 1989.
- [101] Stearns S. D., "Error Surfaces of Recursive Adaptive Filters", *IEEE Transactions on Circuits and Systems*, Vol. CAS-28, No. 6, pp. 603-606, June 1981.
- [102] Swami A. and Mendel J. M., "Adaptive System Identification Using Cumulants", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, pp. 2248-2251, 1988.
- [103] Sweldens W., "The Lifting Scheme : A New Philosophy in Biorthogonal Wavelet Constructions", dans *Wavelet Applications in Signal and Image Processing III*, éditeurs A. F. Laine et M. Unser, pp. 68-79. Proc. SPIE 2569, 1995.

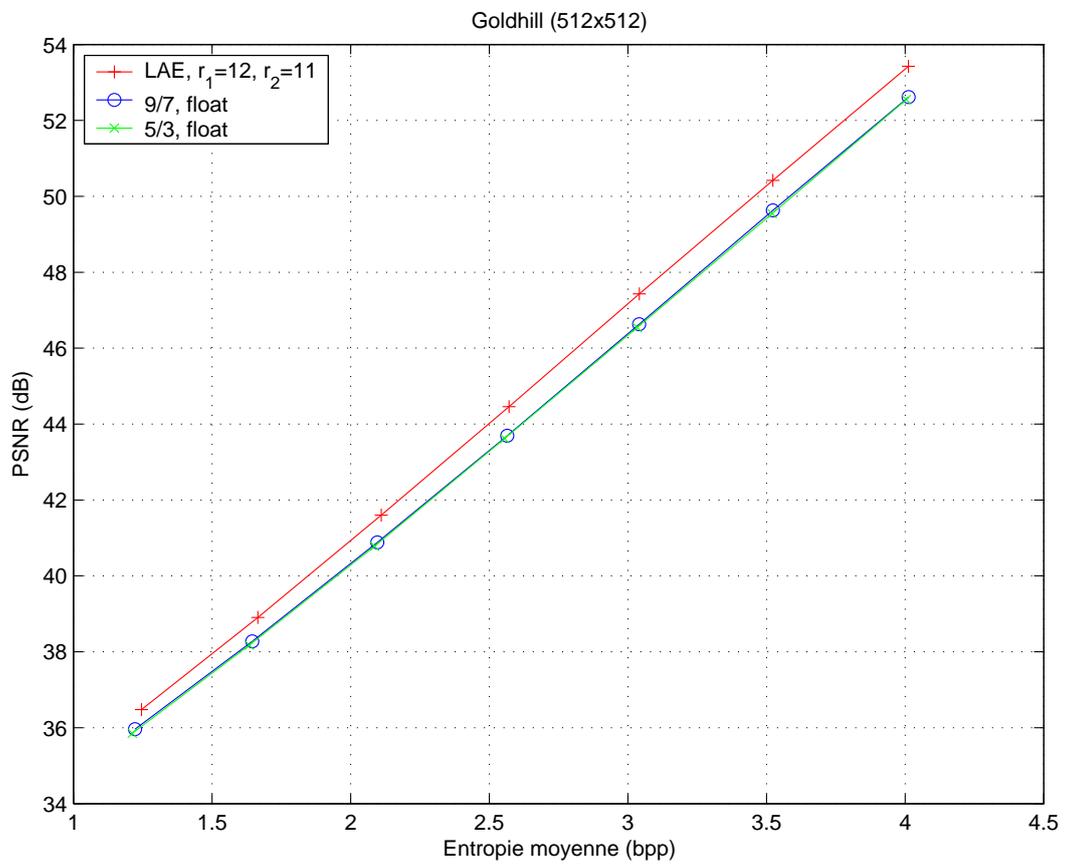
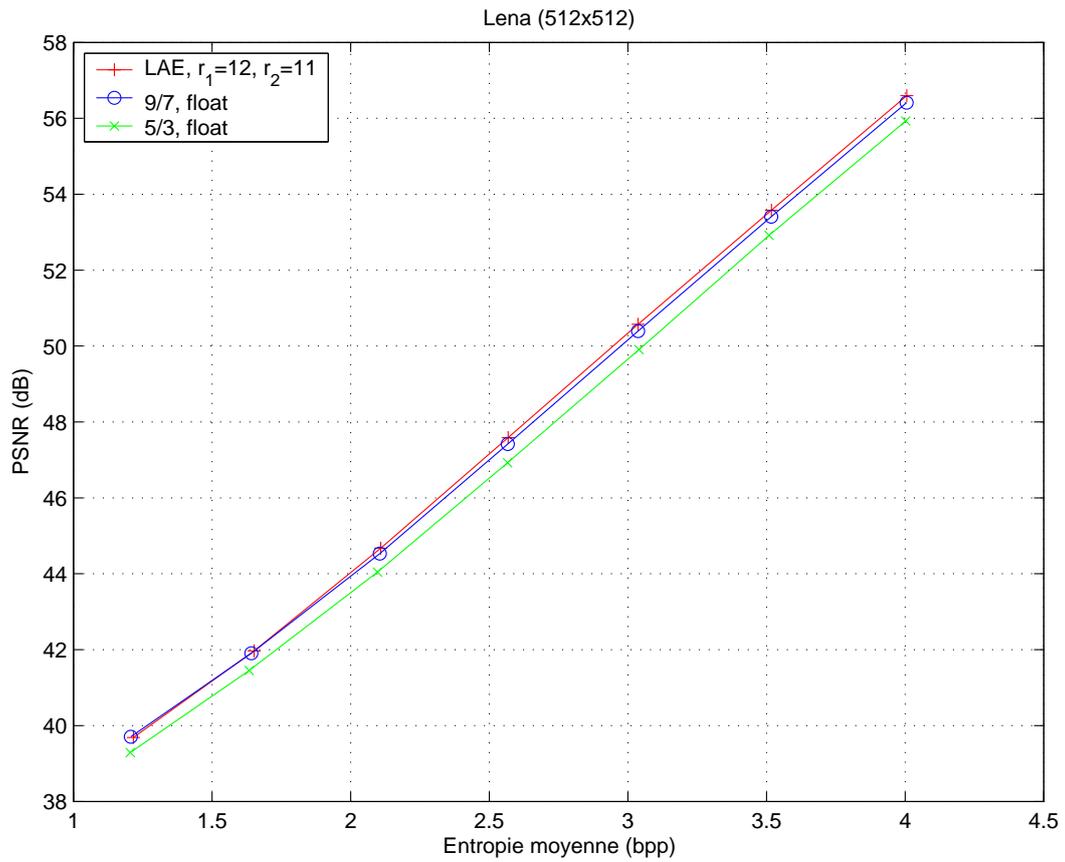
- [104] Sweldens W., “The lifting scheme : A custom-design construction of biorthogonal wavelets”, *Appl. Comput. Harmon. Analysis*, 3(2) :186–200, 1996.
- [105] Sweldens W., “The lifting scheme : A construction of second generation wavelets”, *SIAM J. Math. Analysis*, 29(2) :511–546, 1997.
- [106] Taubman D., “High performance scalable image compression with EBCOT”, *IEEE Trans. on Image Proc.*, 9(7) :1158–1170, 2000.
- [107] Tekalp A. M., Ozkan M. K. and Erdem A. T., “Image Modelling Using Higher Order Statistics with Application to Predictive Image Coding”, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, pp. 1893-1896, 1990.
- [108] Tsatsanis M. K. et Giannakis G. B., “Principal component filter banks for optimal multiresolution analysis”, *IEEE Trans. on Signal Proc.*, 43 :1766–1777, août 1995.
- [109] Tuqan J. et Vaidyanathan P. P., “optimal Low-cost two channel IIR orthogonal filter bank”, *Proc. IEEE Int. Conf. Acoust. Speech, and Signal Proc.*, Munich, Avril 1997.
- [110] Unser M., “An extension of the Karhunen-Loève transform for wavelets and multiresolution analysis”, *Proc. SPIE, Mathematical Imaging : Wavelet Applications in Signal and Image Processing*, 2034 :45–56, San Diego, 1993.
- [111] Usevitch B. E., “A tutorial on modern lossy wavelet image compression : Foundations of JPEG 2000”, *IEEE Signal Processing Magazine*, pp. 22–35, sept. 2001.
- [112] Usevitch B. E., “Optimal bit allocation for biorthogonal wavelet coding”, *Proc. Data Compression Conf.*, pp. 387–395, Snowbird, UT, mars 1996.
- [113] Vaidyanathan P. P., *Multirate systems and filter banks*, Prentice Hall, 1993.
- [114] Vaidyanathan P. P., “Theory of optimal orthonormal subband coders”, *IEEE Trans. on Signal Proc.*, 46 :1528–154, juin 1998.
- [115] Vaidyanathan P. P. et Akkarakaran S., “A review of the theory and applications of optimal subband and transform coders”, *Appl. Comput. Harmonic Analysis*, 14(3) :254–289, mai 2001.
- [116] Vaidyanathan P. P. et Kirac A., “Results on optimal biorthogonal filter banks”, *IEEE Trans. on Circuits and Systems II*, 45 :932–947, 1998.
- [117] Vetterli M., “Filter banks allowing for perfect reconstruction”, *Signal Processing*, 10 :219–244, 1986.
- [118] Wallace G. K., “The JPEG Still Picture Compression Standard”, *Comm. ACM*, 34(4) :30–44, avril 1991.
- [119] Weinberger M. J., Rissanen J. J. et Arps R. B. “Applications of Universal Context Modeling to Lossless Compression of Gray-Scale Images”. *IEEE Transactions on Image Processing*, Vol. 5, No. 4, pp. 575-586, Avril 1996.
- [120] Weinberger M., Seroussi G. et Sapiro G., “LOCO-I : A Low Complexity, Context-Based, Lossless Image Compression Algorithm”, *Proc. IEEE Data Compression Conference*, mars 1996.

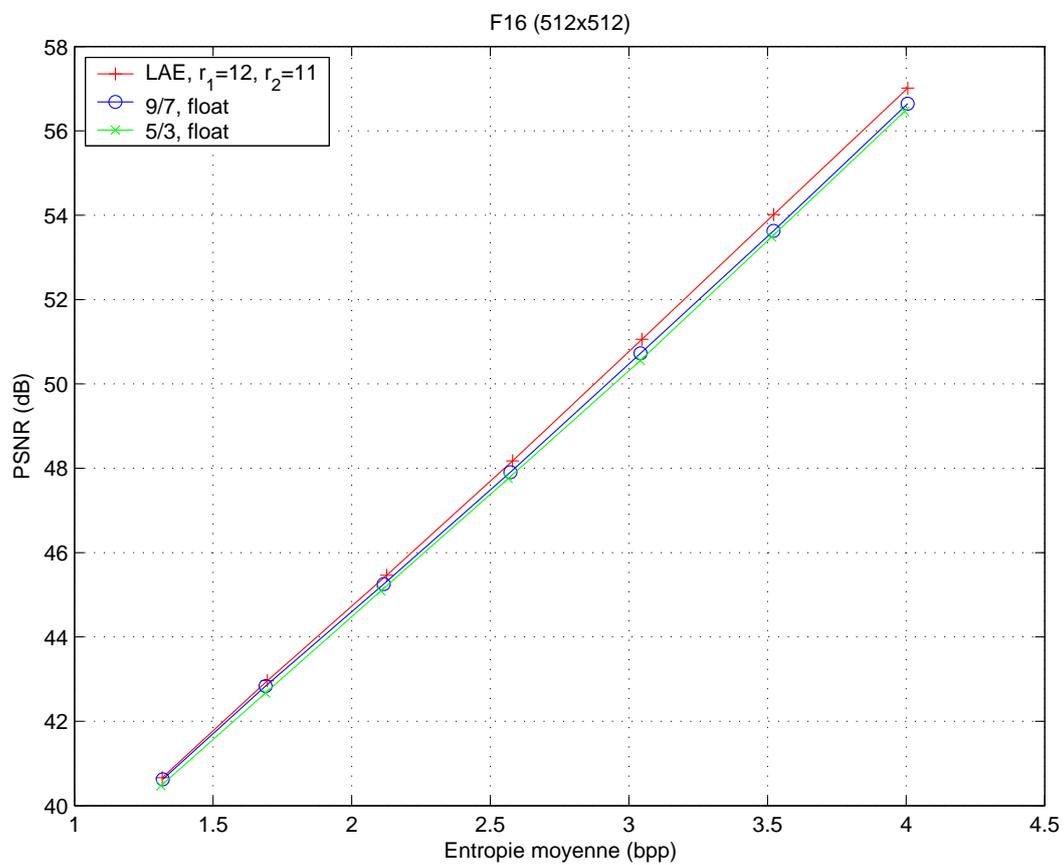
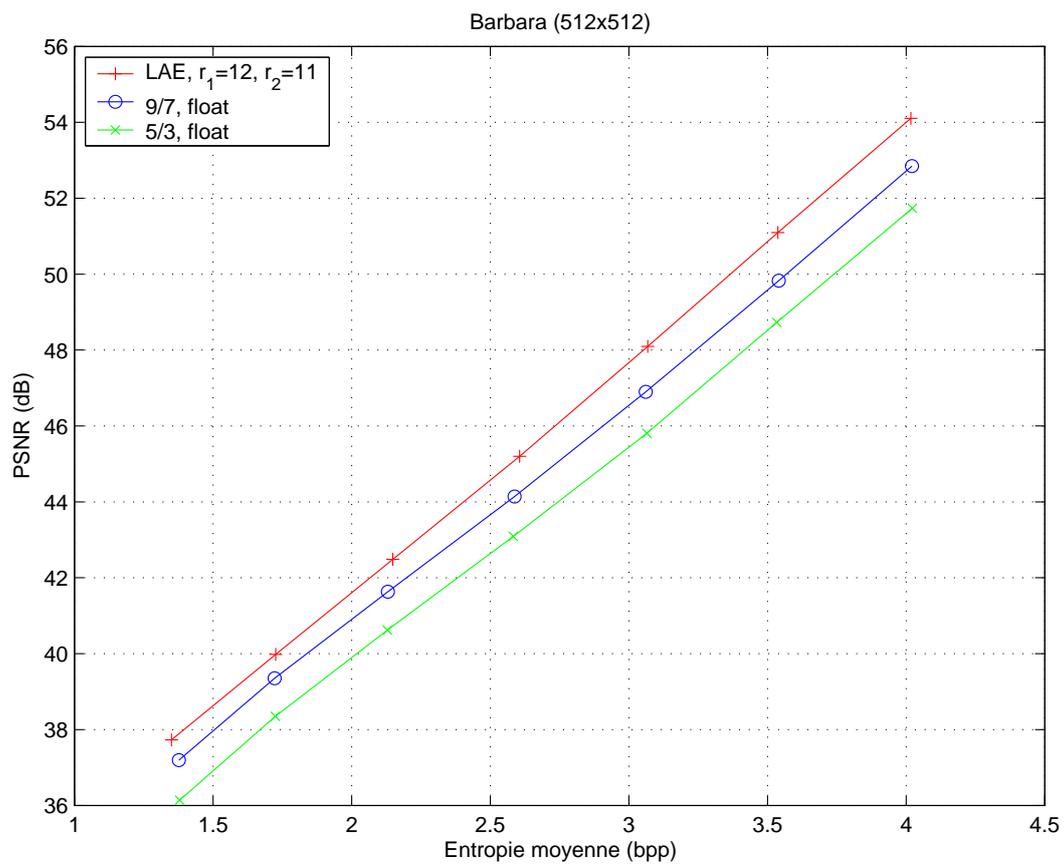
- [121] M. J. Weinberger and G. Seroussi, "Sequential Prediction and Ranking in Universal Context Modeling and Data Compression," *IEEE Transactions on Information Theory*, Vol. 43, No. 5, pp. 1697-1706, September 1997.
- [122] Witten I. H., Neal R. M. et Cleary J. G., "Arithmetic coding for data compression", *Communication of the ACM*, 30(6) :520-540, juin 1987.
- [123] Woods J. W. et Naveen T., "A filter based bit allocation for subband compression of HDTV", *IEEE Trans. on Image Proc.*, 1(3) :436-440, juil. 1992.
- [124] Wu X. et Memon N., "Context-based, Adaptive, Lossless Image Coding", *IEEE Trans. on Communications*, 45(4) :437-444, avril 1997.
- [125] Xiong Z., Ramachandran K. et Orchard M. T., "Space-Frequency Quantization for Wavelet Image Coding", *IEEE Trans. on Image Proc.*, 6(5) :677-693, mai 1997.
- [126] Xuan B. et Banberger R. H., "Multi-dimensional paraunitary principal component filter banks", *Proc. of IEEE Int. Conf. on Acoust. Speech and Signal Processing*, pp. 1488-1491, Detroit, juin 1995.
- [127] Youlal H., Janati-Idrissi M. et Najim M., *Modélisation paramétrique en traitement d'images*, Masson, 1994.
- [128] Zandi A., Allen J. D., Schwartz E. L. et Boliek M., "CREW : Compression with Reversible Embedded Wavelets", *Proc. of Data Compression Conference*, pp. 212-221, 1995.
- [129] Zhuang W., "RLS algorithm with variable forgetting factor for decision feedback equalizer over time-variant fading channels", *Wireless Personal Communications*, 8(1) :15-29, août 1998.
- [130] J. Ziv et A. Lempel. "A universal algorithm for data compression". *IEEE Transactions on Information Theory*, vol. 23(3) : 337-343, 1977.
- [131] J. Ziv et A. Lempel. "Compression of individual sequences via variable rate coding". *IEEE Transactions on Information Theory*, vol. 24(5) : 530-536, 1978.

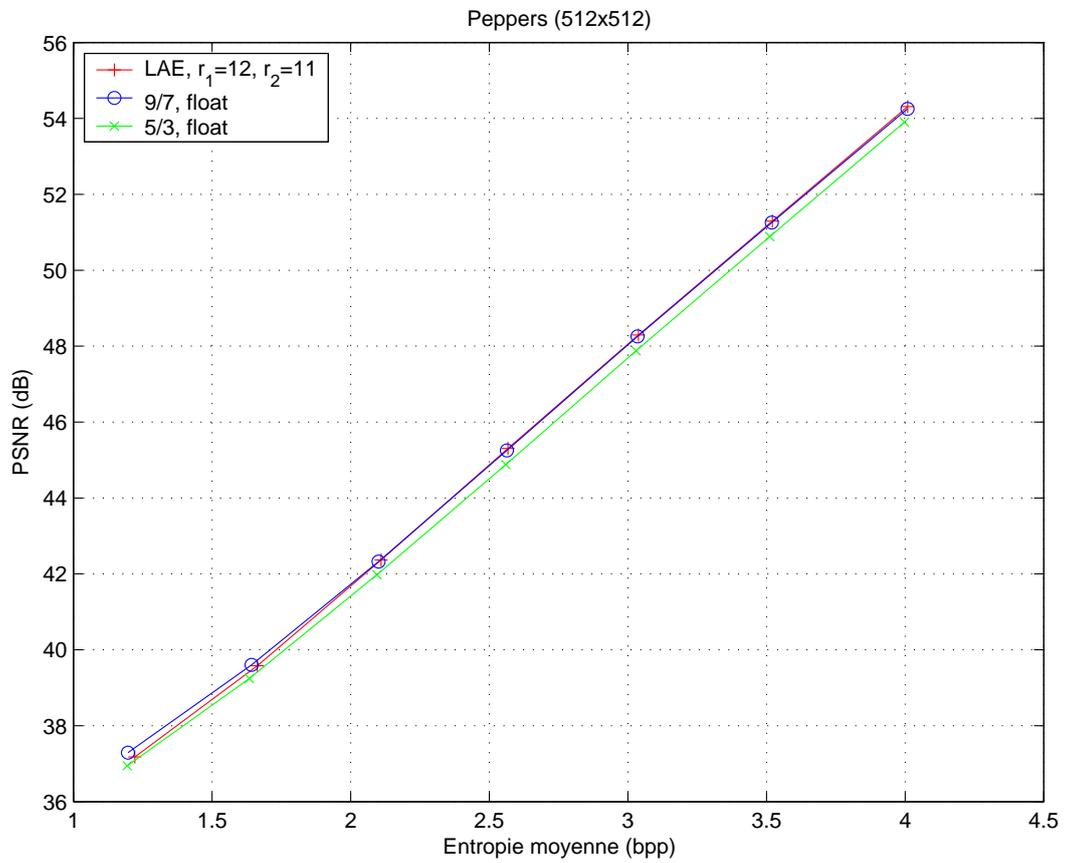
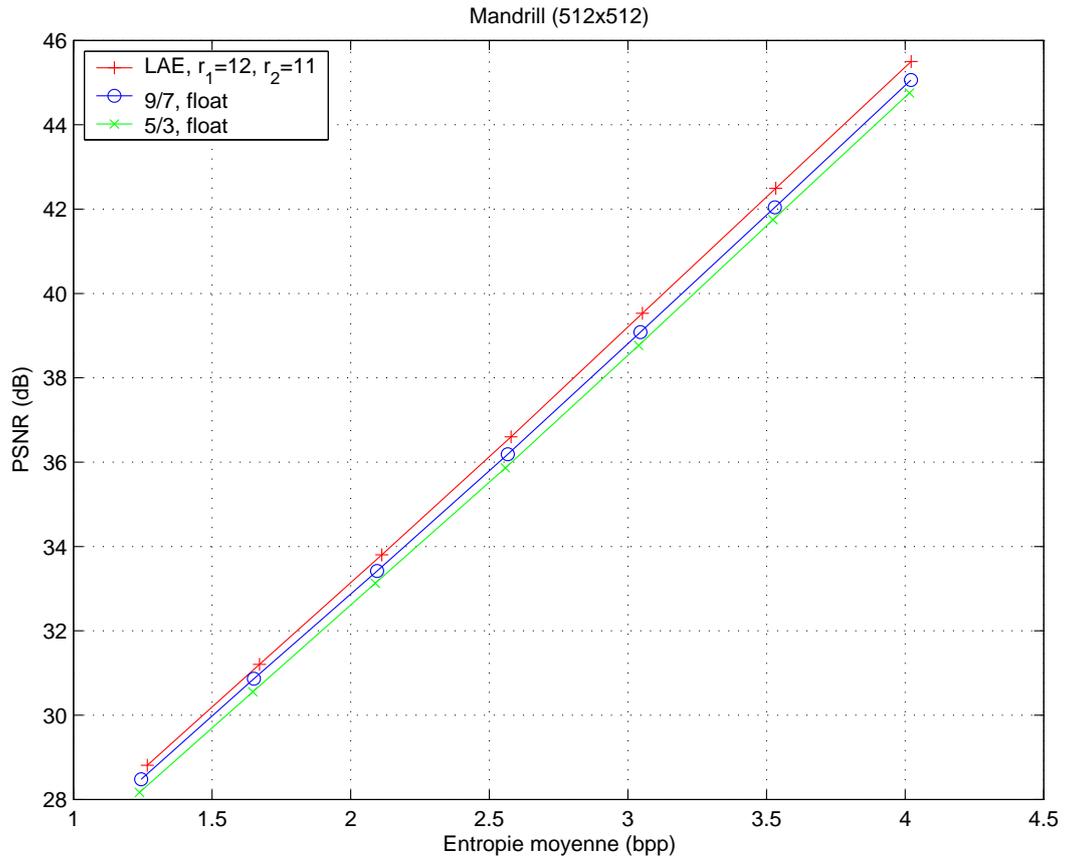
ANNEXE A

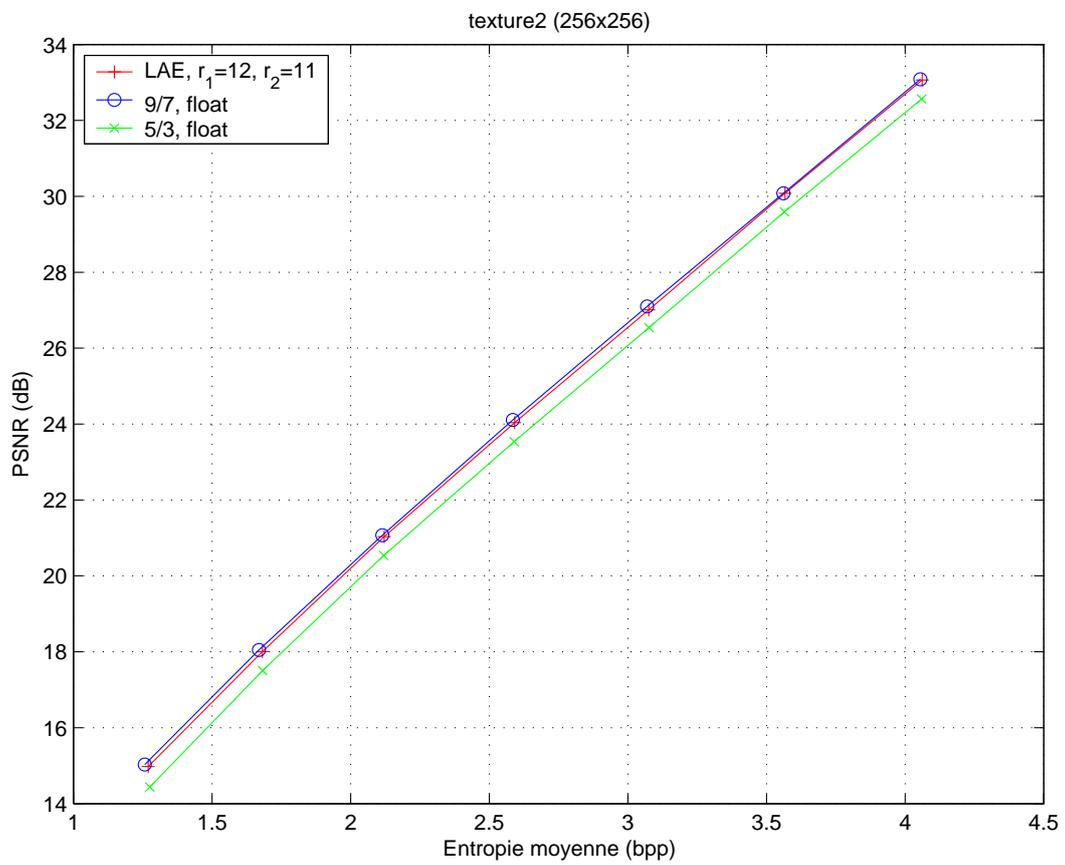
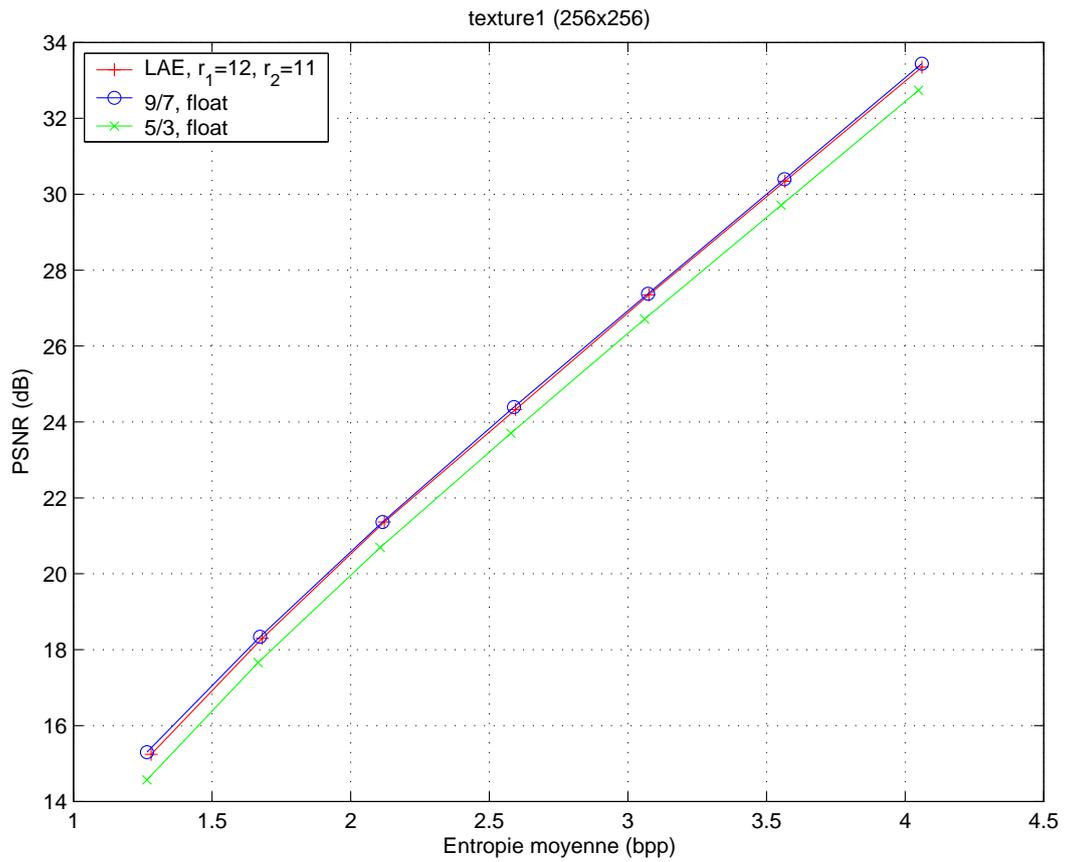
Courbes débit/distorsion relatives à la section 4.4.4

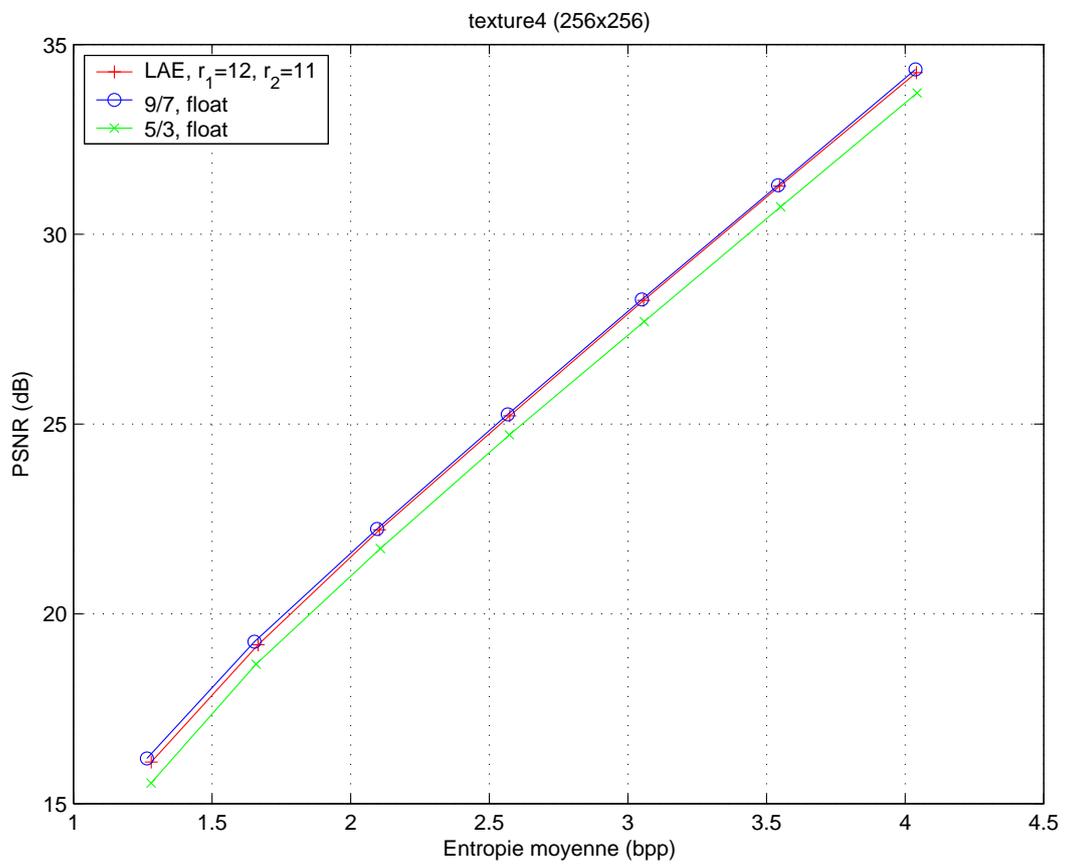
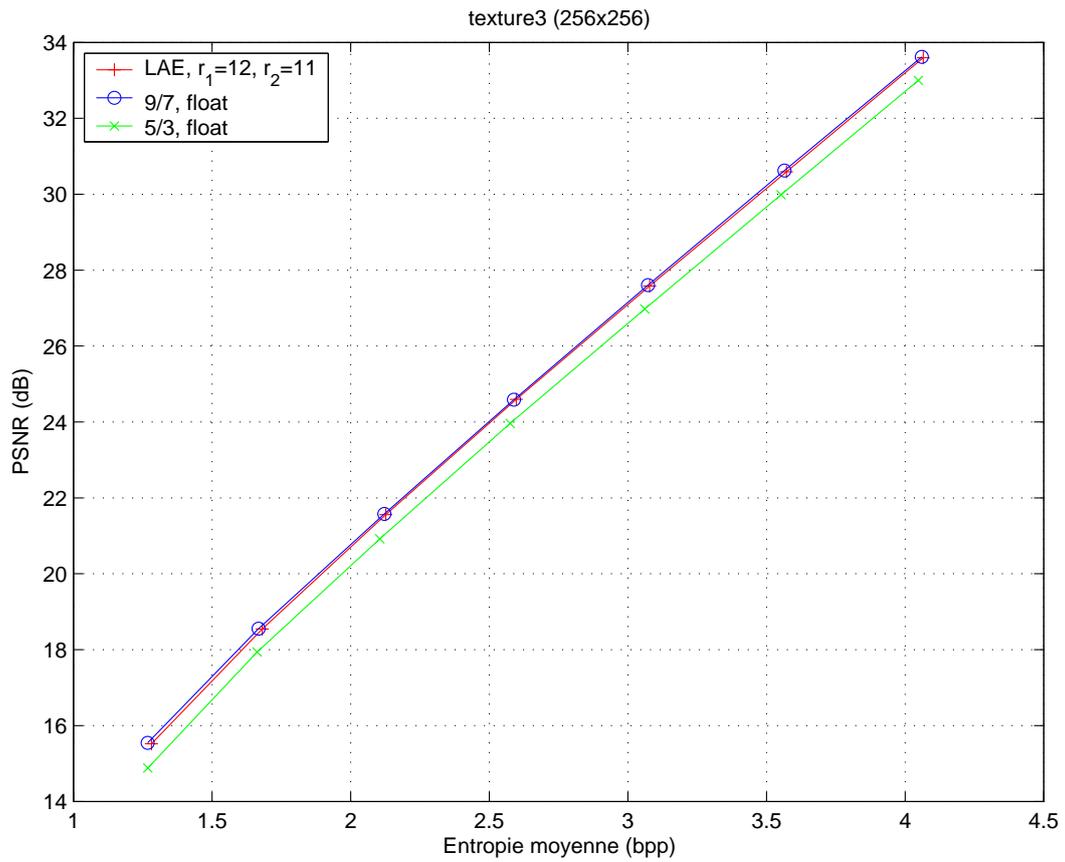
Dans cette annexe sont présentées les courbes expérimentales PSNR versus entropie d'ordre 1 moyenne sont présentées de la méthode LAE avec $r_1 = 12$, $r_2 = 11$ et les transformations (9,7) et (5,3) des images de différents types. Les images sont décomposées avec trois niveaux de décomposition.

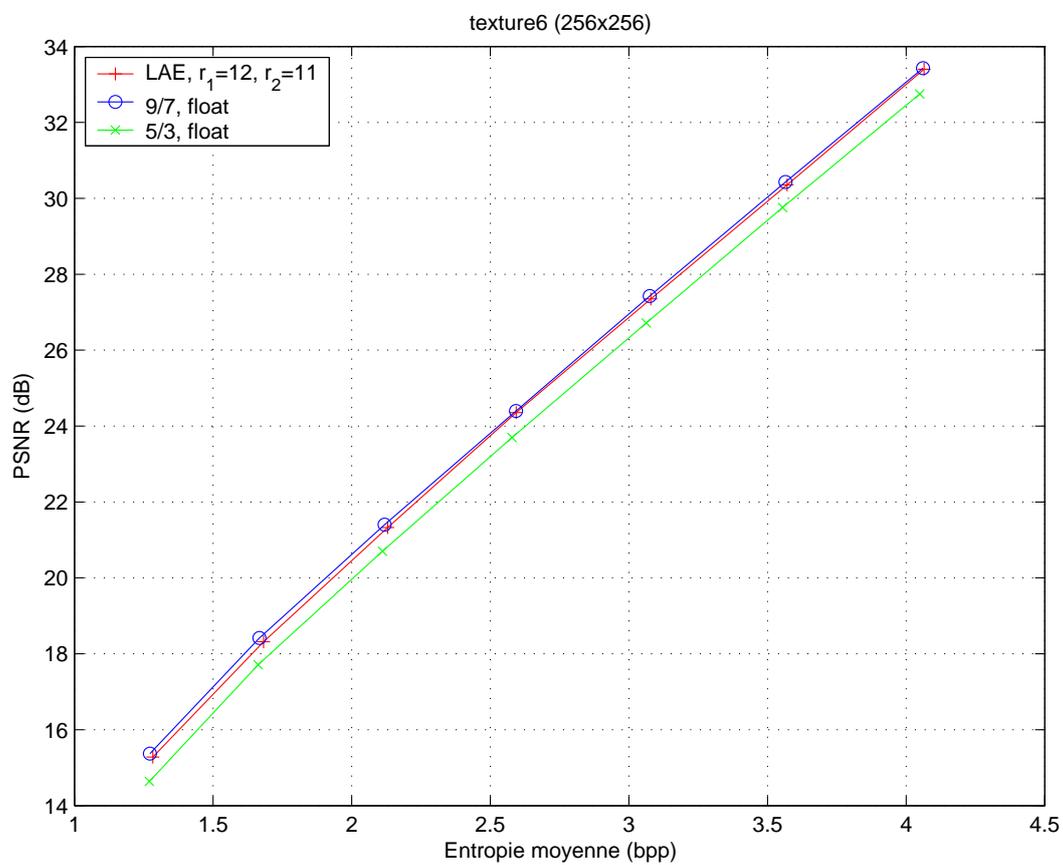
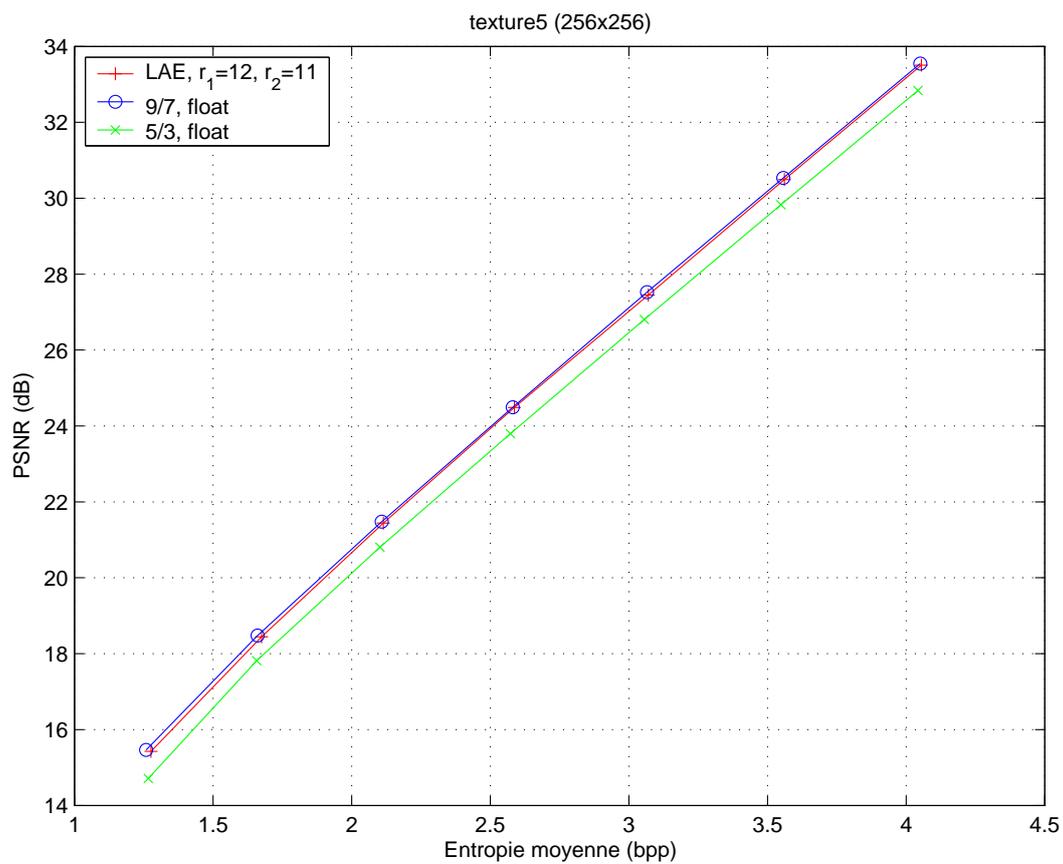


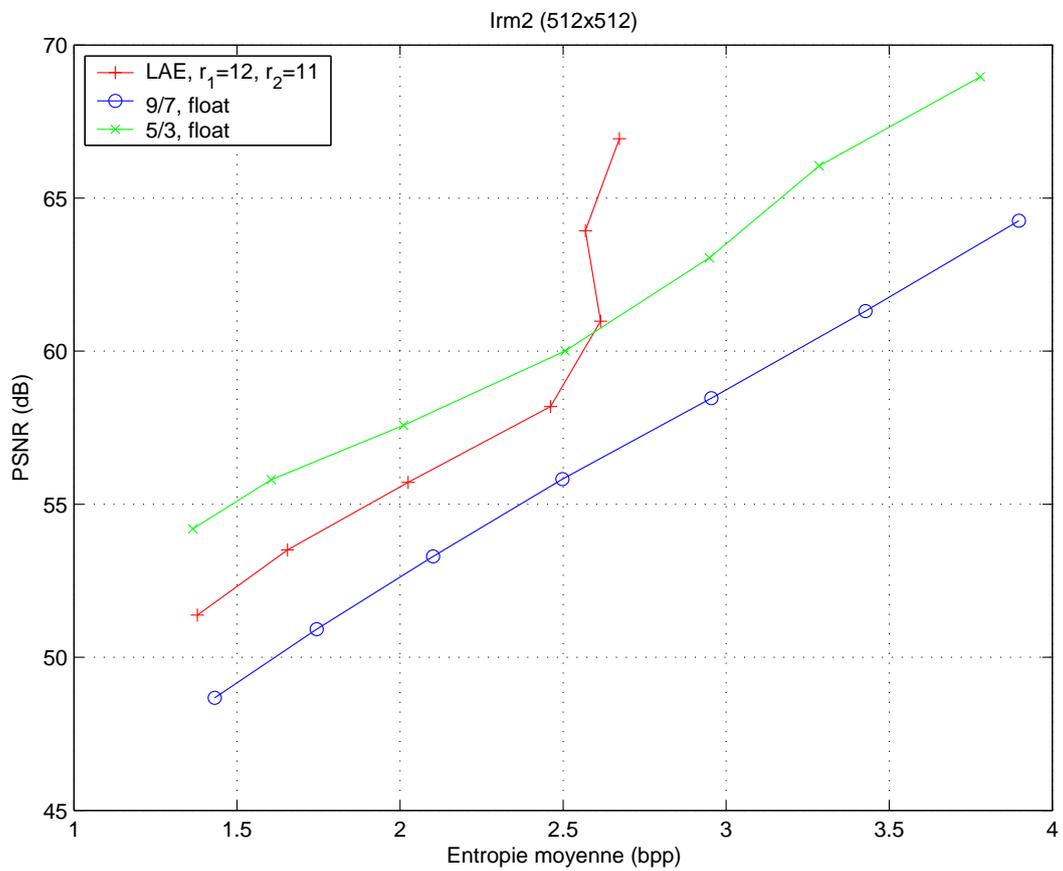
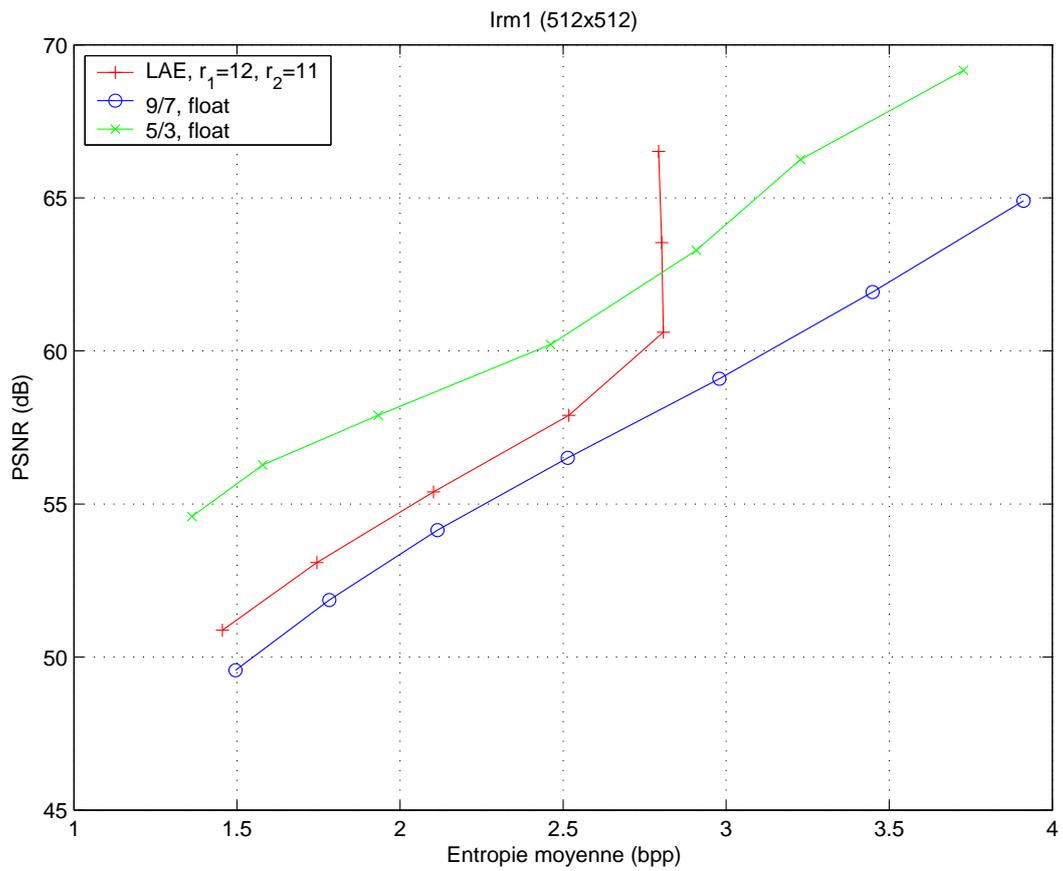


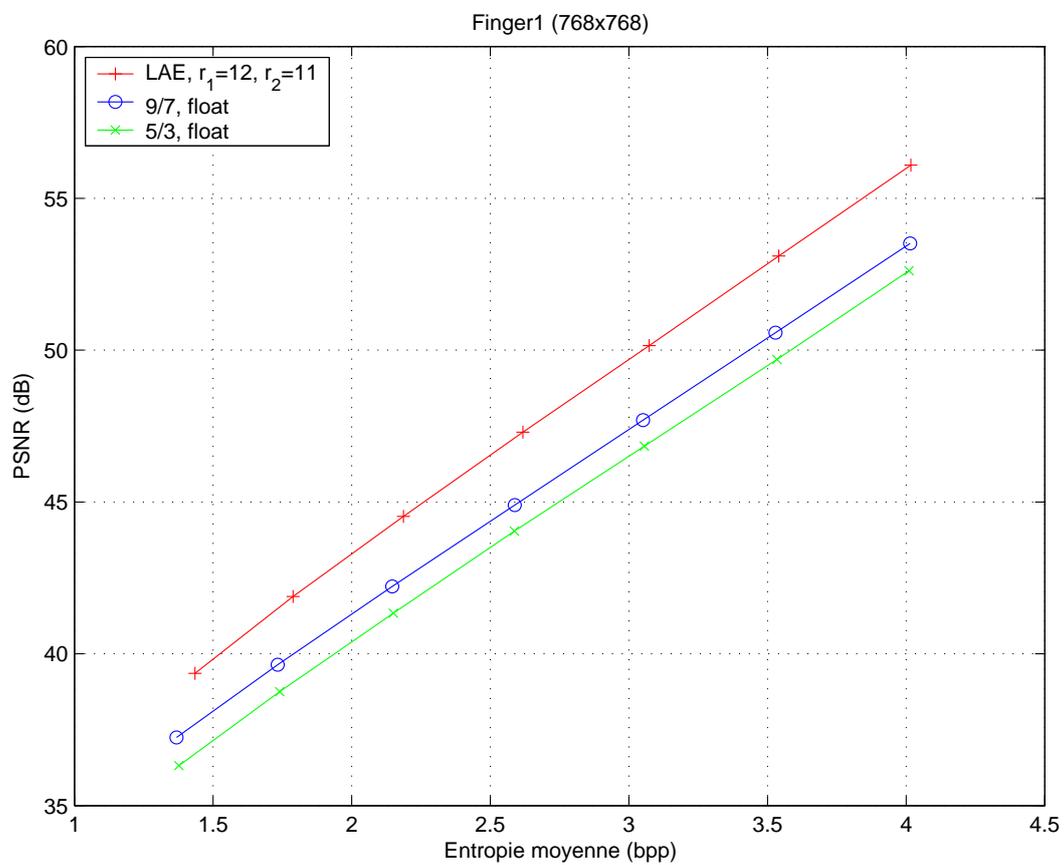
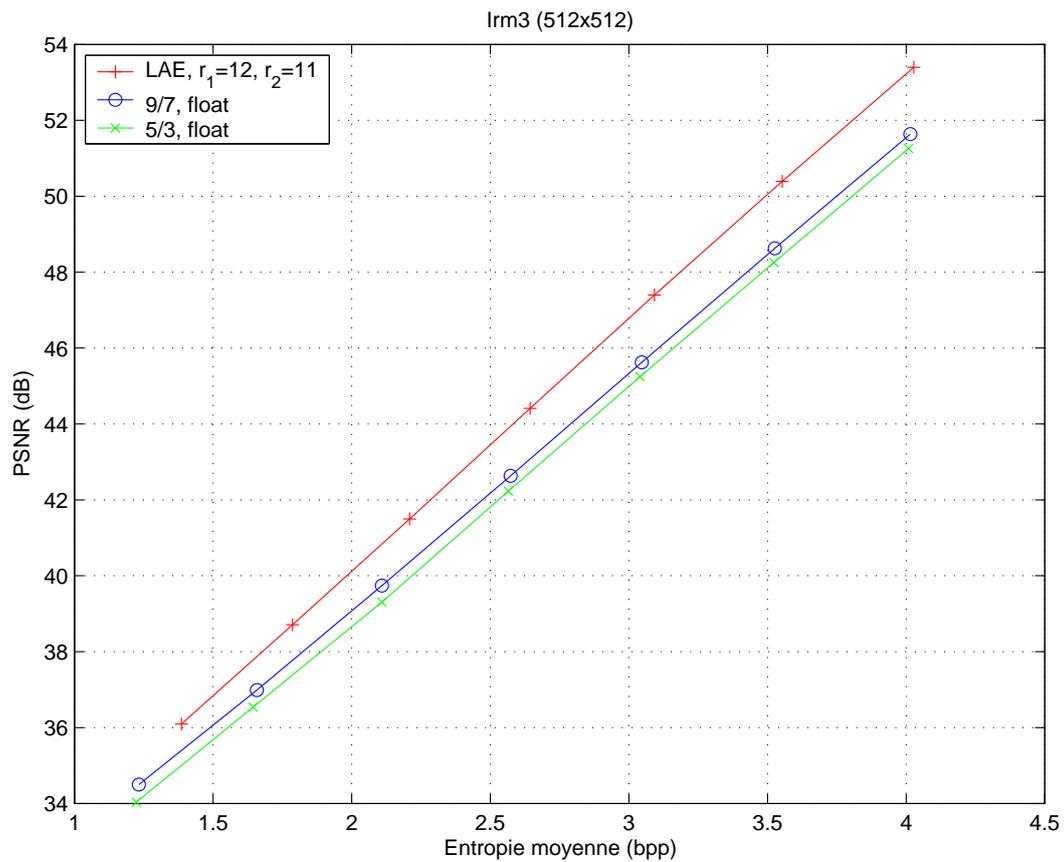


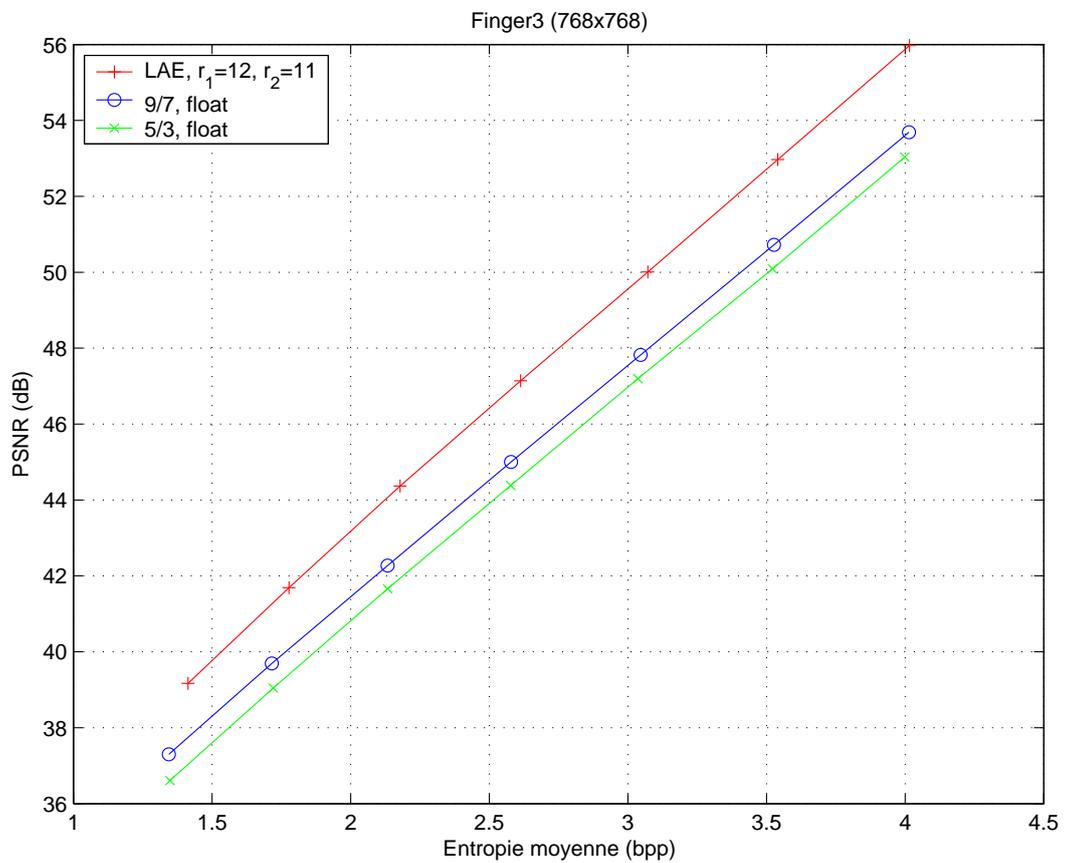
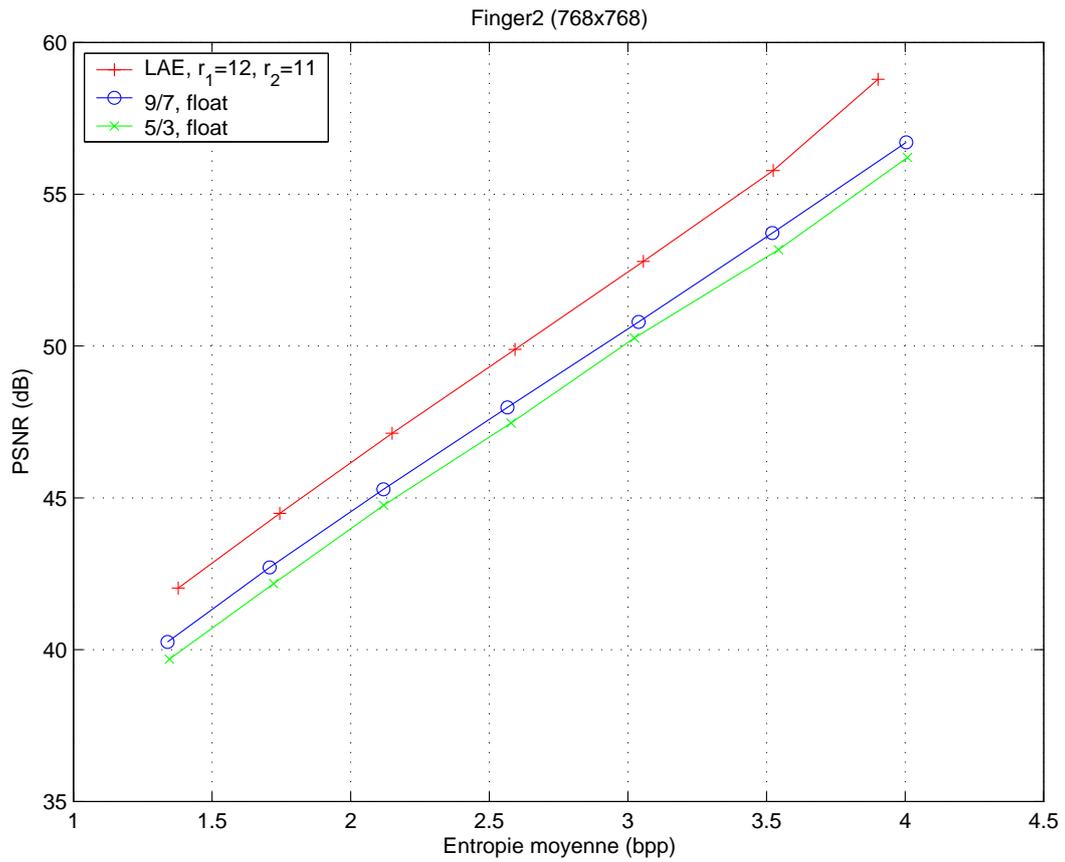












ANNEXE B

Images utilisées dans les tests

B.1 Les mages naturelles



Lena, 512 × 512, 8 bpp



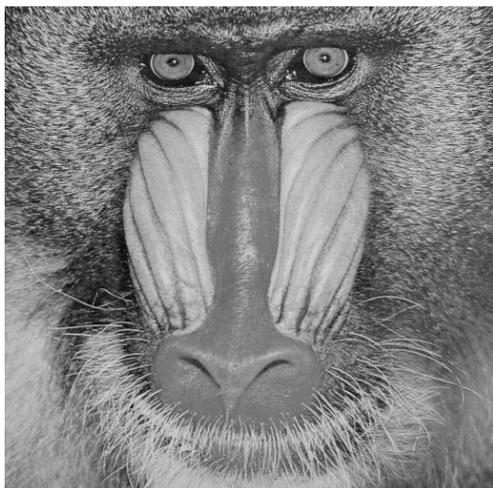
Goldhill, 512 × 512, 8 bpp



Barbara, 512 × 512, 8 bpp



F16, 512 × 512, 8 bpp



Mandrill, 512×512 , 8 bpp



Peppers, 512×512 , 8 bpp



Boat, 512×512 , 8 bpp



Bridge, 512×512 , 8 bpp



Couple, 512×512 , 8 bpp



Elaine, 512×512 , 8 bpp



Man, 1024×1024 , 8 bpp



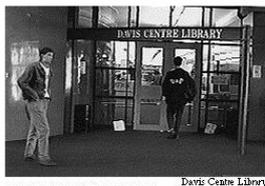
Tank, 512×512 , 8 bpp



Truck, 512×512 , 8 bpp



Zelda, 512×512 , 8 bpp



Davis Centre Library



University May and Design Library



Dana Porter Library

**University of Waterloo
Electronic Library**

Truck, 352×464 , 8 bpp

B.2 Images médicales IRM



AH1, 512 × 512, 8 bpp



AH2, 512 × 512, 8 bpp



AH3, 512 × 512, 8 bpp



AH4, 512 × 512, 8 bpp



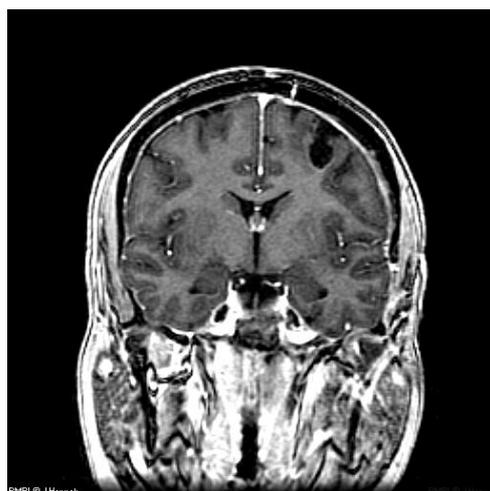
AH5, 512 × 512, 8 bpp



AH6, 512 × 512, 8 bpp



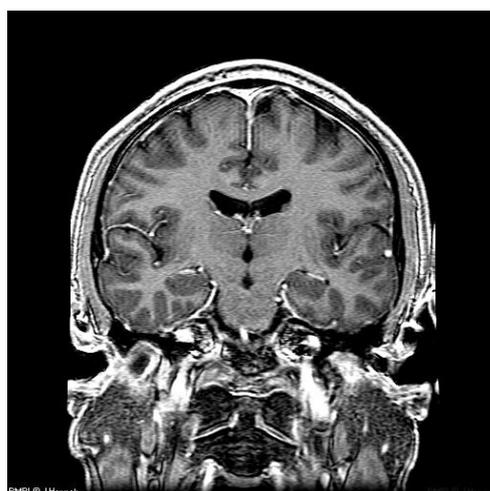
AH7 (IRM6), 512 × 512, 8 bpp



AH8, 512 × 512, 8 bpp



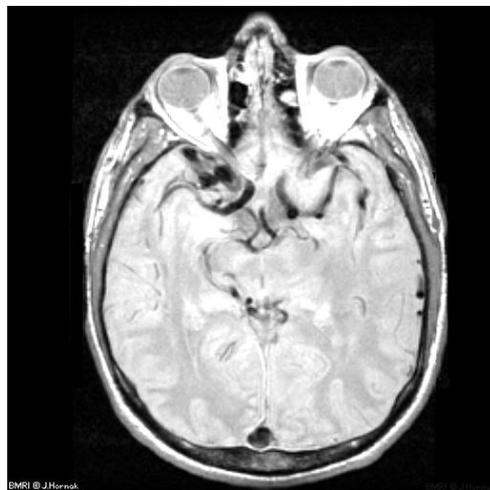
AH9 (IRM1), 512 × 512, 8 bpp



AH10 (IRM5), 512 × 512, 8 bpp



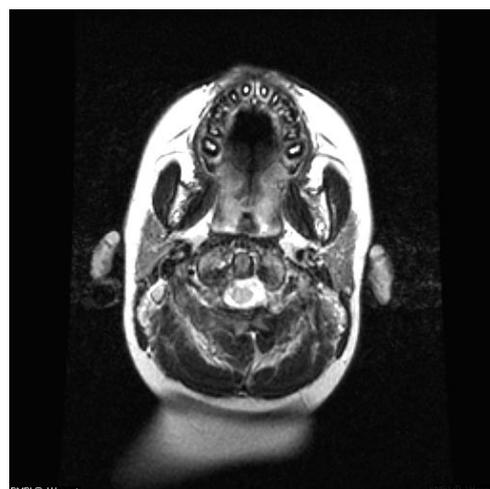
AH11 (IRM2), 512 × 512, 8 bpp



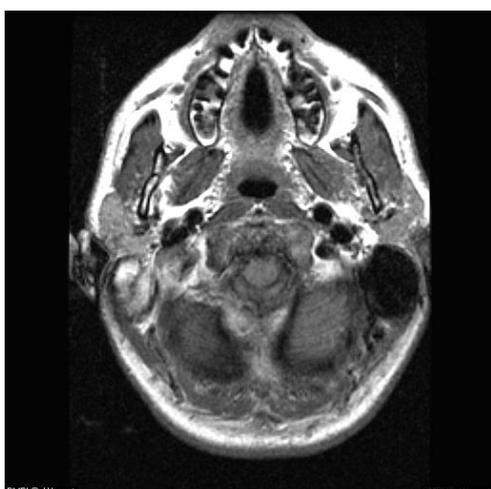
AH12, 512 × 512, 8 bpp



AH13, 512 × 512, 8 bpp



AH14, 512 × 512, 8 bpp



AH15, 512 × 512, 8 bpp



AH16 (IRM4), 512 × 512, 8 bpp



AH17 (IRM3), 512 × 512, 8 bpp

B.3 Images satellitaires



Pentagone, 1024×1024 , 8 bpp



Sanfrancisco, 1024×1024 , 8 bpp



Okland, 1024×1024 , 8 bpp



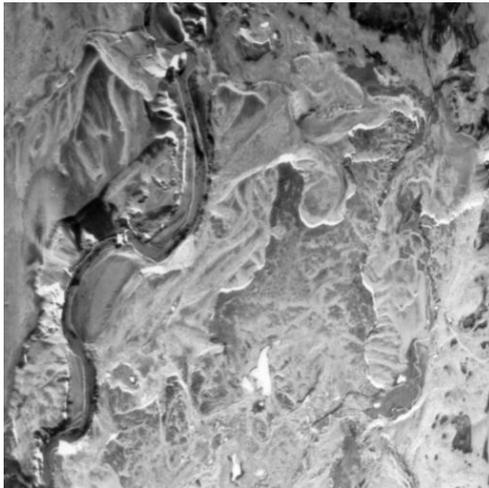
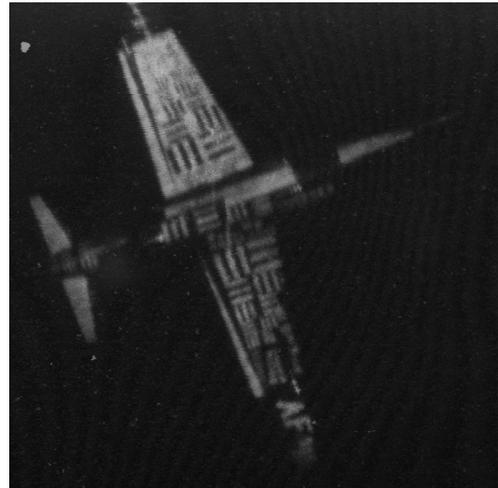
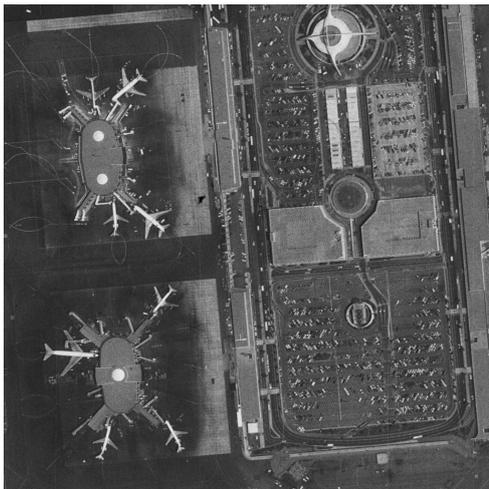
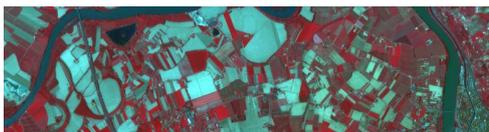
Washington, 512×512 , 8 bpp



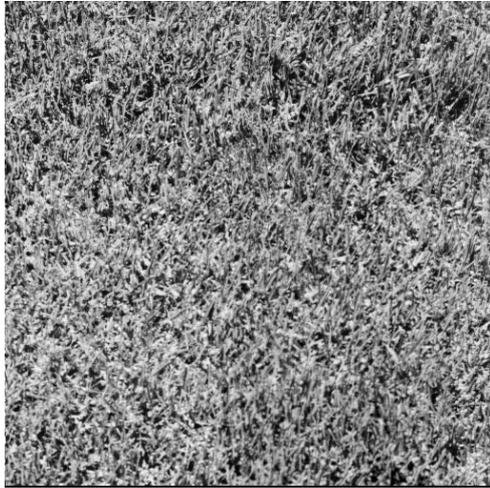
Toulouse, 512×512 , 8 bpp



Gênes, 512×512 , 8 bpp

Mars, 512×512 , 8 bppAirplane, 1024×1024 , 8 bppAirport, 1024×1024 , 8 bppMoissac, 802×212 , 8 bppMontpellier, 1336×856 , 8 bpp

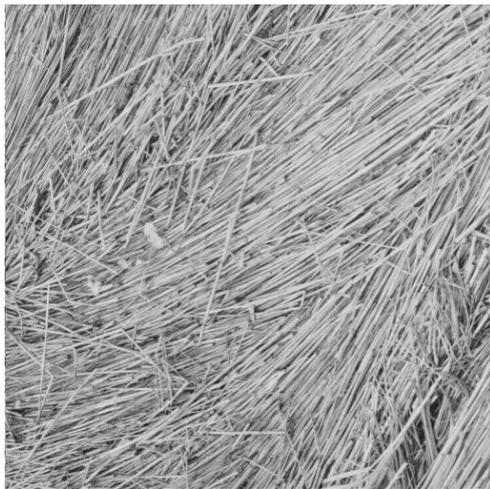
B.4 Images de textures et empreintes digitales



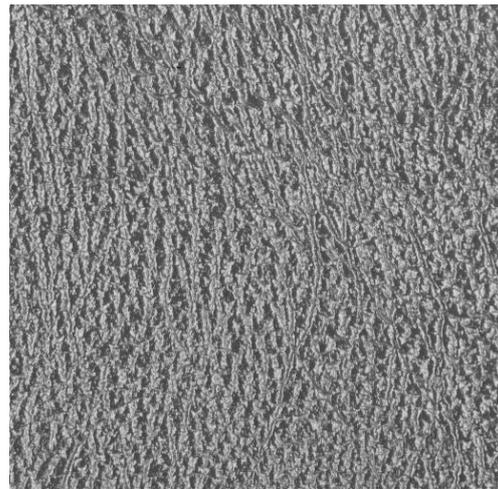
Texture1, 512 × 512, 8 bpp



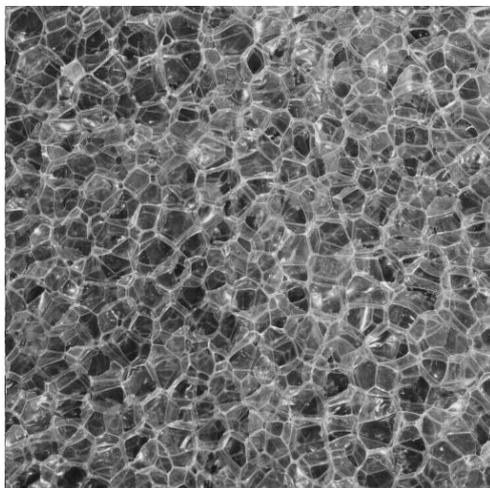
Texture2, 512 × 512, 8 bpp



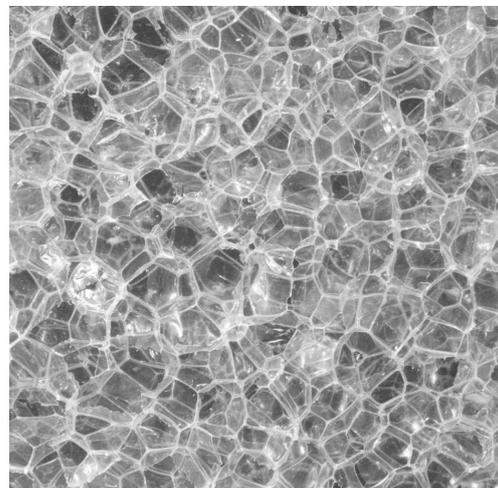
Texture3, 512 × 512, 8 bpp



Texture4, 512 × 512, 8 bpp



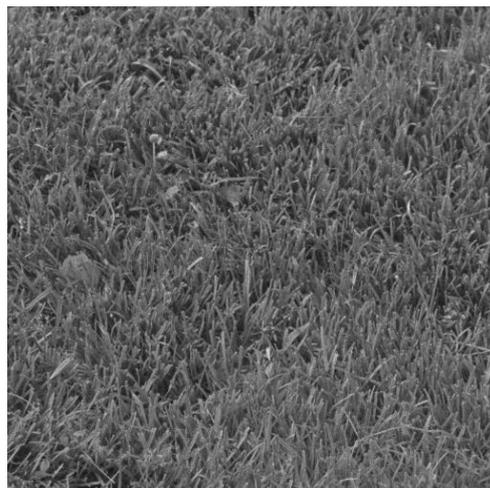
Texture5, 512 × 512, 8 bpp



Texture6, 1024 × 1024, 8 bpp



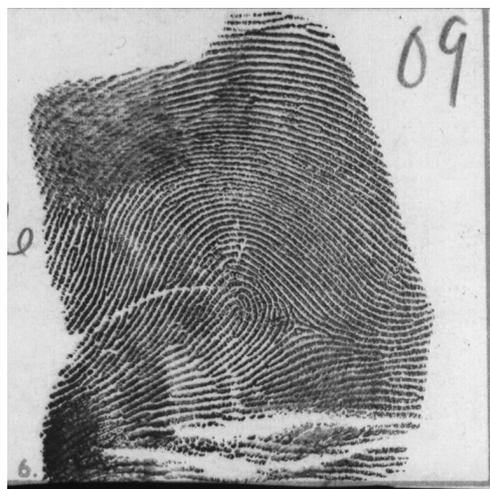
Texture7, 1024 × 1024, 8 bpp



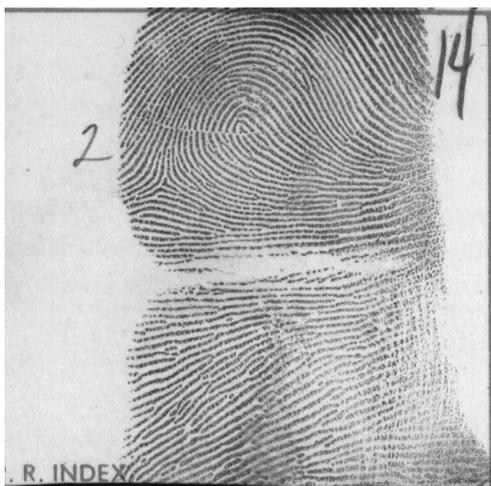
Texture8, 512 × 512, 8 bpp



Finger1, 768 × 768, 8 bpp



Finger2, 768 × 768, 8 bpp



Finger3, 768 × 768, 8 bpp

ANNEXE C

Transformations en ondelettes de \mathbb{Z} dans \mathbb{Z}

Ces transformations sont appliquées sur un signal $x(n)$ pour donner les deux signaux $x_\ell(n)$ et $x_h(n)$. Le premier est un signal d'approximation et le deuxième un signal de détails. Les notations que nous avons adoptées pour désigner les transformations en ondelettes sont celles données par Calderbank et *al.* dans [28]. Elles diffèrent de celles utilisées par Adams et Kossentini dans [9].

C.1 La transformation S

Transformation directe

$$x_\ell(n) = \left\lfloor \frac{x(2n) + x(2n+1)}{2} \right\rfloor \quad (\text{C.1a})$$

$$x_h(n) = x(2n) - x(2n+1). \quad (\text{C.1b})$$

Transformation inverse

$$x(2n+1) = x_\ell(n) - \left\lfloor \frac{x_h(n)}{2} \right\rfloor \quad (\text{C.2a})$$

$$x(2n) = x_h(n) + x(2n+1). \quad (\text{C.2b})$$

C.2 La transformation (2, 2) ou 5/3

Cette transformation est notée 5/3 dans [9]. Transformation directe

$$x_h(n) = x(2n+1) - \left\lfloor \frac{x(2n) + x(2n+2) + 1}{2} \right\rfloor \quad (\text{C.3a})$$

$$x_\ell(n) = x(2n) + \left\lfloor \frac{x_h(n) + x_h(n-1) + 2}{4} \right\rfloor. \quad (\text{C.3b})$$

Transformation inverse

$$x(2n) = x_\ell(n) - \left\lfloor \frac{x_h(n) + x_h(n-1) + 2}{4} \right\rfloor \quad (\text{C.4a})$$

$$x(2n+1) = x_h(n) + \left\lfloor \frac{x(2n) + x(2n+2) + 1}{2} \right\rfloor. \quad (\text{C.4b})$$

C.3 La transformation (4, 2)

Transformation directe

$$x_h(n) = x(2n+1) - \left\lfloor \frac{9}{16} [x(2n) + x(2n+2)] - \frac{1}{16} [x(2n-2) + x(2n+4)] + \frac{1}{2} \right\rfloor \quad (\text{C.5a})$$

$$x_\ell(n) = x(2n) + \left\lfloor \frac{1}{4} [x_h(n-1) + x_h(n)] + \frac{1}{2} \right\rfloor \quad (\text{C.5b})$$

Transformation inverse

$$x(2n) = x_\ell(n) - \left\lfloor \frac{1}{4} [x_h(n-1) + x_h(n)] + \frac{1}{2} \right\rfloor \quad (\text{C.6a})$$

$$x(2n+1) = x_h(n) + \left\lfloor \frac{9}{16} [x(2n) + x(2n+2)] - \frac{1}{16} [x(2n-2) + x(2n+4)] + \frac{1}{2} \right\rfloor \quad (\text{C.6b})$$

C.4 La transformation (2, 4)

Transformation directe

$$x_h(n) = x(2n+1) - \left\lfloor \frac{1}{2} [x(2n) + x(2n+2)] + \frac{1}{2} \right\rfloor \quad (\text{C.7a})$$

$$x_\ell(n) = x(2n) + \left\lfloor \frac{19}{64} [x_h(n-1) + x_h(n)] - \frac{3}{64} [x_h(n-2) + x_h(n+1)] + \frac{1}{2} \right\rfloor \quad (\text{C.7b})$$

Transformation inverse

$$x(2n) = x_\ell(n) - \left\lfloor \frac{19}{64} [x_h(n-1) + x_h(n)] - \frac{3}{64} [x_h(n-2) + x_h(n+1)] + \frac{1}{2} \right\rfloor \quad (\text{C.8a})$$

$$x(2n+1) = x_h(n) + \left\lfloor \frac{1}{2} [x(2n) + x(2n+2)] + \frac{1}{2} \right\rfloor \quad (\text{C.8b})$$

C.5 La transformation (2+2,2)

Cette transformation est notée 5/11-C dans [9]. Transformation directe

$$x_h^{(1)}(n) = x(2n+1) - \left[\frac{1}{2} (x(2n) + x(2n+2)) + \frac{1}{2} \right] \quad (\text{C.9a})$$

$$x_\ell(n) = x(2n) + \left[\frac{1}{4} (x_h^{(1)}(n-1) + x_h^{(1)}(n)) + \frac{1}{2} \right] \quad (\text{C.9b})$$

$$\begin{aligned} x_h(n) = x_h^{(1)}(n) - & \left[\alpha \left(-\frac{1}{2} x_\ell(n-1) + x_\ell(n) - \frac{1}{2} x_\ell(n+1) \right) \right. \\ & \left. + \beta \left(-\frac{1}{2} x_\ell(n) + x_\ell(n+1) - \frac{1}{2} x_\ell(n+2) \right) + \gamma x_h^{(1)}(n) + \frac{1}{2} \right] \quad (\text{C.9c}) \end{aligned}$$

avec $\alpha = 1/8$, $\beta = 1/8$, $\gamma = 0$.

Transformation inverse

$$\begin{aligned} x_h^{(1)}(n) = x_h(n) + & \left[\alpha \left(-\frac{1}{2} x_\ell(n-1) + x_\ell(n) - \frac{1}{2} x_\ell(n+1) \right) \right. \\ & \left. + \beta \left(-\frac{1}{2} x_\ell(n) + x_\ell(n+1) - \frac{1}{2} x_\ell(n+2) \right) + \gamma \right] \quad (\text{C.10a}) \end{aligned}$$

$$x(2n+1) = x_h^{(1)}(n) + \left[\frac{1}{2} (x(2n) + x(2n+2)) + \frac{1}{2} \right] \quad (\text{C.10b})$$

$$x(2n) = x_\ell(n) - \left[\frac{1}{4} (x_h^{(1)}(n-1) + x_h^{(1)}(n)) + \frac{1}{2} \right] x_h^{(1)}(n) + \frac{1}{2} \quad (\text{C.10c})$$

avec $\alpha = 1/8$, $\beta = 1/8$, $\gamma = 0$.

C.6 La transformation (4,4)

Cette transformation est notée 13/7-T dans [9]. Transformation directe

$$\begin{aligned} x_h(n) = x(2n+1) - & \left[\frac{9}{16} [x(2n) + x(2n+2)] \right. \\ & \left. - \frac{1}{16} [x(2n-2) + x(2n+4)] + \frac{1}{2} \right] \quad (\text{C.11a}) \end{aligned}$$

$$\begin{aligned} x_\ell(n) = x(2n) + & \left[\frac{9}{32} [x_h(n-1) + x_h(n)] \right. \\ & \left. - \frac{1}{32} [x_h(n-2) + x_h(n+1)] + \frac{1}{2} \right]. \quad (\text{C.11b}) \end{aligned}$$

Transformation inverse

$$\begin{aligned} x(2n) = x_\ell(n) - & \left[\frac{9}{32} [x_h(n-1) + x_h(n)] \right. \\ & \left. - \frac{1}{32} [x_h(n-2) + x_h(n+1)] + \frac{1}{2} \right] \quad (\text{C.12a}) \end{aligned}$$

$$x(2n+1) = x_h(n) + \left[\frac{9}{16} [x(2n) + x(2n+2)] - \frac{1}{16} [x(2n-2) + x(2n+4)] + \frac{1}{2} \right] \quad (\text{C.12b})$$

C.7 La transformation (6,2)

Transformation directe

$$x_h(n) = x(2n+1) - \left[\frac{75}{128} [x(2n) + x(2n+2)] - \frac{25}{256} [x(2n-2) + x(2n+4)] + \frac{3}{256} [x(2n-4) + x(2n+6)] + \frac{1}{2} \right] \quad (\text{C.13a})$$

$$x_\ell(n) = x(2n) + \left[\frac{1}{4} (x_h(n-1) + x_h(n)) + \frac{1}{2} \right]. \quad (\text{C.13b})$$

Transformation inverse

$$x(2n) = x_\ell(n) - \left[\frac{1}{4} (x_h(n-1) + x_h(n)) + \frac{1}{2} \right] \quad (\text{C.14a})$$

$$x(2n+1) = x_h(n) + \left[\frac{75}{128} [x(2n) + x(2n+2)] - \frac{25}{256} [x(2n-2) + x(2n+4)] + \frac{3}{256} [x(2n-4) + x(2n+6)] + \frac{1}{2} \right]. \quad (\text{C.14b})$$

C.8 La transformation 9 – 7

Transformation directe

$$x_h^{(1)}(n) = x(2n+1) + \left[\alpha [x(2n) + x(2n+2)] + \frac{1}{2} \right] \quad (\text{C.15a})$$

$$x_\ell^{(1)}(n) = x(2n) + \left[\beta [x_h^{(1)}(n) + x_h^{(1)}(n-1)] + \frac{1}{2} \right] \quad (\text{C.15b})$$

$$x_h^{(2)}(n) = x_h^{(1)}(n) + \left[\gamma [x_\ell^{(1)}(n) + x_\ell^{(1)}(n+1)] + \frac{1}{2} \right] \quad (\text{C.15c})$$

$$x_\ell^{(2)}(n) = x_\ell^{(1)}(n) + \left[\delta [x_h^{(2)}(n) + x_h^{(2)}(n-1)] + \frac{1}{2} \right] \quad (\text{C.15d})$$

$$x_h^{(3)}(n) = x_\ell^{(2)}(n) + x_h^{(2)}(n) \quad (\text{C.15e})$$

$$x_\ell^{(3)}(n) = x_\ell^{(2)}(n) \quad (\text{C.15f})$$

$$x_h^{(4)}(n) = x_h^{(3)}(n) \quad (\text{C.15g})$$

$$x_\ell^{(4)}(n) = x_\ell^{(3)}(n) + \left[(K-1)x_h^{(3)}(n) + \frac{1}{2} \right] \quad (\text{C.15h})$$

$$x_h^{(5)}(n) = - \left[\frac{1}{K} x_\ell^{(4)}(n) + \frac{1}{2} \right] + x_h^{(4)}(n) \quad (\text{C.15i})$$

$$x_\ell^{(5)}(n) = x_\ell^{(4)}(n) \quad (\text{C.15j})$$

$$x_h(n) = x_h^{(5)}(n) \quad (\text{C.15k})$$

$$x_\ell(n) = x_\ell^{(5)}(n) + \left[[K - K^2]x_h^{(5)}(n) + \frac{1}{2} \right] \quad (\text{C.15l})$$

Transformation inverse

$$x_\ell^{(5)}(n) = x_\ell(n) - \left[[K - K^2]x_h^{(5)}(n) + \frac{1}{2} \right] \quad (\text{C.16a})$$

$$x_h^{(5)}(n) = x_h(n) \quad (\text{C.16b})$$

$$x_\ell^{(4)}(n) = x_\ell^{(5)}(n) \quad (\text{C.16c})$$

$$x_h^{(4)}(n) = \left[\frac{1}{K}x_\ell^{(4)}(n) + \frac{1}{2} \right] + x_h^{(5)}(n) \quad (\text{C.16d})$$

$$x_\ell^{(3)}(n) = x_\ell^{(4)}(n) - \left[(K - 1)x_h^{(3)}(n) + \frac{1}{2} \right] \quad (\text{C.16e})$$

$$x_h^{(3)}(n) = x_h^{(4)}(n) \quad (\text{C.16f})$$

$$x_\ell^{(2)}(n) = x_\ell^{(3)}(n) \quad (\text{C.16g})$$

$$x_h^{(2)}(n) = x_h^{(3)}(n) - x_\ell^{(2)}(n) \quad (\text{C.16h})$$

$$x_\ell^{(1)}(n) = x_\ell^{(2)}(n) - \left[\delta[x_h^{(2)}(n) + x_h^{(2)}(n-1)] + \frac{1}{2} \right] \quad (\text{C.16i})$$

$$x_h^{(1)}(n) = x_h^{(2)}(n) - \left[\gamma[x_\ell^{(1)}(n) + x_\ell^{(1)}(n+1)] + \frac{1}{2} \right] \quad (\text{C.16j})$$

$$x(2n) = x_\ell^{(1)}(n) - \left[\beta[x_h^{(1)}(n) + x_h^{(1)}(n-1)] + \frac{1}{2} \right] \quad (\text{C.16k})$$

$$x(2n+1) = x_h^{(1)}(n) - \left[\alpha[x(2n) + x(2n+2)] + \frac{1}{2} \right] \quad (\text{C.16l})$$

$$\alpha = -1,586134342$$

$$\beta = -0,05298011854$$

$$\gamma = 0,8829110762$$

$$\delta = 0,4435068522$$

$$K = 1,149604398$$

C.9 Les transformations S+P et TS

Transformation directe (avec un balayage croissant de n)

$$x_h^{(1)}(n) = x(2n+1) - x(2n) \quad (\text{C.17a})$$

$$x_\ell(n) = x(2n) + \left\lfloor \frac{x_h^{(1)}(n)}{2} + \frac{1}{2} \right\rfloor \quad (\text{C.17b})$$

$$x_h(n) = x_h^{(1)}(n) + \lfloor \alpha_{-1} (x_\ell(n-2) - x_\ell(n-1)) + \alpha_0 (x_\ell(n-1) - x_\ell(n)) + \alpha_1 (x_\ell(n) - x_\ell(n+1)) - \beta_1 x_h^{(1)}(n+1) \rfloor. \quad (\text{C.17c})$$

Transformation inverse (avec un balayage décroissant de n)

$$x_h^{(1)}(n) = x_h(n) - \lfloor \alpha_{-1} (x_\ell(n-2) - x_\ell(n-1)) + \alpha_0 (x_\ell(n-1) - x_\ell(n)) + \alpha_1 (x_\ell(n) - x_\ell(n+1)) - \beta_1 x_h^{(1)}(n+1) \rfloor \quad (\text{C.18a})$$

$$x(2n) = x_\ell(n) - \left\lfloor \frac{x_h^{(1)}(n)}{2} + \frac{1}{2} \right\rfloor \quad (\text{C.18b})$$

$$x(2n+1) = x_h^{(1)}(n) + x(2n) \quad (\text{C.18c})$$

Selon les valeurs de α_{-1} , α_0 , α_1 et β_1 , trois prédicteurs sont proposés :

Prédicteur/Coef.	α_{-1}	α_0	α_1	β_1
A	0	1/4	1/4	0
B	0	1/4	3/8	1/4
C	-1/16	4/16	8/16	6/16

La transformation S+P avec le prédicteur A est aussi appelée la transformation TS ([28]).

Synthèse de bancs de filtres adaptés, application à la compression des images

Les travaux développés dans cette thèse portent sur les décompositions multirésolution dans un cadre de *lifting scheme*, appliquées à la compression d'images. Pour commencer, une structure de décomposition consistant en un "lifting scheme généralisé" est proposée. Ce schéma permet d'exploiter toute l'information disponible au décodage dans l'étape de prédiction. Cela est rendu possible par l'ajout d'un filtre de prédiction supplémentaire par rapport à la structure classique de *lifting scheme*. Le schéma proposé est ensuite associé à deux méthodes d'adaptation. La première, appelée GAE, suppose une stationnarité globale de l'image, tandis que la seconde, LAE ne suppose qu'une stationnarité locale de l'image. Dans ce dernier cas, les filtres prédictifs sont adaptatifs.

Trois applications de ces méthodes en compression d'images sont ensuite proposées. Dans un premier temps, une comparaison des performances en compression sans perte sur des images de textures, synthétiques, gaussiennes, à stationnarités locale et globale (vérifiant les hypothèses plus haut), est réalisée. Sur ces signaux, les mesures d'entropie d'ordre 1 ont montré que les méthodes proposées offrent en moyenne un gain en codage de 0,5 bpp (GAE) et 0,8 bpp (LAE) par rapport à la décomposition en ondelette (9,7), de 0,8 bpp (GAE) et 1,11 bpp (LAE) par rapport à la (5,3) et de 0,41 bpp (GAE) et 0,65 bpp (LAE) par rapport à la méthode de Gerek et Çetin. La deuxième application concerne le codage sans perte d'images réelles de natures variées. Les gains par rapport à l'état de l'art se sont révélés plus faibles que ceux obtenus pour les images synthétiques. Enfin, la dernière application traite les cas du codage progressif et du codage avec perte. Pour la compression avec pertes, nous avons modifié la méthode LAE pour palier aux problèmes de divergence dus à l'impossibilité au niveau du décodeur de reconstruire les filtres prédictifs à partir d'échantillons quantifiés. Elle se révèle plus efficace que lorsque l'on utilise les filtres usuels de longueur fixe (9,7) et (5,3).

Mots clés : Bancs de filtres adaptés, transformée en ondelettes, *lifting scheme* adapté, filtrage adaptatif, compression sans perte, codage progressif.

Signal-adapted filter banks, application to image compression

This thesis is concerned with multiresolution decompositions in a lifting scheme framework as well as its application to image compression. First, a decomposition structure consisting of a "generalized lifting scheme" is proposed, using all information available in the decoding stage for prediction. This new scheme relies on an additional predictive filter which does not appear in classical lifting scheme structures. The proposed scheme is then associated to two adaptation methods. The first one, called GAE, assumes a global stationnarity of the image, while the second one, called LAE assumes only a local stationnarity. In the latter, adaptive filters are used.

A study of the performance in lossless coding is done by using a set of Gaussian synthetic images with local and global stationarities (satisfying the assumptions on the images). Estimations of the first order entropy of the transformed image showed that the proposed methods offer on average an improvement of 0.5 bpp (GAE) and 0.8 bpp (LAE) compared to the (9,7) wavelet transform, 0.8 bpp (GAE) and 1.11 bpp (LAE) compared to the (5,3) wavelet transform and 0.41 bpp (GAE) and 0.65 bpp (LAE) compared to the method of Gerek & Çetin (a method using LMS-based adaptive filters). The second part is concerned with the study of performance in lossless coding of still images with varied features. The obtained gains in bit rate compared to the state of the art are lower than those obtained for the synthetic images. Finally, in the last part, the application to progressive coding and lossy coding is studied. In order to prevent the LAE-divergence phenomenon due to the quantization step in lossy compression, we modified its structure. Measurements of rate/distortion showed the, for high bit rates, that the modified LAE outperforms the (9,7) and (5,3) wavelet transforms.

Keywords : Wavelets and signal-adapted filter banks, signal-adapted lifting scheme, adaptive filtering, lossy/lossless image compression, progressive coding.